



VI Simpósio sobre Fatores Humanos em Sistemas Computacionais
CURITIBA-PR-BRASIL, 17 A 20 DE OUTUBRO DE 2004
Mediando e Transformando o Cotidiano

Anais do Simpósio

Editores:

Laura Sánchez García
Sérgio Roberto P. da Silva



VI Simpósio sobre Fatores Humanos em Sistemas Computacionais
CURITIBA-PR-BRASIL, 17 A 20 DE OUTUBRO DE 2004
Mediando e Transformando o Cotidiano

Anais do Simpósio

Editores:

Laura Sánchez García
Sérgio Roberto P. da Silva

Simpósio sobre Fatores Humanos em Sistemas Computacionais :
Mediando e Transformando o Cotidiano (6. : 2004 :
Curitiba, PR)

S612s VI Simpósio sobre Fatores Humanos em Sistemas Computacionais :
Mediando e Transformando o Cotidiano (17 a 20 de outubro de 2004):
anais do simpósio / Laura Sánchez García, Sérgio Roberto P. da Silva
(editores) – Curitiba, PR : [s.n.], 2004.
228 p. : il

ISBN 85-7669-017-9

1. Interação humano-computador - Simpósio. 2. Interface humano-computador - Simpósio. I. Sánchez García, Laura. II. Silva, Sérgio Roberto Pereira da. III. Título.

CDD 21. ed. 004.19

ELIANE M. S. JOVANOVIČ CRB 9/1250

Sumário

Boas-vindas da Organização	vii
Comitê de Programa, Promoção, Organização e Apoio	ix
Revisores	xi
Palestras Convidadas	1
Artigos Completos	7
Avaliação de Usabilidade e Avaliação Sense-Making da Ferramenta RVQ-Tool Clodis Bocarolis, John Maycol de Moraes, Patrícia Rucker de Bassi, Laura Sánchez García	9
Avaliação de usabilidade para os computadores de mão: um estudo comparativo entre três abordagens para ensaios de interação Adriana Holtz Betiol, Walter de Abreu Cybis	19
Avaliação heurística participativa do CollabSS Marcos Augusto F. Borges, M. Cecília C. Baranauskas	27
Avaliando a Qualidade da Interação de Usuários com Dispositivos Móveis Milene Silveira, Márcio Pinho, Adriano Gonella, Patrick Calvetti	37
Como conhecer usuários através do Método de Explicitação do Discurso Subjacente (MEDS) Ana Maria Nicolaci-da-Costa, Carla Faria Leitão, Daniela Romão-Dias	47
Contribuições de Visualização para a Área de Educação à Distância Celmar Guimarães da Silva, Heloísa Vieira da Rocha	57
Estudo do impacto do design e das formas de uso sobre a recuperação de informações em fóruns de discussão online Adéle Malta Pontes, Carla Faria Leitão, Clarisse Sieckenius de Souza, Simone Diniz Junqueira Barbosa, Violeta S.T.D.B Quental	69
Ferramentas para Apoio à Construção de Sistemas de Ajuda Online Alex Pacheco da Silva, André Luiz Farias, Ana Paula Lorenzoni, Micheline Parizotto, Milene Silveira	81
Inserindo IHC em Empresas Brasileiras de Informática: Relato, Discussão e Lições Aprendidas Antônio Marcos Endler, Marcelo Soares Pimenta	91
Integração IHC e ES: Processo de Planejamento da Reengenharia de Software Guiado pela Avaliação de Usabilidade – PPR-U Sérgio Luisir Discola Junior, Júnia Coutinho Anacleto Silva, André Constantino da Silva, Marcela Xavier Ribeiro	101
Investigando Aspectos de Interação em Aplicações SIG na Web voltadas ao Domínio Agrícola Juliano Schimiguel, M. Cecília C. Baranauskas, Claudia Bauzer Medeiros	113
Uso de Conhecimento Prévio do Domínio em Programação por Demonstração Raqueline R. de Moura Penteado, Sérgio Roberto P. da Silva	123
TFlex: Um Simulador de Teclado com Múltiplos Modos de Varredura Marcio Juliato, Eduardo H. Tanaka, Christian Baudet, Miguel Galves, Thiago T. Coelho, Heloísa Vieira da Rocha	133
Um método de avaliação para interfaces baseadas em mapas Maria Lúcia Arraes Seixas, Clarisse Sieckenius de Souza	143
Using an Interaction Model to Support Communication among HCI Design Team Members from Multidisciplinary Backgrounds (apresentação em Português) Maíra Greco de Paula, Simone Diniz Junqueira Barbosa	155

Artigos Resumidos

Avaliação de Acessibilidade na <i>Web</i> com a Participação do Usuário: Um Estudo de Caso	165
Amanda Meincke Melo, M. Cecília C. Baranauskas, Fabiana Fator Gouvêa Bonilha	
Desafios de Avaliação de Interfaces de Ambientes Educacionais - Um Estudo de Caso	169
Raquel Oliveira Prates, Carla Leitão Faria, Rosa Maria V. de Figueiredo	
Estratégias Cognitivas para Aumento da Qualidade do Hiperdocumento que Contém o Material Instrucional para EAD	173
Vânia Paula de Almeida, Júnia C. A. Silva	
Integrando a Visão da ES e da IHC através da Aplicação de Padrões sobre o Modelo de Prototipação	177
André Constantino da Silva, Júnia Coutinho A. Silva, Rosângela A. Delosso Penteado, Sérgio Roberto P. da Silva	
Modelando a interação do NiTA: um estudo de caso e extensões ao MoLIC	181
Bruno Santana da Silva, Simone Diniz Junqueira Barbosa	
<i>Range-sliders</i> alternativos para representar grandes quantidades de valores	185
Celmar Guimarães da Silva, Heloísa Vieira da Rocha	
Suporte empírico e neurolingüístico em hipermídia adaptativa	189
Adriana Barbosa Santos, Elcimar Alves Socorro	
The importance of aesthetics on the perceived usability of VLEs: preliminary results	193
R. Parizotto-Ribeiro, N. Hammond, J. Mansano, C. Cziulik	
The Online Community Framework's Performance as an Epistemic Tool	197
Clarissa M ^a de Almeida Barbosa, Carla Faria Leitão, Clarisse Sieckenius de Souza, Jenny Preece	
Um Projeto de Interfaces para um Jogo Multimídia Direcionado a Portadores de Síndrome de Down	201
Ana Paula Canal, Carla Gonçalves Brum, Marcos Luís Cassal	
Uma experiência de construção da ajuda <i>online</i> segundo o paradigma da engenharia semiótica	205
Thaís Chiappetta Salgado, Clarisse Sieckenius de Souza	
Uma proposta para o desenvolvimento de interfaces de usuário multi-plataforma com tecnologia <i>Web</i>	209
Macilon Araújo Costa Neto, Jair C. Leite	
Utilizando ontologia para a descrição da interação em componentes de interface de usuário	213
Lirisnei Gomes de Sousa, Jair C Leite	
Oficinas	217
Tutorias	229

Promoção:



Organização:



Apoio:



Boas-vindas da Organização

Bem-vindos ao IHC 2004!

É com enorme satisfação que apresentamos os Anais do VI IHC – Simpósio sobre Fatores Humanos em Sistemas Computacionais, realizado em Curitiba de 17 a 20 de outubro de 2004, e promovido pelo CEIHC — Comitê Especial de Interação Humano-Computador — da Sociedade Brasileira de Computação.

Na medida em que a difusão das tecnologias da informação e comunicação cresce, a transformação do cotidiano de diversos segmentos da sociedade se mostra irreversível. Hoje, os sistemas computacionais, na forma de grandes redes de computadores, como a Internet, ou de pequenos artefatos computacionais, como telefones celulares, dão suporte a uma infinidade de atividades humanas, antes de difícil execução ou simplesmente inimagináveis. Esta transformação da vida diária mediada pela tecnologia tem demandado da academia, da indústria e de outros segmentos da sociedade uma constante atualização de suas organizações e de suas formas de trabalho, fomentando muitas vezes a consolidação de novas disciplinas e novos nichos de mercado.

Em consonância com esta demanda social, a Interação Humano-Computador (IHC) tem por objetivo o desenvolvimento e a avaliação de atividades humanas mediadas por sistemas computacionais. De diversidade disciplinar elevada, pois as facetas da tecnologia são muitas, a área de Interação Humano-Computador tem crescido com o suporte diferenciado de conhecimento desenvolvido em uma infinidade de disciplinas, incluindo Informática, Psicologia, Antropologia, Design, Sociologia, Comunicação, Letras e Linguística e muitas outras.

Seguindo os esforços anteriores e os anseios da comunidade, esta edição do IHC tem por meta duas frentes adicionais, uma disciplinar e outra institucional. A disciplinar tem por objetivo aprimorar a integração com outras disciplinas do perfil da área no Brasil, fomentando a inserção de respectivos profissionais na comunidade. A institucional visa fomentar a integração entre academia, indústria e outros segmentos da sociedade. Em conformidade com estas metas, tanto os tópicos sugeridos para a submissão de contribuições como o perfil do Comitê de Programa procuraram reforçar a interação de IHC com outras disciplinas, áreas e instituições. Ainda dentro deste espírito, o IHC04 traz dois pesquisadores de renome internacional com palestrantes convidados. O Professor Henry Lieberman com a palestra “*Common Sense Reasoning for Interactive Applications*” e o Professor Liam Bannon com a palestra “*Interaction Design: Another step towards a more human-centred approach to ICTs?*”.

No âmbito da Universidade Federal do Paraná, é mister agradecermos o apoio incondicional recebido dos amigos do Departamento (representados por Marcos Sunye) e do Programa de Pós-Graduação em Informática (representados por Alexandre Direne). A todos os colegas do Comitê de Programa que ajudaram com críticas construtivas, decisões técnicas e apoio logístico cabe, também, passar nossos sinceros agradecimentos. No âmago da nossa Comunidade, de forma especial, somos gratos à disponibilidade total para o suporte seguro de Luis Ernesto Merkle, assim como ao "ombro de mãe" de Clarisse Sieckenius de Souza. Por último, o IHC 2004 não estaria sendo possível sem a contribuição inestimável, competente e amiga de Olga Bellon. Gostaríamos ainda de agradecer o apoio do CNPq, da CAPES e da Fundação Araucária.

Quanto a Curitiba, a cidade hospedeira, ela é conhecida nacional e mundialmente como cidade ideal. Alguns aspectos da Cidade realmente fazem jus à sua imagem. Por outro lado, um olhar acadêmico alternativo e independente sustenta que Curitiba detém índices sociais semelhantes aos de qualquer outra cidade do mesmo porte e que a sua imagem é resultado de um bem-sucedido marketing urbano. Façam a sua própria leitura!

Laura Sanchez García e
Sérgio Roberto da Silva

Comitê de Programa, Promoção, Organização e Apoio

Comitê de Programa

Coordenação Geral Coordenação de Oficinas Coordenação de Tutoriais

Sérgio Roberto P. da Silva
Junia Coutinho Anacleto Silva
Marcelo Morandini

Informática — UEM
Computação — UFSCar
Informática — UEM

Comitê

Adriana Benevides Soares
Álvaro João Magalhães Queiroz
Ana Maria Nicolaci da Costa
Carlos Alberto Faraco
Clarisse Sieckenius de Souza
Heloisa Vieira Rocha
Jair Cavalcanti Leite
Junia Coutinho Anacleto Silva
Laura Sánchez García
Luciana Marta Silveira
Luiz Ernesto Merkle
Marcelo Morandini
Marcelo Pimenta
Marcos Roberto da Silva Borges
Maria Cecília C. Baranauskas
Maristela Mitsuko Ono
Milene Selbach Silveira
Priscila Lena Farias
Raquel Oliveira Prates
Sérgio Roberto P. da Silva
Simone Diniz Junqueira Barbosa
Sueli Mara S. P. Ferreira

Ciências Cognitivas — UGF-Rio
Ciências Cognitivas e Semiótica - Unicamp
Psicologia — PUC-Rio
Letras e Linguística — UFPR
Informática — PUC-Rio
Computação — Unicamp
Computação — UFRN
Computação — UFSCar
Informática — UFPR
Tecno-Arte — CEFET-PR
Informática — CEFET-PR
Informática — UEM
Informática — UFRGS
Computação — UFRJ
Computação — Unicamp
Design — CEFET-PR
Informática — PUC-RS
Design e Semiótica — PUC-SP
Computação — UERJ
Informática — UEM
Informática — PUC-Rio
Ciência da Informação — USP

Promoção:

Sociedade Brasileira de Computação (SBC) — Comitê Especial de Interação Humano-Computador (CEIHC)

Organização:

Universidade Federal do Paraná (UFPR)
Departamento de Informática
Laura Sánchez García

Apoio:

Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES)
Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq)
Fundação Araucária

Revisores

Adéle Malta Pontes - PUC-Rio
Adriana Benevides Soares - UFRJ
Álvaro João M. Queiroz - Unicamp
Ana Maria Nicolaci - PUC-Rio
Carla Faria Leitão - PUC-Rio
Carla Maria Dal Sasso Freitas - UFRGS
Celmar Guimarães da Silva - Unicamp
Clarisse Sieckenius de Souza - PUC-Rio
Eduardo Hideki Tanaka - Unicamp
Elizabeth Furtado - Unifor
Heloisa Vieira da Rocha - Unicamp
Hilton José Silva de Azevedo - CEFET-PR
Jair Cavalcanti Leite - UFRN
Janne Oeiras - Unicamp
Joice Lee Otsuka - UFRN
Junia Coutinho Anacleto Silva - UFSCar
Laura Sánchez García - UFPR
Luciana Martha Silveira - CEFET-PR
Luiz Ernesto Merkle - CEFET-PR
Maíra Greco de Paula - PUC-Rio
Marcelo Morandini - UEM
Marcelo Soares Pimenta - UFRGS
Márcia de Borba Campos - PUC-RS
Marcos Roberto da Silva Borges - UFRJ
Maria Cecília Calani Baranauskas - Unicamp
Maria Cecília Martins - Unicamp
Maristela Mitsuko Ono - CEFET-PR
Milene Selbach Silveira - PUC-RS
Priscila Lena farias - PUC-SP
Raquel Oliveira Prates - UERJ
Ricardo Luís Lachi - Unicamp
Rodrigo de Oliveira - Unicamp
Rosane de Albuquerque dos Santos Abreu - PUC-Rio
Sandra Sueli Vieira Mallin - CEFET-PR
Sérgio Roberto P. da Silva - UEM
Simone Diniz Junqueira Barbosa - PUC-Rio
Sueli Mara Soares Pinto Ferreira - USP
Tatiana Aires Tavares - UFRN

Palestras Convidadas

Henry Lieberman (Media Laboratory, Massachusetts Institute of Technology, EUA)

Common Sense Reasoning for Interactive Applications

Abstract: A long-standing dream of Artificial Intelligence has been to put common sense knowledge into computers-enabling machines to reason about everyday life. Some projects, such as Cyc, have begun to amass large collections of such knowledge. However, it is widely assumed that the use of common sense in interactive applications will remain impractical for years, until these collections can be considered sufficiently complete and common sense reasoning sufficiently robust. Recently, at the MIT Media Lab, we have had some success in applying common sense knowledge in a number of intelligent Interface Agents, despite the admittedly spotty coverage and unreliable inference of today's common sense knowledge systems. This paper will survey several of these applications and reflect on interface design principles that enable successful use of common sense knowledge.

Henry Lieberman has been a research scientist at the MIT Media Laboratory since 1987. His research interests focus on the intersection of artificial intelligence and the human interface. He directs the Lab's Software Agents group, which is concerned with making intelligent software that provides assistance to users through interactive interfaces. In 2001 he edited *Your Wish is My Command*, which takes a broad look at how to get away from "one size fits all" software, introducing the reader to a technology that one day will give ordinary users the power to create and modify their own programs. In addition, he is working on agents for browsing the Web and for digital photography, and has built an interactive graphic editor that learns from examples and annotation on images and video. He worked with the late Muriel Cooper, who headed the Lab's Visual Language workshop, in developing systems that support intelligent visual design; their projects involved reversible debugging and visualization for programming environments, and developing graphic metaphors for information visualization and navigation. From 1972-87, he was a researcher at the MIT Artificial Intelligence Laboratory, and subsequently joined with Seymour Papert in the group that originally developed the educational language Logo, and wrote the first bitmap and color graphics systems for Logo. He also worked with Carl Hewitt on Actors, an early object-oriented, parallel language, and developed the notion of prototype object systems and the first real-time garbage collection algorithm. He holds a doctoral-equivalent degree (Habilitation) from the University of Paris VI and was a visiting professor there in 1989-90.

Liam J. Bannon (Interaction Design Centre -IDC, University of Limerick, Ireland)

Interaction Design: Another step towards a more human-centred approach to ICTs?

Abstract: The field of Human-Computer Interaction (HCI) has been in existence for a quarter of a century. In this talk, I wish to focus on where it may be heading. The understanding of "use" is being coupled with the understanding of "experience". Ubiquitous and pervasive computing concepts are replacing the classic image of the desktop computer. Issues of seamlessness and mobility are becoming crucial. Conceptions of computer "users" are also changing, from being information-processing devices to embodied human, and social, actors. Design, with a capital D, will play an increasingly large role in the development of new products and services. Attention will also turn to issues of engagement, play, and the role of agents. The vision of Ambient Intelligence as presented in recent EU IST Framework programmes will be critically examined. I will discuss what implications the new paradigm of Interaction Design might have, in terms of conceptual frameworks and methods, and present some of the recent work being conducted in our own Interaction Design Centre as illustration.

Liam J. Bannon is Professor of Computer Science in the Dept. of Computer Science and Information Systems, a University Fellow, and Director of the Interaction Design Centre at the University of Limerick, Ireland. He was trained in psychology and computer science at University College Dublin, (B.Sc. (Hons.), Trinity College, Dublin, (M.Sc.), and the University of Western Ontario, Canada (Ph.D.). He was one of the first people to be awarded a Honeywell Human Factors Pre-Doctoral Internship in 1978, held at the Systems Research Centre, in Minneapolis, USA. He subsequently held a post-doctoral fellowship at the Institute of Cognitive Science, University of California, San Diego, working with Don Norman on the Human-Machine Interaction project which resulted in the landmark book *User Centered System Design* (1986). He has lived and worked abroad for many years, holding appointments in Universities in Canada, the USA, Denmark, Sweden, Finland, France, Italy and The Netherlands. He has also worked as a consultant with many international organizations, including the European Parliament, Danish Ministry of Research, NATO, Xerox PARC, Xerox Research Lab, Cambridge. He is widely published and cited. The Interaction Design Centre which he heads comprises over 20 researchers, and has received extensive national and EU research funding for its work in the areas of HCI, CSCW and Interaction Design. He is a founding editor of the *CSCW Journal*, and on the editorial boards of several other Journals, including: *Cognition, Technology & Work*; *Computer-Assisted Learning*, and *Requirements Engineering, Universal Access in the Information Society* and the *International Journal of Cognitive Technology*. He is the co-editor of several books, including, with Zenon Pylyshyn, *Perspectives on the Computer Revolution* (Ablex, 1989) and with Ursula Barry & Olav Holst, *Information Society- Impact on the Way of Life* (Tycooly, 1982). Professor Bannon has received a *Recognition of Service Award* by the ACM at the ACM CHI'2000 Conference in Amsterdam, where he worked as Tutorials Co-Chair. He has also been awarded the *Special Achievement in Research Award* in 1998 by the University of Limerick. Apart from his work in HCI and CSCW, he is also known for his writings on participatory design, and on public issues concerning the so-called "information society".

Artigos Completos

Avaliação de Usabilidade e Avaliação *Sense-Making* da Ferramenta RVQ-Tool

Clodis Boscarioli

Universidade Estadual do Oeste do Paraná
Rua Universitária, 2069 – Cascavel - PR
boscaroli@unioeste.br

Patrícia Rucker de Bassi

Universidade Tuiuti do Paraná
Rua Nicarágua 1833 – Curitiba – PR
bassi@onda.com.br

John Maykol de Moraes

Universidade Estadual do Oeste do Paraná
Rua Universitária, 2069 – Cascavel - PR
John_Maykol@hotmail.com

Laura Sánchez García

Universidade Federal do Paraná
Rua Mal. Mallet, 400 ap. 41 Curitiba – PR
laura@inf.ufpr.br

RESUMO

A tendência atual em avaliação é tentar identificar os problemas de usabilidade tão logo estes sejam detectados na interface. Uma vez identificado, o problema deve ser solucionado, ou, ao menos, ter seus efeitos minimizados. O *Design Participativo* e o *Design Cooperativo* são apoiados em um processo iterativo, que tem, na avaliação, um componente nevrálgico. Dentre as abordagens de avaliação de interfaces de sistemas computacionais consolidadas na Ciência da Computação está a Engenharia de Usabilidade. Em paralelo, na Ciência da Informação, a metodologia *Sense-making* contribui com explicações e técnicas que permitem identificar situações-problema no uso de sistemas de informação, não necessariamente informatizados. Este artigo descreve a aplicação complementar do *sense-making* e de *checklist* de usabilidade como mecanismos de avaliação em uma interface de Sistemas Visuais de Consulta a Banco de Dados, o protótipo RVQ-Tool (*Relacional Visual Query Tool* – Uma Ferramenta de Consulta Visual Relacional), discutindo os resultados obtidos.

ABSTRACT

Now-a-days the focus on user-interface evaluation is trying to identify its problems as soon as they can be detected. Once identified, the problem can be solved, or, at least, their effects minimized. The Participant Design and the Cooperative Design have their support on an iterative process, which have on evaluation their main component. One approach for evaluation on user-interface on computer

VI Simpósio sobre Fatores Humanos em Sistemas Computacionais — Mediando e Transformando o Cotidiano. Hotel Bourbon, Curitiba, 17 a 20 de outubro de 2004. UFPR, CEIHC—SBC.

systems, that is consolidate on Computation, is the Usability Engineering. At the same time, on Science Information, the *Sense-making* methodology contributes with explanations and techniques that permit the identification of problems-situations while been use for systems information, not necessary on computer systems. This paper describes and applies two methods of user-interface evaluation, the usability test and the *sense-making*, on a Visual Query System, the RVQ-Tool (*Relational Visual Query Tool*) prototype. This paper also discusses the results of these evaluation processes.

Palavras-chave

Interface humano-computador: interfaces de usuário, avaliação/metodologia.

INTRODUÇÃO

Há poucas décadas, quando a disciplina de Interação Humano-Computador (IHC) surgiu, uma boa interface era aquela que propiciasse aos usuários a manipulação de válvulas, alavancas e botões em número menor. Nos dias atuais o computador mudou de uma simples troca de caracteres alfanuméricos para interfaces que utilizam vários estilos de interação (janelas, ícones, botões, menu, etc) e recursos computacionais.

A interação do usuário com a máquina tornou-se tão importante quanto o processamento realizado pela mesma. Nesse pressuposto, pesquisadores vêm desenvolvendo técnicas e ferramentas para facilitar o *design* e a avaliação de interfaces de acordo com as necessidades do usuário, com base nos conceitos de IHC.

Segundo Souza em [12] a avaliação da interface é importante no processo de *design*, pois por meio dela é possível estimar o sucesso ou o insucesso tanto da funcionalidade quanto da interação no sistema proposto. Apesar de o projetista ter como base teorias, diretrizes e princípios de *design*, é necessário avaliar o resultado obtido ao longo do processo.

Existem duas formas de avaliação de interfaces, as formativas e as somativas. As formativas são aquelas feitas durante o processo de *design*, permitindo que se identifique e conserte um problema de interação antes que a aplicação esteja concluída, ou até mesmo antes de ser implementada. As somativas, por sua vez, avaliam o produto já implementado [12].

O protótipo utilizado para a realização das avaliações é uma ferramenta de consultas visual a banco de dados relacionais, o RVQ-Tool [9]. A partir da crescente utilização dos Sistemas Gerenciadores de Banco de Dados (SGBDs) para armazenamento de grandes volumes de informação e da complexidade dos conceitos envolvidos na área de Banco de Dados (BD) surgiu a necessidade de tentar elaborar sistemas interativos que facilitassem a construção de consultas para o usuário final. Deste esforço surgiram os VQSs (*Visual Query Systems* – Sistemas de Consulta Visual) [4], melhor indicados por terem como objetivo aproximar os conceitos do mundo real, percebidos pelo usuário, dos esquemas que descrevem os dados dos sistemas. Os sistemas de consulta visual são ferramentas desenvolvidas para permitir que o usuário crie consultas para a extração dos dados desejados sem a necessidade de manipular uma linguagem de consulta textual como a SQL (*Structured Query Language* – Linguagem de Consulta Estruturada).

A avaliação proposta para esse protótipo é do tipo somativa. Para esta avaliação foram abordadas duas vertentes: os testes de usabilidade e um estudo qualitativo *sense-making*. A idéia de aplicar duas metodologias diferentes justifica-se tanto pelo fato da sua apropriação para a validação do ambiente de interface em questão (uma simplificação da integração das propostas apresentadas em [1] e [2]), de cunho aplicativo, quanto pela comparação entre elas, de interesse metodológico.

O artigo inicia-se abordando as duas metodologias de avaliação adotadas, *checklist* de usabilidade e estudo qualitativo do *sense-making*, seguido do processo de avaliação da ferramenta RVQ-Tool. Neste ponto é realizada uma breve explanação sobre a ferramenta, as classes de consultas empregadas na avaliação, a metodologia de avaliação utilizada e análise das respostas. Ao final, são apresentadas conclusões das avaliações realizadas pelas duas metodologias e as sugestões de trabalhos futuros.

BASE TEÓRICA

Engenharia de Usabilidade

Usabilidade é o termo usado para descrever a qualidade de uso de uma interface. Interfaces com usabilidade aumentam a produtividade dos usuários, diminuem a ocorrência de erros, ou importância destes, e, não menos importante, contribuem para a satisfação dos usuários [14].

Sob este aspecto, uma avaliação de usabilidade tem o objetivo de fazer medição quantitativa dos valores

alcançados pelo sistema nos fatores descritos de facilidade de aprendizado do sistema, facilidade de uso, satisfação do usuário, flexibilidade e produtividade [14].

Para isso, devem ser realizados testes do protótipo juntamente com os usuários finais para obter um parâmetro de medição concreto. Esses testes proporcionam resultados que permitem verificar se os objetivos iniciais de projeto foram alcançados satisfatoriamente ou não [12].

Um método de avaliação pode envolver várias técnicas de avaliação de interfaces. O método de avaliação utilizando *checklists* é derivado de [8] e envolve principalmente técnicas de avaliação heurística e testes de usabilidade.

Os *checklists* são questões normalmente fechadas, formuladas a partir de uma série de critérios, conforme detalhado em [7] e [8].

Esta metodologia de avaliação foi utilizada para avaliar a interface do protótipo, com base nas questões descritas em [7], porém, adaptado à realidade de elaboração de consultas a BD de forma visual, objetivando apoiar a inspeção da interface e descobrir seus problemas ergonômicos mais substantivos.

Estudo Qualitativo do *Sense-making*

Apesar de os testes de usabilidade serem capazes de identificar alguns problemas triviais, eles não têm como foco principal os diferentes aspectos da interface e não fornecem nenhum indicador de que ação tomar para se alcançar o resultado desejado [12].

Para alcançar este resultado é necessário um estudo que compreenda a atividade humana de observação, interpretação e compreensão do mundo exterior, inferindo-lhe sentidos lógicos, advindos do uso dos esquemas interiores como o *sense-making* [5].

A base conceitual do *sense-making* (Carter, 1989 apud [5]) é que “o homem cria idéias para transpor os ‘vazios’ (*gaps*) que lhe são apresentados em decorrência da descontinuidade sempre presente na realidade”.

Para Cheuk, em [5], o estado cognitivo do ser humano representa, metaforicamente, um movimento contínuo no tempo e no espaço, sempre em frente, como em uma estrada. A posição atual do indivíduo ao longo desta trajetória é dada em função de onde já esteve (experiências, ambientes, etc), de onde está (no presente) e de para onde vai (futuro).

A metodologia também descreve os usuários como criaturas inteligentes e criativas capazes de compreender o sentido das coisas, a partir da incorporação de conhecimentos tanto advindos do mundo exterior quanto dele próprio. Quando não têm dúvidas ou indecisões quanto às suas necessidades, os indivíduos se movem continuamente em direção aos seus objetivos. Entretanto, lacunas aparecem frequentemente. A ponte necessária para transpô-las surge em função das estratégias empregadas pelos indivíduos para buscar e

utilizar as fontes potenciais de informação, isto é, para solucionar o problema. Esta idéia é ilustrada na Figura 1.

A idéia expressa na Figura 1 leva diretamente à idéia de descontinuidade. Assume-se que o ser humano passa por várias etapas em suas experiências. Cada momento é considerado novo, mesmo que seja repetição de comportamentos passados, sendo nesta teoria considerado uma nova etapa, porque ocorre em um momento diferente no tempo e no espaço.

Algumas das formas de compreensão buscadas pelo *sense-making* são: Como o indivíduo interpreta e transpõe este momento? Quais estratégias ele usa para solucionar a situação na qual defrontou-se com a lacuna? Como ele interpreta esse problema e as possibilidades de resolvê-lo? Como se move taticamente para isso? Como reinicia sua jornada?

Na Figura 2 pode-se ver a metáfora gerada pelo modelo de Dervin descrito em [6], conhecido como modelo de três pontas, construído sobre o triângulo Situação-Lacuna-Usos/Ajuda. Pode-se perceber que, com o passar do tempo o usuário passa por várias etapas, que podem individualmente, apresentar novos vazios, novas buscas, novos usos e assim sucessivamente.

A situação é o componente mais abrangente deste modelo. É o contexto temporal e espacial no qual surge a necessidade de informação, estabelece-se o período em que a busca e uso da informação vai ocorrer, e leva (ou não) à compreensão do problema. (Dervin, 1983 apud [5]) coloca uma situação de necessidade de informação como aquela em que o senso interno individual tende a se esgotar, e a pessoa deve criar um novo senso para tal situação, que é algo que está sempre em mudança com o tempo.

O indivíduo, no contexto de alguma situação, encontra uma série de lacunas, ou seja, pontos que ele não compreende ou compreende apenas em parte, e que o leva a interromper seu trajeto. Cada uma destas lacunas pode ser definida como a situação problemática, um estado anômalo de

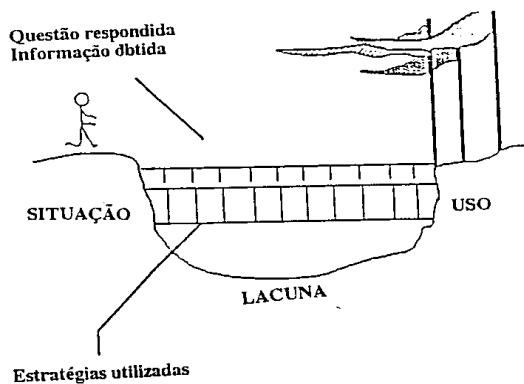


Figura 1: A metáfora do *sense-making*. Fonte: [6].

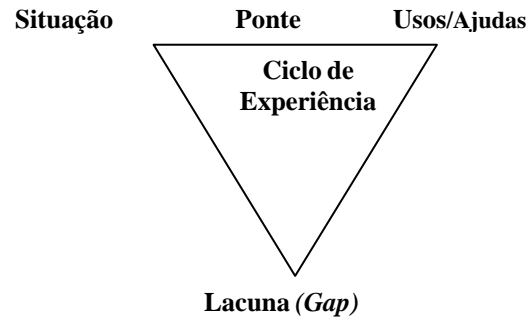


Figura 2: Triângulo do *sense-making*. Fonte: Adaptado de [5]

conhecimento, uma situação na qual um indivíduo está tentando chegar à compreensão de alguma coisa [5].

O indivíduo é, então, levado a fazer algum tipo de uso de qualquer ponte que seja construída para transpor o vazio defrontado por ele. Uso/ajuda, portanto, é o emprego dado ao conhecimento recém adquirido ou uma nova informação útil.

Desta forma, ao se tentar estudar e compreender como determinada pessoa percebe sua condição, esta abordagem assume ser imprescindível avaliar esses três pontos básicos: situação-lacuna-uso/ajuda, como uma base mínima para auto-orientação. Somente assim estar-se-á respeitando o fato que pessoas percebem o mundo de maneiras diferentes.

Ainda, de acordo com [5] o emprego desta abordagem em estudos de comportamento de busca e uso da informação pressupõe, também, a aceitação dos seguintes atributos:

- a) Individualidade: usuários devem ser tratados como indivíduos e não como conjunto de atributos demográficos;
- b) Situacionalidade: cada usuário se movimenta através de uma única realidade de tempo e espaço;
- c) Utilidade da informação: diferentes indivíduos utilizam a informação de maneira própria, e informação é o que auxilia a pessoa a compreender sua situação;
- d) Padrões: analisando as características individuais de cada usuário, tem-se o propósito de chegar aos processos cognitivos comuns à maioria.

Sense-making é uma teoria densa e não fornece um padrão para a elaboração de um questionário a ser aplicado. Ao contrário, frisa que as questões devem ser tão abertas quanto possível, de tal maneira que o usuário entrevistado possa responder, com naturalidade, sobre uma situação específica no uso do sistema, que em nosso estudo é a elaboração de uma consulta a um BD na ferramenta visual.

As questões foram criadas com o intuito de captar a compreensão do usuário da interface e do domínio da aplicação, sendo implicitamente organizadas sobre os pontos básicos da teoria: situação-lacuna-uso/ajuda. Esta organização é subjacente, não sendo evidenciada aos usuários finais.

Uma das principais contribuições deste artigo é a utilização da técnica *sense-making* para avaliar a usabilidade de interfaces¹.

A técnica de *checklists* é bem conhecida na área de IHC por ser minimalista, limitada às questões predefinidas e cuja interpretação pode ser de difícil realização, sendo, portanto, ferramentas muito úteis para avaliar questões pontuais. Contudo, não são recomendadas para avaliar questões mais abstratas, tais como aprendizagem da interface, desenvolvimento de tarefas complexas, etc, pontos que podem ser identificados pela abordagem cognitiva do *sense-making*, que emerge como complementar aos *checklists*.

O PROCESSO DE AVALIAÇÃO DA FERRAMENTA RVQ-TOOL - *Relational Visual Query Tool*

Baseado em estudos realizados em [1] e [2] sobre *design* de interfaces para BD e nas teorias de IHC, foi desenvolvida a ferramenta RVQ-Tool (*Relational Visual Query Tool* – Ferramenta de Consultas Visuais Relacionais), que permite ao usuário elaborar, por meio dos elementos da interface, consultas para a extração de dados de um esquema de banco de dados, visando aumentar a usabilidade e a funcionalidade de um ambiente para elaboração de consultas.

RVQ-Tool classifica-se como um ambiente visual de consultas a banco de dados [13], cujo principal objetivo é comunicar de forma clara ao usuário as informações referentes ao conteúdo do banco de dados, focando em características essenciais e omitindo detalhes desnecessários [3]. As informações, mapeadas e tratadas no sistema como conceitos internos de banco de dados, são apresentadas ao usuário de forma clara, via interface.

A construção da interface gráfica da ferramenta foi baseada no estilo de interação WIMP (*Windows, Icons, Menus and Pointers* - Janelas, Ícones, Menus e Apontadores) utilizada pelo ambiente *Windows*, que pode ser considerado um *framework* de interface apoiado pela tecnologia de interfaces gráficas (GUI – *Graphical User Interfaces* – Interfaces Gráficas do Usuário).

Assume-se neste protótipo que o usuário possui um conhecimento básico do estilo WIMP (que seja um usuário de ambiente *Windows*). Caso ele não detenha tal conhecimento, maior será o esforço despendido para a compreensão do funcionamento da RVQ-Tool, necessitando recorrer a sistema de ajuda *on-line* e/ou treinamento.

A Ferramenta RVQ-TOOL – Breve Descrição

RVQ-Tool permite ao usuário elaborar consultas para extração de dados de um esquema de banco de dados,

visando aumentar a usabilidade e funcionalidade de um ambiente para elaboração da consulta.

A compreensão da realidade de interesse utiliza estratégia *top-down*, onde primeiramente o usuário percebe os aspectos gerais, ou seja, as tabelas e seus atributos, mesmo que indiretamente. Após, pode selecionar os atributos desejados e estabelecer critérios filtrando os dados. Outros detalhes mais específicos podem ser vistos e/ou aplicados pelo usuário, como atributos chave-primária, chave-estrangeira, tipo e tamanho dos atributos e funções agregadas.

O usuário é guiado durante todo o processo da elaboração da consulta, sendo a interface uma espécie de assistente de tarefas (*wizard*) que estabelece etapas bem definidas para a tarefa, podendo retroceder ou avançar entre essas etapas. Para que o usuário tome ciência em qual etapa está e/ou ações devem ser realizadas e da sua ordem, o sistema disponibiliza as mensagens: Explicativas, Dicas, Barras de Título dos Formulários e Caixas de Mensagem.

O ambiente do protótipo é dividido em três etapas: a primeira etapa (Etapa 1) consiste na seleção do esquema de trabalho, onde o usuário poderá optar entre abrir um esquema de banco de dados (um arquivo de banco dados) ou abrir uma consulta existente (um arquivo contendo a pseudolinguagem do protótipo).

Depois de aberto o esquema de dados ou a consulta, passa-se para a segunda etapa (Etapa 2), a elaboração da consulta, que é subdividida em duas sub-etapas: na primeira (Sub-etapa 2.1) o usuário deverá primeiramente selecionar os atributos (informar quais as instâncias que aparecerão na visualização do resultado), em seguida (Sub-etapa 2.2) o usuário pode estabelecer critérios relacionados aos atributos selecionados, conforme Figura 3, restringindo o número de instâncias.

Após o usuário confirmar a sub-etapa 2.1 e/ou sub-etapa 2.2 sempre é dado um *feedback* na forma de pseudolinguagem,

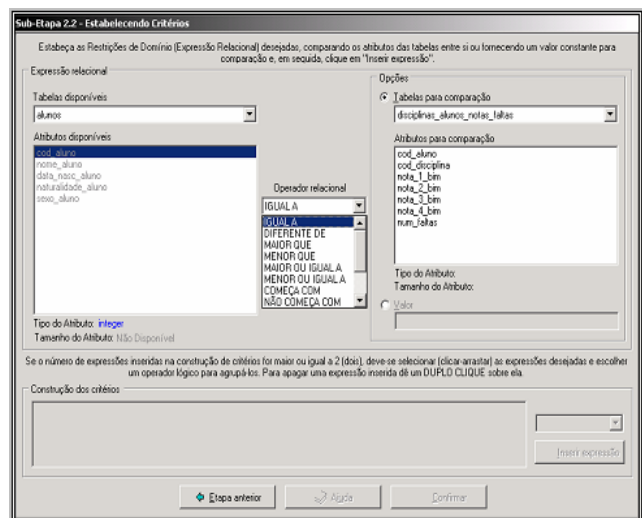


Figura 3: Sub-etapa 2.2 – Estabelecendo critérios.

¹ Suprimiu-se, neste artigo, as informações completas sobre a metodologia de avaliação – questionários e respostas – por restrições de espaço, mas estas podem ser disponibilizadas, em contato com os autores.

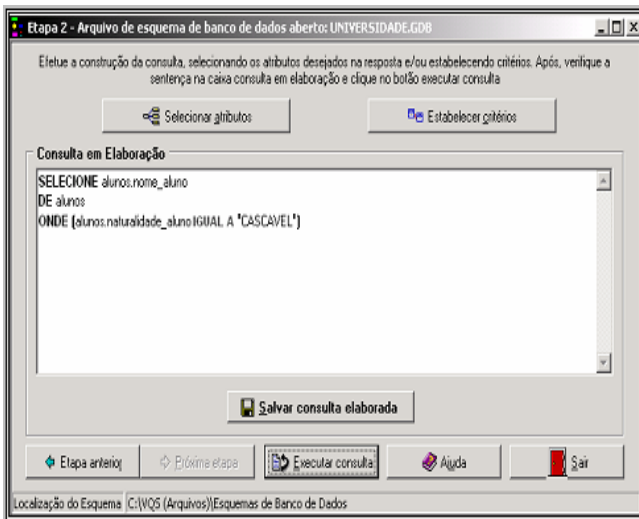


Figura 4: Exibição de feedback em pseudolinguagem.

que exprime o estado atual da elaboração da consulta, como ilustra a Figura 4.

Concluída a seleção de atributos e/ou estabelecimento de critérios o usuário poderá executar a consulta passando para a etapa final (Etapa 3), na qual ocorre a visualização do resultado, onde o usuário pode visualizar, de forma planar, os dados que foram extraídos da base de dados a partir da consulta formulada. Ainda nesta etapa, o usuário possui a opção de exportar os dados obtidos da consulta elaborada para arquivos do tipo texto (.txt), do tipo Microsoft Word (.doc) ou Microsoft Excel (.xls). Esta exportação de resultados é uma espécie de flexibilidade da ferramenta, que permite ao usuário utilizar os resultados obtidos em outras ferramentas para outros fins, como por exemplo, gerar gráficos, incluir uma tabela resposta em um relatório, etc.

O usuário pode salvar a consulta elaborada, de forma a armazenar todo o processo de interação, podendo reaproveitá-lo em outras situações de elaboração de consulta ou mesmo, visualizar novamente o resultado de sua execução.

Também é possível, após a execução da consulta retornar a etapa anterior e selecionar ou remover atributos e critérios.

Questões Metodológicas

O domínio de aplicação escolhido foi o de uma universidade. O esquema associado envolve disciplinas, cursos, alunos e professores e suas relações em uma instituição qualquer. Neste contexto, um conjunto de consultas representativas foi especificado. Esta seção destina à apresentação do processo de avaliação da ferramenta (teste de usabilidade e *sense-making*) sobre este esquema de trabalho.

As consultas utilizadas neste estudo foram divididas em três classes: de nível baixo, médio e avançado, conforme a complexidade que cada consulta apresenta, segundo [4]. O

Operador	Peso
Identidade/Projeção em uma subconsulta ou consulta.	$n^{1.5}$ (onde n é o nível de profundidade)
Renaming	3
Select	$n^{1.2}$ (onde n é o número de condições)
Complement	1.5
Self join	3.5
Natural join	1.5
Theta join	1.5

Tabela 1: Pesos adotados para mensurar a complexidade de uma consulta. FONTE: Adaptada de [4].

intuito desta divisão vai ao encontro da expectativa de descobrir quão útil é uma ferramenta visual para elaboração de consultas avançadas, buscando evidenciar características que possam ser utilizadas de forma complementar, no auxílio à interação e aprendizado de conceitos de BD.

Em [4] é evidenciada uma discussão envolvendo diversos especialistas em linguagens de consulta a BD para determinar um peso inicial para cada operador relacional, de forma a computar a complexidade de uma consulta, com base no conjunto de operadores utilizado por ela, como descreve a Tabela 1.

Classe de Nível Baixo: o nível de complexidade desta classe foi definido a partir do peso da consulta elaborada, que deve ser < 3 (menor que 3).

Classe de Nível Médio: a complexidade desta classe também foi definida a partir do peso da consulta elaborada, que deve ser ≥ 3 (maior ou igual a 3) e ≤ 5 (menor ou igual a 5).

Classe de Nível Avançado: Esta última classe possui complexidade a partir do peso > 5 (maior que 5) da consulta elaborada.

Segue um exemplo de como classificar uma consulta nas classes supracitadas. Considerando a seguinte consulta desejada pelo usuário: Elaborar uma consulta que mostre o nome dos cursos e sua respectiva data de fundação, para todos os cursos fundados entre o período de '1/1/1980' e '1/1/1990'.

A consulta acima é classificada como de nível médio, pois só possui uma projeção (SELECIONE), e, portanto, $n^{1.5} = 1^{1.5}$ mais o valor das duas condições, $n^{1.2} = 2^{1.2} = 2,29$. Logo, o peso definido pelo nível de complexidade é 3,29, pois $1^{1.5} + 2^{1.2} = 3,29$.

A Figura 5 mostra como ficaria a consulta elaborada pelo usuário no protótipo.

A avaliação da interface do protótipo contou com 32 (trinta e dois) usuários, todos com conhecimento em ambientes *Windows*, agrupados por classes, em conformidade com o conhecimento na área de Banco de Dados; classe 01, abrangendo 17 pessoas com pouco ou nenhum conhecimento de banco de dados; classe 02, composta por 08 pessoas com conhecimento introdutório de banco de

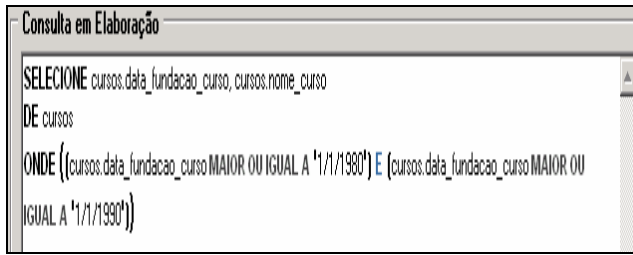


Figura 5: Uma consulta de classe média em RVQ-Tool

dados e classe 03, contendo 07 pessoas com domínio na área em tela.

Foram elaborados dois questionários diferenciados: o teste de usabilidade para avaliação da ferramenta com *checklists*, envolvendo critérios como prestação, legibilidade, *feedback*, consistência, entre outros, somando um total de 80 (oitenta) questões (sendo 01 aberta e 79 fechadas); e outro sobre o estudo qualitativo *sense-making* contendo questões de situação-lacuna-uso/ajuda, evidenciando questões sobre o aprendizado/domínio da interface, somando um total de 23 (vinte e três) questões (sendo 18 abertas e 05 fechadas). Cabe salientar que, dada uma classe de usuário, esta foi dividida e, um subgrupo respondeu a um questionário e o restante dos usuários ao outro. Isto foi feito no intuito de comparar para cada um dos perfis de usuários as metodologias utilizadas.

Os participantes foram submetidos ao uso da ferramenta RVQ-Tool para a formulação de 03 (três) consultas, em um tempo de 45 (quarenta e cinco) minutos, onde cada consulta elaborada deveria ser salva para posterior análise. Concluída essa sessão, ambos os questionários foram entregues aos usuários para preenchimento.

Análise das Respostas Comuns aos Dois Questionários

As respostas comuns aos dois questionários dizem respeito à interação do participante com a ferramenta de forma geral, e seus resultados foram:

- 62,5% (vinte participantes) acharam-na de interação fácil;
- 37,5% (doze participantes) acharam a interação um pouco difícil;
- 75% (vinte e quatro participantes) acharam necessário treinamento antes da utilização da ferramenta;
- 25% (oito participantes) acharam que treinamento é desnecessário;
- 78% (25 participantes) cometeram menos de 06 (seis) erros durante a interação com a ferramenta;
- Das três consultas elaboradas apenas 35% (11 participantes) conseguiram concluir dentro do tempo estipulado;
- 65% (21 participantes) não conseguiram concluir todas as consultas, alegando tempo insuficiente.

Análise das Respostas do Checklist de Usabilidade

A partir dos resultados computados foi possível elaborar um resumo de problemas que foram identificados pelo *checklist* do teste de usabilidade utilizado, conforme Tabela 2, disponível no Anexo 1.

As respostas agrupadas na Tabela 2 dizem respeito à interação do participante com a ferramenta de forma específica e foram agrupadas por critério. Tendo como objetivo generalizar e evidenciar os itens a serem melhorados na interface, que apresentava, por exemplo, transições automáticas (não comandadas pelo usuário) entre as etapas de elaboração de uma consulta, nem sempre percebidas pelo usuário durante a construção da consulta.

Análise das Respostas do Sense-making

Com base no estudo qualitativo do *sense-making* foi elaborado um resumo dos problemas encontrados sobre o triângulo Situação-Lacuna-Usado/Ajuda constatados nas etapas da elaboração da consulta, apresentados na Tabela 3, disponível no Anexo 2.

Para as situações apresentadas na Tabela 3, os seguintes itens foram constatados nas ações dos usuários:

- Verificar no formulário com cabeçalhos das consultas a executar se estava faltando alguma informação, pois a interface não lhe fornecia todas as respostas necessárias;
- Testar várias possibilidades, ou seja, o usuário executava grande parte dos recursos da interface, senão todos, para a compreensão da elaboração da consulta;
- Ler novamente os recursos das interfaces para verificar se alguma informação não havia sido deturpada ou ignorada. Isto é, se a partir dos testes o participante não tinha plena ciência do que estava acontecendo, este reiniciava todo o processo lendo minuciosamente a interface, interpretando todas as dicas, ícones, mensagens de orientação e estados dos componentes disponíveis.

Para as lacunas expostas na Tabela 3 foram identificados:

- Uso de conhecimentos anteriores. Fazendo com que o participante utilizasse o conhecimento baseado na experiência com outros sistemas para relacionar cada situação que encontrasse durante a elaboração da consulta com algo similar à sua realidade;
- Processo iterativo de tentativa e erro, até encontrar a resposta cabível. Neste caso, o usuário se guiava durante a elaboração da consulta pelo *feedback* da interface e sua conclusão era baseada no uso do próprio conhecimento ao qual associava mentalmente os elementos de cada ação.

Para os usos/ajudas mencionados na Tabela 3 constatou-se que os usuários participantes do experimento consideravam que se o processo para obtenção das respostas baseado no seu conhecimento e nas associações mentais eram “corretas”, então, este seria repetido até que as respostas obtidas não correspondessem mais ao esperado.

Ainda na tabela 3, as pontes para o caso situação-lacuna-uso criadas foram:

- *Feedback* da consulta elaborada onde, a primeira parte que auxiliava o participante a acreditar que a ponte feita (experiência e associações mentais) para sobrepor a lacuna daquela situação estava seguindo o caminho correto era o *feedback* da consulta elaborada, mostrando que estava de acordo com o que ele esperava;

- Obtenção dos resultados da consulta foi a segunda parte que auxiliou os participantes a completar suas conclusões sobre a certeza da escolha da ponte correta, ou seja, o retorno dos dados extraídos do esquema de banco de dados, de acordo com suas expectativas. Este fato o estimulava a repetir as mesmas pontes para a obtenção de novos resultados.

O principal obstáculo identificado na obtenção das pontes para o caso situação-lacuna-uso/ajuda da tabela 3 foi a combinação de atributos de várias tabelas no resultado da consulta. Por exemplo, notas dos alunos com número de faltas e nome do professor e código da disciplina. Às vezes, a ponte escolhida pelo participante levava a uma resposta correta, mas o participante nem sempre enxergava e acabava escolhendo outras formas para obtenção dos resultados (outras pontes) devido às várias fontes de dados, que nem sempre mostrava claramente se os resultados obtidos estavam corretos.

As avaliações supracitadas apesar de apresentarem alguns pontos comuns, identificaram outros diferentes, ampliando a área de análise da interação do usuário com o protótipo, uma vez que nesta primeira avaliação os usuários só contavam com os *widgets* presentes na interface.

A metodologia *sense-making* colaborou em muito com a reformulação dos elementos de metacomunicação da interface avaliada, indicando que esta pode ser comparada a técnicas como avaliação heurística e teste de comunicabilidade [10], sendo este último de principal interesse por gerar tanto medidas quantitativas quanto qualitativas.

O método de avaliação de comunicabilidade é baseado na engenharia semiótica e tem como objetivo avaliar a interface do software quanto a sua propriedade de comunicabilidade. Em relação às medidas quantitativas é possível avaliar os problemas mais graves, ou pelo menos mais frequentes, de interação e usabilidade do produto. Já quanto às medidas qualitativas o teste permite identificar pontos de rupturas da comunicação *designer*-usuário. Como o teste é baseado em interjeições potenciais do usuário quando da manipulação da interface, é possível informar ao *designer* o que foi que o usuário não entendeu, dando indicadores de que tipo de informação precisa ser alterada ou acrescentada para que ele a compreenda [10].

Embora nenhuma avaliação heurística tenha sido aplicada à ferramenta, tentou-se respeitar o conjunto de heurísticas desenvolvidas por (Nielsen, 1993 apud [11]), sobre problemas de usabilidade, de forma a tentar evitá-los ou, ao menos, minimizá-los. Estas heurísticas de usabilidade foram respeitadas com base na apropriação dos conceitos e no uso de técnicas da Engenharia Semiótica, envolvendo o

seguinte conjunto de regras: estética e *design* mínimos, coerência entre o sistema e o mundo real, reconhecer é melhor que lembrar, consistência e padronização, visibilidade do *status* do sistema, controle e liberdade de ações, flexibilidade e eficiência de uso, ajuda aos usuários a reconhecer, diagnosticar e recuperar erros, prevenir erros, existência de documentação e ajuda *on-line* [11].

Por exemplo, o ambiente do protótipo considera as heurísticas citadas quando procura minimizar as expressões da área de banco de dados, atuando em um nível mais focado no domínio da aplicação. Contudo, a forte dependência do domínio de banco de dados é um fator limitante, que deve ser contornado pelo projeto de comunicação. Também, o usuário tem opções no sistema para retomar atividades (quando retorna a fases de seleção de atributos ou de definição de critérios) ou mesmo cancelá-las (por meio de botões específicos como o Etapa Anterior, que possui a função de cancelar a etapa corrente).

CONCLUSÕES

Muitas ferramentas têm sido propostas para resolver o problema da interação do usuário com um banco de dados de maneira visual, mas todas, de alguma forma, deixaram espaços que outras ferramentas vieram posteriormente preencher. Este trabalho objetivou preencher uma destas lacunas, evidenciando os problemas de uma ferramenta específica por meio da aplicação de duas abordagens teóricas de avaliação de interface-usuário, tentando, dessa forma, trilhar novos rumos a serem seguidos a partir das deficiências identificadas nesse processo.

Concluiu-se que quanto maior o conhecimento sobre banco de dados, mais facilitada/intuitiva tornou-se a interação com a ferramenta. Contudo, para um primeiro contato, a ferramenta mostrou-se potencialmente útil para usuários leigos no domínio.

Uma prova concreta da afirmação anterior é que cada um dos trinta e dois participantes conseguiu efetuar ao menos uma consulta correta, o que vem a confirmar que a utilização desse tipo de ferramenta para usuário com nenhum ou pouco conhecimento na área de BD é útil e bastante promissora.

As respostas obtidas a partir do *checklist* de usabilidade mostraram que havia rupturas na comunicação da interface indicando a necessidade da reformulação das mensagens e readequação dos processos de interação do participante com os componentes e estados da interface, de forma a minimizar a “sensação de perdido” do participante durante a elaboração da consulta. Já as respostas relatadas nos questionários *sense-making* evidenciaram a necessidade de um treinamento prévio e da disponibilização de um sistema de auxílio *on-line* na ferramenta, com o intuito de fornecer subsídios conceituais na área de BD, o que, certamente, diminuiria o índice de erros na elaboração das consultas.

Como já dito, o uso de questionários *sense-making* vem complementar os *checklists* de usabilidade pela sua maior

abrangência e subjetividade, não sendo passível uma comparação entre ambos.

Os testes relativos a ambas metodologias identificaram problemas na interação do participante com a ferramenta, trazendo insumos significativos para este estudo. A complementaridade das metodologias é notória, uma vez que a primeira (*checklist*) teve impacto direto sobre o uso da interface retornando resultados diretos, mas restritos a fatores pontuais previstos pela tecnologia em questão. E a segunda (*sense-making*) evidenciou problemas que são observados principalmente de forma indireta, com um ganho nas informações que fogem ao escopo abordado, ou seja, possíveis problemas ou modos de interação que nem sempre foram observados e/ou imaginados pelo *designer*.

Como primeiro resultado, este experimento mostrou que, mesmo tendo sido construído com base em sólidos princípios teóricos, o ambiente de interface ainda carece de revisão. Metodologicamente, este resultado reforça a premissa do *Design Cooperativo*, que afirma que não importa quão bom seja o *designer*, pois ele sempre erra na primeira tentativa, justificando o modelo iterativo em estrela que tem, no centro, o processo de avaliação.

Como segundo resultado teórico, o experimento reforça a relevância da abordagem interdisciplinar do *design* e na avaliação de interfaces, de forma a se lançar mão de teorias e resultados oriundos de diversas áreas do conhecimento. Neste caso a Ciência da Computação e a Ciência da Informação, no tratamento de problemas novos ou reeditados pelo contexto tecnológico.

Nesta etapa, propositadamente, os usuários não tiveram treinamento prévio sobre a ferramenta e também não foi disponibilizado um sistema de ajuda *on-line*, deixando os participantes do experimento guiarem-se apenas pelas informações visuais e textuais da interface.

A partir dos resultados apresentados melhorias foram inseridas no protótipo RVQ-Tool, tais como: uma apresentação inicial das etapas do protótipo, modificação das mensagens de aviso, erro e confirmação. Na seqüência, uma nova avaliação foi aplicada a partir da interação com usuários considerados mais inexperientes. A análise final desses dados validou o protótipo como sendo um sistema de consulta visual com facilidades de interação para extração de dados a BD, permitindo ao usuário a possibilidade de aprendizado, compreensão da elaboração de uma consulta a BD, uso de flexibilidades para estender a novas aplicações, minimização do esforço físico e cognitivo e um grau de desempenho que pode ser considerado como bom.

Esta pesquisa tratou de lacunas encontradas em sistemas de consulta visual, contudo, ainda há um grande campo de pesquisas e, como trabalhos futuros nesta vertente de avaliação de interfaces e incremento de usabilidade de tais ferramentas pretende-se:

- Submeter RVQ-Tool a outros usuários, com outros domínios de aplicação, avaliando-o e, pelo *feedback*

otimizar sua interface, melhorando sua comunicabilidade com o usuário;

- Aumentar o tempo de teste e disponibilizar o *help* do sistema e comparar resultados com a avaliação já realizada;

- Elaborar algumas extensões na parte de acesso a banco de dados da ferramenta;

- Estender e aprofundar a integração de metodologias de *design* e avaliação de interfaces oriundas das Ciências da Computação e da Informação;

- Realizar comparação da técnica de *sense-making* com o método de avaliação de comunicabilidade de um software, descrito em [10].

REFERÊNCIAS

1. Bassi, Patricia R 2002. *Um Ambiente de Interface Visual para a geração de Sub-esquemas para uma Ferramenta de Consulta baseada no Modelo ERC+*. Dissertação de Mestrado. Departamento de Informática. Setor de Ciências Exatas, Universidade Federal do Paraná. Curitiba, 2002.

2. Boscarioli, C. 2002. *Um Ambiente de Interface Visual para a Elaboração de Predicados para uma Ferramenta de Consultas a BD Sob o Modelo ERC+*. Dissertação de Mestrado. Departamento de Informática. Setor de Ciências Exatas, Universidade Federal do Paraná. Curitiba, 2002.

3. Boscarioli, C.; García, L. S.; Sunye, M. S. Uma Interface Visual para a Elaboração de Consultas em uma Ferramenta baseada no Modelo ERC+. *Proceedings V Symposium on Human Factors in Computer Systems*. pg. 165-176. Fortaleza, 2002.

4. Catarci, T.; Santucci, G. *Diagramatic Vs Textual Query Languages: A Comparative Experiment* - Dipartimento di informatica e Sistemistica, Università di Roma "La Sapienza", 1992.

5. Cheuk, B. *Qualitative Sense-Making Study of the information seeking situations faced by professionals in three workplace contexts*, 1999.

6. Dervin, B. *An overview of Sense-Making research: concepts, methods and results to date*. In: International Communications Association Annual Meeting. Dallas, 1983.

7. labIUtil. "*Checklists*", Relatório Técnico, O labIUtil, *Laboratório de Utilizabilidade de Informática* da UFSC (INE/CTC/UFSC), 2003.

8. Matias, M. *Checklist: Uma Ferramenta De Suporte À Avaliação Ergonômica De Interfaces*, Dissertação de Mestrado, PPGEP-UFSC – Florianópolis, 1995. Disponível em <http://www.eps.ufsc.br/disserta/matias/>

9. Moraes, J. M. RVQ-TOOL - Uma Ferramenta de Consulta Visual para Banco de Dados Relacionais. Monografia de Graduação. Colegiado de Informática, Centro de Ciências Exatas e Tecnológicas, Universidade Estadual do Oeste de Paraná, Cascavel, 2003.
10. Prates, R.; Souza, C.; Barborsa, S. *A Method for Evaluating the Communicability of User Interfaces*. In ACM Interactions, 2000.
11. Romani, L. A S.; Baranauskas, M. C. C. *Avaliação Heurística de um Sistema altamente dependente do domínio*. Relatório Técnico. Instituto de Computação. UNICAMP, 1998.
12. Souza, C.; Leite, J.; Prates, R.; Barbosa, S. *Projeto de Interfaces de Usuários: Perspectivas Cognitivas e Semióticas. Apostila do Curso de Projeto de Interfaces de Usuário* – Departamento de Informática, PUC-Rio. Rio de Janeiro – RJ, 1999.
13. Spaccapietra, S.; Parent, C.; Sunye, M. S.; Yetongnon, K., Leva, A. di. An Object + Relationship Paradigm for Database Applications. Readings in Object-oriented Systems. D. Rine (Ed.), IEEE Press, 1995.
14. Winckler, Marcos; Pimenta, Marcelo S. *Avaliação de Usabilidade de Sites Web*. X Escola de Informática da SBC-Sul - ERI 2002, Caxias do Sul, Criciúma, Cascavel, 2002.

ANEXOS

Anexo 1

Presteza	Difícil compreensão de funções adicionais
	Não percepção de tamanho e tipo dos atributos
	Inexistência de auxílio (ajuda <i>on-line</i>)
Agrupamento por Formato	Nem sempre era perceptível a obrigatoriedade das opções
Feedback	Algumas opções não apresentaram retorno de uma ação
Legibilidade	Algumas janelas eram sobrepostas, dificultando a visualização
Ações Mínimas	Difícil movimentação caractere a caractere
Ações Explícitas	Certas opções não foram comandadas pelo usuário
Controle de Usuário	Sem interrupção das ações em andamento
	Sem reversão de uma ação
Proteção Contra Erros	Não testava todas as ações possíveis a partir da simulação
	Nem sempre possuía duplo aviso em caso de perda irreversível de dados da consulta em elaboração
Mensagens de Erros	Nem sempre as mensagens de erro eram claramente informadas
	Nem sempre era fácil corrigir um erro
Consistência	Os símbolos de entrada de dados não eram padronizados
Significados	Títulos das janelas pouco explicativos
	Rótulos de opções pouco significativos
	O vocabulário empregado (jargão técnico) possui certo grau de complexidade

Tabela 2: Problemas vistos pelo *checklist* de usabilidade.

Anexo 2

Situação	Lacuna	Ajuda/Us	Pontes
Início da consulta sem conhecimento prévio na área de BD	O que deve ser feito primeiro (qual direção seguir)	Esgotar o número de tentativas	A consulta elaborada ou em elaboração
	Conceito de atributos		
Abertura do esquema de BD de modo interativo	Conceito de critérios		
Mensagens mais destacadas	Como selecionar os atributos		
A complexidade da elaboração da consulta	Como utilizar os operadores E/OU		Os resultados da consultas
Localização das Tabelas	Quando se devia confirmar ou voltar à etapa anterior		
Passos para a seleção dos atributos	Quando se apaga um critério com várias expressões agrupadas se era ou não perdido todo o critério	Experiência com outros programas	
O agrupamento dos critérios	Para que servem as funções adicionais		
	Como elaborar outra consulta partindo do zero		

Tabela 3: Problemas identificados pelo *sense-making*.

Avaliação de usabilidade para os computadores de mão: um estudo comparativo entre três abordagens para ensaios de interação

Adriana Holtz Betiol

Pontifícia Universidade Católica do Paraná
R. Imaculada Conceição, 1155, Curitiba, PR
abetiol@netpar.com.br

Walter de Abreu Cybis

Universidade Federal de Santa Catarina
CP 476, Florianópolis, SC
cybis@inf.ufsc.br

RESUMO

Esta pesquisa teve como objetivo investigar qual a influência do contexto de avaliação nos resultados da avaliação de usabilidade da interface com o usuário de uma aplicação de Internet móvel para telefone celular baseada na técnica de ensaios de interação. Os ensaios de interação foram realizados segundo três abordagens distintas, utilizando: (i) emulador operado em laboratório, (ii) telefone celular operado em laboratório e (iii) telefone celular conectado a uma câmera sem fio operado em campo. Em relação ao desempenho do usuário e à identificação de problemas de usabilidade o experimento identificou mais semelhanças que diferenças nos resultados obtidos nos três contextos de avaliação.

ABSTRACT

The objective of this research was to investigate the influence of the evaluation context in the results of usability tests conducted to evaluate the user interface of a mobile Internet application for mobile phones. The usability tests were conducted according to three different approaches: (i) using a computer-based mobile phone emulator inside the laboratory, (ii) using a mobile phone inside the laboratory and (iii) using a mobile phone linked to a wireless camera in the field. The results regarding user performance and problem identification showed the existence of more similarities than significant differences between the results of the three evaluation contexts.

Palavras-chave

Avaliação de usabilidade, computadores de mão, ensaios de interação, telefones celulares.

INTRODUÇÃO

Um importante tema que se apresenta atualmente na área de Interação Humano-Computador ressalta a importância do contexto de uso dos computadores de mão na avaliação de usabilidade destes equipamentos. Questiona-se a validade dos ensaios de interação

realizados dentro do laboratório de usabilidade convencional, pois aparentemente este não estaria apto a simular adequadamente todos os elementos pertencentes a um contexto tão dinâmico e diversificado quanto o do usuário móvel [8]. O uso de emuladores na avaliação de usabilidade também está sendo questionado por alguns autores [3,15] que não acreditam ser possível obter medidas confiáveis para a usabilidade dos computadores de mão utilizando um emulador no computador em substituição ao equipamento real. Pesquisas como [1,16,18] destacam a importância de medir a usabilidade dos computadores de mão fora do laboratório em situações mais próximas ao contexto real de uso destes equipamentos. Segundo Brewster [1] “a usabilidade dos computadores de mão não deve ser testada somente dentro do laboratório onde muitos problemas importantes com a interface podem passar despercebidos”. Entretanto, faltam referências na literatura que permitam avaliar concretamente qual o impacto da escolha do contexto de avaliação nos resultados da avaliação de usabilidade dos computadores de mão. A partir destas constatações este trabalho teve como proposta realizar um estudo comparativo com o objetivo de investigar qual a influência real exercida por diferentes contextos de avaliação nos resultados da avaliação de usabilidade dos computadores de mão baseada na técnica de ensaios de interação.

DESCRIÇÃO DO EXPERIMENTO

Os ensaios de interação foram realizados com o objetivo de avaliar a usabilidade geral da interface com o usuário de um portal de Internet móvel. Este portal disponibiliza o acesso a informações e serviços variados como programação de lazer, notícias e gerenciamento de contas de *e-mail*. Todos esses serviços podem ser consultados a qualquer momento a partir de qualquer lugar que esteja dentro da área de cobertura da operadora de telefonia móvel como aeroportos, shoppings, hotéis ou restaurantes. Não é necessário nenhum pré-requisito para acessar este serviço e qualquer pessoa que possua um telefone celular pode ser considerada um potencial cliente.

Foram definidas três abordagens distintas para a realização dos ensaios de interação a partir dos contextos de avaliação. Estes contextos diferem quanto a três aspectos: o equipamento utilizado para a navegação no portal (telefone celular ou emulador), o local de

VI Simpósio sobre Fatores Humanos em Sistemas Computacionais — Mediando e Transformando o Cotidiano. Hotel Bourbon, Curitiba, 17 a 20 de outubro de 2004. UFPR, CEIHC—SBC.

realização dos ensaios de interação (laboratório de usabilidade ou em campo) e os equipamentos utilizados para registrar a interação (computador, câmera de documentos ou câmera sem fio). As demais componentes dos contextos de avaliação como a definição da amostra de participantes, a escolha das tarefas, medidas e métricas de usabilidade foram exatamente as mesmas para as três abordagens.

Foram selecionados 36 participantes (18 homens e 18 mulheres), entre 21 e 40 anos, de diferentes formações e profissões. Os participantes foram distribuídos aleatoriamente formando três grupos homogêneos de 12 pessoas, desta forma cada um dos usuários participou de um único ensaio de interação. Todos eram usuários de telefone celular e com experiência de pelo menos um ano no uso do computador. Os participantes não haviam usado serviços de Internet móvel anteriormente e também não conheciam o modelo do telefone celular usado no experimento.

Contexto de Avaliação 1: Dentro do Laboratório Utilizando o Emulador

O primeiro grupo realizou os ensaios de interação dentro do laboratório utilizando o emulador no computador. Embora o emulador permitisse o uso do teclado o mesmo foi retirado para que o participante utilizasse apenas o mouse, desta forma a entrada de dados só era possível clicando com o mouse nos botões do telefone celular que estavam presentes na tela do computador, como em uma situação real de uso com o aparelho. Todo o sistema estava conectado, através de placa de vídeo e adaptador, a uma televisão integrada a um vídeo cassete. Desta forma, foi possível gravar em fita de vídeo toda a interação do usuário com a interface. Um microfone também foi conectado para gravar o áudio da sessão. Não havia câmeras filmando o participante, a imagem gravada em fita foi somente a da tela do computador. O avaliador se encontrava na mesma sala que o participante observando a interação através do monitor de televisão. Este contexto será referenciado como “contexto do emulador”.

Contexto de Avaliação 2: Dentro do Laboratório Utilizando o Telefone Celular e a Câmera de Documentos

O segundo grupo realizou os ensaios de interação dentro do laboratório utilizando um telefone celular fixado em um tripé posicionado sob uma câmera de documentos. O tripé foi desenvolvido especialmente para este trabalho com dimensões que permitissem um perfeito manuseio do aparelho como se o usuário o estivesse segurando nas mãos. Os graus de liberdade da base de apoio também foram cuidadosamente definidos para possibilitar o melhor ângulo de visão tanto para o usuário quanto para a filmagem da tela e do teclado do telefone (Figura 1). Entretanto o usuário não poderia mover o tripé de sua posição inicial em relação à câmera de documentos e nem alterar a posição do telefone em relação ao tripé, pois o telefone celular não deveria sair da área de visualização da câmera.

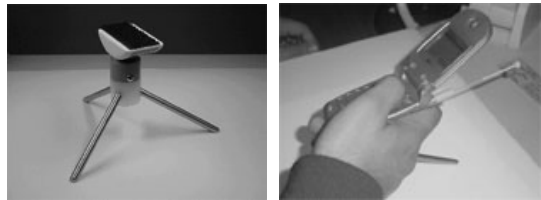


Figura 1: Tripé para fixação do telefone celular

Todo o sistema estava conectado através de placa de vídeo e adaptador do computador a uma televisão integrada a um vídeo cassete. Desta forma foi possível gravar em fita de vídeo toda a interação do usuário com a interface. Um microfone também foi conectado para gravar o áudio da sessão. Não havia câmeras filmando o participante, a imagem gravada em fita foi somente a do telefone celular. O avaliador se encontrava na mesma sala que o participante observando a interação através do monitor de televisão. Este contexto será referenciado como “contexto da câmera de documentos”.

Contexto de Avaliação 3: Fora do Laboratório Utilizando o Telefone Celular e a Minicâmera Sem Fio

O terceiro grupo realizou os ensaios de interação fora do laboratório em um ambiente movimentado e que reproduzia vários possíveis locais pertencentes ao contexto de uso deste tipo de equipamento. Para filmar a interação com o equipamento em um ambiente externo é necessária a utilização de equipamentos como câmeras e microfones móveis e portáteis e que funcionem de forma autônoma, ou seja, que não precisem estar conectados à rede elétrica durante a realização dos testes. Devido à dificuldade de encontrar no Brasil um sistema adequado e à impossibilidade de importá-lo em função do alto custo, foi desenvolvida uma solução especialmente para esta pesquisa que permitiu filmar a realização dos testes fora do laboratório. O sistema desenvolvido, inspirado em [13] possibilita a filmagem da tela e do teclado do telefone celular em qualquer ambiente, independentemente da necessidade de conexão à rede elétrica. O sistema é composto de uma minicâmera sem fio com microfone embutido que se comunica via radiofrequência com um receptor, que pode estar localizado a uma distância de até 100 metros da câmera. Estes equipamentos operam com baterias e permitem gravar a imagem colorida e também o áudio da sessão. Foi necessário desenvolver também um sistema de fixação da câmera, do telefone celular e da bateria. Este sistema foi especialmente projetado para que o usuário pudesse operar o aparelho normalmente e ao mesmo tempo fosse possível gravar as imagens da interação com qualidade. A câmera foi fixada no suporte de modo que a sua movimentação fosse solidária à movimentação do telefone (Figura 2). O suporte possui vários graus de liberdade o que permite a movimentação da câmera possibilitando vários ajustes diferentes com o objetivo de obter a imagem desejada do telefone sem prejudicar a visão do usuário e a qualidade da filmagem.



Figura 2: Sistema da minicâmera sem fio

A minicâmera envia os sinais de áudio e vídeo através de radiofrequência a um receptor conectado a uma câmera filmadora digital portátil, que também opera com bateria e grava em fita as imagens da interação (Figura 3).



Figura 3: Sistema de recepção e gravação

O participante realizou o teste sentado, segurando o sistema formado pela minicâmera e o telefone nas mãos. Como o telefone estava fixo e solidário em relação à minicâmera, o participante podia se movimentar e agir normalmente. Esta condição contribuiu para reproduzir com maior fidelidade o contexto real de uso sem prejudicar a filmagem. O avaliador estava sentado próximo ao participante observando a interação através do visor da câmera de vídeo. Não havia câmeras filmando o participante, a imagem gravada em fita foi somente a do telefone celular. Este contexto será referenciado como “contexto da câmera sem fio”.

O Emulador e o Telefone Celular

Vários modelos de telefones celulares e diversos emuladores foram pesquisados com o objetivo de encontrar o par que apresentasse o maior número possível de similaridades em termos de disposição e funcionalidade das teclas e da visualização da aplicação. Com base neste critério foram escolhidos um emulador da Openwave [14] e o software UP Simulator Openwave v.4.1.1 na versão compatível com a versão do *browser* do telefone celular selecionado. O modelo de telefone celular escolhido foi o BD4000 da LG [10] que opera usando a rede 2,5G na tecnologia CDMA 1X. A disposição e a funcionalidade das teclas para entrada de dados, seleção e navegação são rigorosamente as mesmas no emulador e no telefone celular. Os textos e menus da aplicação são apresentados na tela seguindo a mesma disposição, entretanto há duas diferenças principais em relação às duas telas: o número de linhas na tela e a barra de rolagem. O emulador apresenta na tela duas linhas a mais de texto em relação ao telefone celular e a indicação

da barra de rolagem de texto que aparece no canto direito do telefone celular não existe no emulador.

Procedimento

As avaliações foram distribuídas alternadamente entre os contextos de avaliação e realizadas ao longo de duas semanas. Cada ensaio de interação durou em média uma hora e meia e todos os testes seguiram rigorosamente o mesmo roteiro que foi elaborado a partir de [5,17]. O participante foi recebido e lhe foi explicado como seria conduzida a avaliação, o que ele teria que fazer e quanto tempo duraria a sessão. Foram mostrados ao participante todos os equipamentos utilizados no teste, e também foi explicado que seu rosto não estaria sendo filmado e que todo o procedimento seria anônimo. Após preencher um pequeno questionário sobre o seu perfil, o participante foi instruído quanto às principais funcionalidades do telefone celular ou do emulador e lhe foram demonstrados os comandos básicos para iniciar a navegação na Internet móvel. O usuário deveria realizar sete tarefas independentes umas das outras, todas relacionadas a busca de informações no portal. Foram escolhidas tarefas representativas de tarefas reais no uso do serviço de Internet móvel, que foram entregues ao participante uma de cada vez em ordem crescente de dificuldade. O avaliador observou a interação através do monitor de televisão (dentro do laboratório) e do visor da câmera filmadora (fora do laboratório) sem interferir na execução da tarefa. Todas as observações do avaliador durante a interação, bem como possíveis comentários e reações do participante foram anotados em uma ficha. Ao final de cada tarefa foi realizada uma rápida entrevista com o participante para que este comentasse possíveis dificuldades relacionadas aos incidentes de interação observados.

Ao final da avaliação foi realizada uma discussão com o participante para obter comentários gerais sobre a avaliação e esclarecimentos sobre as possíveis dificuldades que ele poderia ter enfrentado durante a interação. As sessões geraram um total de 32 horas de fita de vídeo que foram analisadas com os seguintes objetivos: medir o tempo total de duração da tarefa; analisar o resultado da tarefa executada pelo usuário para determinar se os objetivos foram ou não atingidos; identificar os problemas de interação enfrentados pelo usuário classificando-os por grau de severidade.

Critérios para Comparação

Os contextos de avaliação foram comparados a partir de dois critérios: o desempenho do usuário em cada contexto de avaliação e o desempenho do contexto de avaliação na identificação de problemas de usabilidade.

Desempenho do Usuário

O desempenho do usuário pôde ser medido a partir da definição de usabilidade definida pela ISO 9241-11 [6]: “A eficiência, a eficácia e a satisfação com as quais determinados usuários realizam determinadas tarefas em um determinado contexto de uso”. Uma vez determinadas as componentes da usabilidade por esta

definição como sendo a eficácia, eficiência e satisfação, foi utilizado o Método de Medida da Performance, parte integrante do projeto MUSiC [11], para estabelecer as definições e os métodos de medidas dessas componentes.

Eficácia

A eficácia foi medida como o percentual de objetivos corretos atingidos pelo usuário dentre o total de objetivos necessários para completar a tarefa e calculada pela fórmula:

$$\text{Eficácia} = \text{qualidade} \times \text{quantidade} / 100 \%$$

Eficiência

A eficiência foi medida como a relação entre a eficácia e o tempo total gasto pelo usuário para completar a tarefa e calculada pela fórmula:

$$\text{Eficiência} = \text{eficácia} / \text{tempo total da tarefa}$$

Satisfação

As medidas subjetivas sobre a satisfação do usuário em relação ao uso do sistema foram obtidas através do questionário SUS-System Usability Scale [2] que cobre diversos aspectos da usabilidade do sistema como complexidade, necessidade de treinamento e consistência.

Carga de Trabalho

Além das diferenças entre os equipamentos utilizados para realizar a navegação (emulador X telefone celular) e para registrá-la (computador X câmera de documentos X câmera sem fio), os usuários estavam expostos a diferentes ambientes (dentro do laboratório X fora do laboratório) e situações (local silencioso X local de movimento e barulho). Desta forma, decidiu-se também avaliar qual a carga de trabalho imposta ao participante pelo contexto de avaliação utilizando o método NASA Task Load Index [12].

Desempenho do Contexto de Avaliação

O desempenho do contexto de avaliação na identificação dos problemas de usabilidade ficou definido em função do número de problemas de usabilidade identificados, da gravidade desses problemas e do número de horas gastas na realização das avaliações.

Para poder determinar a capacidade do contexto de avaliação em identificar problemas mais graves, a cada problema de usabilidade identificado foi atribuído um grau de severidade definido em função de dois fatores: o impacto que o problema causou no usuário e a frequência com que ele ocorreu [7]. O impacto foi definido como uma escala de três pontos [4]:

- Impacto 1 – o usuário fica um pouco confuso ou aborrecido e hesita rapidamente.
- Impacto 2 – o usuário é levado a um erro do qual ele consegue se recuperar.
- Impacto 3 – o usuário é levado a um erro do qual ele não consegue se recuperar, impossibilitando-o de concluir a tarefa com sucesso.

A frequência foi definida em função do número de usuários que enfrentaram aquele problema durante a interação. Desta forma foi possível estabelecer as seguintes medidas e métricas para a análise do desempenho de cada contexto de avaliação:

- O número total de problemas de usabilidade identificados.
- A soma dos graus de severidade de todos os problemas de usabilidade identificados [7]. A partir desta métrica foi possível identificar a capacidade efetiva de cada contexto de avaliação em detectar os incidentes de interação relacionados aos graus de impacto.
- A relação benefício/custo definida como a soma dos graus de severidade dividida pelo número total de horas gastas na avaliação [7].

RESULTADOS

Desempenho do Usuário

Para comparar os resultados das três situações experimentais foi empregado o método da Análise de Variância – ANOVA. A Tabela 1 apresenta um resumo da comparação dos resultados da usabilidade medida nos três contextos de avaliação, indicando onde ocorreu uma diferença estatisticamente significativa para esses resultados.

Tabela 1:Resumo da usabilidade medida

	Câmera de documentos							Câmera de documentos							Emulador						
	X							X							X						
	Emulador							Câmera sem fio							Câmera sem fio						
	Tarefa							Tarefa							Tarefa						
	1	2	3	4	5	6	7	1	2	3	4	5	6	7	1	2	3	4	5	6	7
Eficácia	X	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=	=
Eficiência	=	X	X	=	X	X	=	=	X	=	=	=	=	=	=	=	=	=	=	=	=
Satisfação	X							X							=						
Carga de trabalho	=							=							=						

X = menores resultados no contexto da câmera de documentos

Eficácia

A análise estatística mostrou que não houve diferenças para a eficácia medida nos três contextos de avaliação com exceção da tarefa 1 que apresentou diferença entre as médias dos contextos do emulador e da câmera de documentos ($\alpha=5\%$, $F(3,32) = 5,28$, $p=0,01$). Para compreender melhor estes resultados é preciso analisar a tarefa 1. Esta tarefa tinha como objetivos encontrar o adversário, a data e o horário do próximo jogo do Internacional no Campeonato Brasileiro de Futebol. Como todos os participantes conseguiram encontrar a tela que continha essas informações a qualidade para todos foi 100%, portanto o valor da eficácia foi

determinado em função da quantidade da informação encontrada. A quantidade foi calculada como 30% para o adversário correto, 35% para a data correta e 35% para o horário correto. A Figura 4 ilustra dois exemplos de telas no emulador e no telefone celular para esta tarefa.

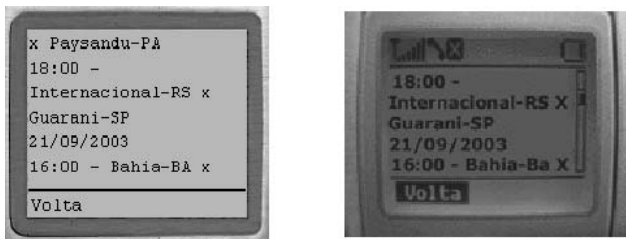


Figura 4: Telas da tarefa 1 no emulador e no telefone

Todos os participantes acertaram o adversário (Guarani), entretanto a maioria fez confusão com a data. Ao ler a informação os participantes entenderam que a data seria 21/09 quando na realidade a data correta era 20/09, que não estava mais visível na tela. A mesma confusão aconteceu com o horário do jogo quando muitos pensaram que seria 16:00h e, na realidade, o horário correto era 18:00h. No contexto do emulador 50% dos participantes encontraram todas as informações corretas, enquanto que os outros 50% acertaram o adversário e o horário, mas falaram a data errada. No contexto da câmera de documentos somente 8,3% dos participantes conseguiram encontrar todas as informações corretas, 33,3% acertaram o adversário e o horário enquanto 58,3% acertaram somente o adversário. No contexto da câmera sem fio 41,6% encontraram todas as informações corretas, 8,3% acertaram o adversário e o horário enquanto 50% acertaram somente o adversário. Essa diferença na eficácia para a tarefa 1 nos contextos do emulador e da câmera de documentos poderia ser atribuída à pequena diferença que existe entre o tamanho da tela do emulador e do telefone celular. Na tela menor do telefone o participante poderia ter uma dificuldade maior em identificar a data e o horário corretos. Entretanto, esta diferença entre os tamanhos das telas não provocou uma diferença significativa entre as médias da eficácia nos contextos do emulador e da câmera sem fio. Entre os contextos da câmera sem fio e da câmera de documentos, onde os participantes visualizavam a mesma tela do telefone celular, também não houve diferença significativa entre as médias. Os resultados da eficácia também indicaram que o índice de usuários que abandonaram as tarefas foi maior no contexto da câmera de documentos em relação aos dois outros contextos de avaliação.

Eficiência

As médias da eficiência do usuário para as tarefas 1, 4 e 7 foram estatisticamente iguais nos três contextos de avaliação ($\alpha=5\%$, $F(3,32)=1,58$, $p=0,22$; $\alpha=5\%$, $F(3,32)=1,33$, $p=0,27$; $\alpha=5\%$, $F(3,32)=3,28$, $p=0,04$ respectivamente). A eficiência do usuário foi menor no contexto da câmera de documentos em relação ao

contexto do emulador para as tarefas 2,3,5 e 6 ($\alpha=5\%$, $F(3,32)=9,91$, $p=0,0004$; $F(3,32)=9,01$, $p=0,0007$; $F(3,32)=4,27$, $p=0,022$; $F(3,32)=3,62$, $p=0,0378$ respectivamente).

A eficiência foi definida como a razão eficácia/tempo total da tarefa. Como a eficácia para as tarefas 2, 3 e 6 foi 100% nos três contextos, podemos concluir que a menor eficiência nessas tarefas decorre do fato de que no contexto da câmera de documentos os participantes levaram mais tempo para atingir os objetivos em relação aos participantes que realizaram os testes no emulador. Esta constatação é comprovada quando fazemos a ANOVA para os tempos médios nos três contextos e verificamos que o tempo no contexto da câmera de documentos é sempre maior para as tarefas 2,3 e 6 ($\alpha=5\%$, $F(3,32)=5,96$, $p=0,0061$; $F(3,32)=3,87$, $p=0,03$; $F(3,32)=5,11$, $p=0,01$ respectivamente). A menor eficiência para a tarefa 5 pode ser atribuída ao fato de que três participantes desistiram de completar esta tarefa no contexto da câmera de documentos e os que conseguiram completá-la levaram mais tempo neste contexto do que no contexto do emulador. Entre os contextos da câmera de documentos e da câmera sem fio não houve diferença significativa para a eficiência das tarefas, exceto para a tarefa 2 onde a eficiência do usuário foi menor também no contexto da câmera de documentos. Como a eficácia desta tarefa foi 100% nos dois contextos, esta diferença na eficiência pode ser atribuída ao maior tempo que os usuários levaram para executá-la no contexto da câmera de documentos em relação ao contexto da câmera sem fio ($\alpha=5\%$, $F(3,32)=5,96$, $p=0,0061$). Não houve diferença significativa entre a eficiência medida no contexto do emulador e no contexto da câmera sem fio em nenhuma das tarefas.

Satisfação

O resultado da aplicação do questionário SUS [2] para medir a satisfação do usuário em relação à interação com o sistema nos três contextos de avaliação foi de 72,92 no contexto do emulador, 58,33 no contexto da câmera de documentos e 71,88 no contexto da câmera sem fio. Os resultados da análise estatística mostraram que a satisfação do usuário em relação à interação com o sistema foi significativamente reduzida no contexto da câmera de documentos em relação aos outros dois contextos de avaliação ($\alpha=5\%$, $F(3,32)=4,60$, $p=0,01$).

Carga de Trabalho

Os testes ANOVA mostraram que não houve diferenças estatísticas para os resultados obtidos para o valor geral do NASA TLX nos três contextos de avaliação ($\alpha=5\%$, ($F(3,32)=2,40$, $p=0,10$)). Também não foi detectada diferença estatística entre os fatores isoladamente, a Demanda Mental ($\alpha=5\%$, ($F(3,32)=0,82$, $p=0,44$)), a Demanda Física ($\alpha=5\%$, ($F(3,32)=1,14$, $p=0,33$)), a Demanda Temporal ($\alpha=5\%$, ($F(3,32)=0,60$, $p=0,55$)), o Esforço ($\alpha=5\%$, ($F(3,32)=1,69$, $p=0,19$)), a Performance ($\alpha=5\%$, ($F(3,32)=2,31$, $p=0,12$)) e a Frustração ($\alpha=5\%$,

($F(3,32)=1,63, p=0,21$). Estes resultados indicam que sob o ponto de vista do usuário não houve diferença estatisticamente significativa entre os contextos de avaliação no que diz respeito à carga de trabalho.

Desempenho dos Contextos de Avaliação

Quantidade de Problemas de Usabilidade

Foram reportados um total de 43 problemas de usabilidade na interação com o sistema, distribuídos nos três contextos de avaliação conforme a Figura 5.

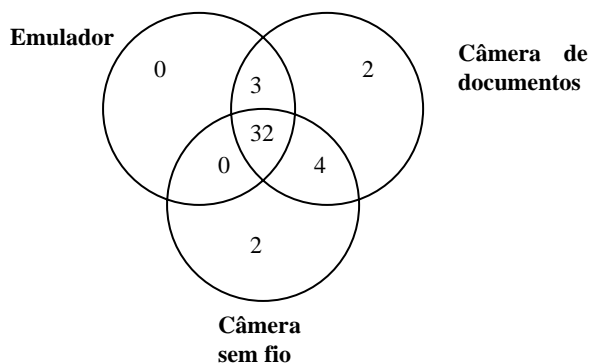


Figura 5: Distribuição dos problemas de usabilidade

Estes resultados mostram que o contexto da câmera de documentos foi o contexto no qual foi possível identificar o maior número de problemas de usabilidade, ou seja, 95,35% de todos os problemas encontrados. Este valor corresponde a 8% a mais de problemas de usabilidade identificados em relação ao contexto da câmera sem fio e 17% a mais em relação ao contexto do emulador. Considerando o número de problemas de usabilidade identificados, o contexto do emulador apresentou um resultado extremamente satisfatório uma vez que neste contexto foi possível identificar mais de 80% do total de problemas encontrados. A partir da análise da distribuição dos problemas constata-se que não houve nenhum problema de usabilidade identificado exclusivamente no contexto do emulador, entretanto os contextos da câmera de documentos e da câmera sem fio detectaram 2 problemas exclusivos cada um. Do total de 8 problemas que não foram identificados no contexto do emulador somente um deles era um problema relativo às características do aparelho de telefone celular que estava sendo utilizado e não poderia ter ocorrido no emulador. Este problema se refere à habilitação de um campo para entrada de texto. No emulador este campo já aparece habilitado enquanto que no telefone celular é necessário que o usuário pressione uma tecla antes de dar início à digitação. Como não há nenhuma indicação de que é necessário este passo a mais para habilitar o campo, os participantes ficaram bastante confusos. A interface do emulador facilitou a usabilidade neste contexto uma vez que o campo já aparece habilitado.

Severidade dos Problemas de Usabilidade

O contexto da câmera de documentos foi o contexto que totalizou o maior grau de severidade de problemas tendo

um resultado 23% superior em relação ao contexto da câmera sem fio e 59% superior em relação ao contexto do emulador. A análise da Figura 6 mostra que os problemas de usabilidade identificados ocorreram com uma frequência muito maior no contexto da câmera de documentos para todos os graus de impacto do problema.

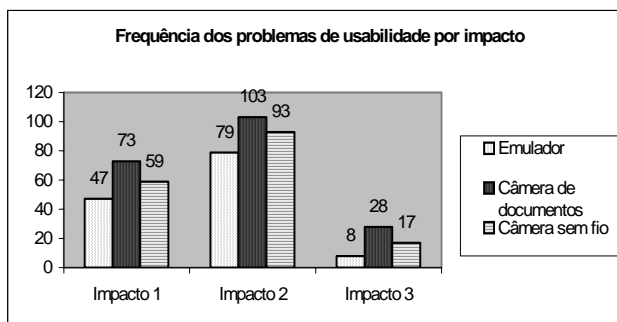


Figura 6: Frequência dos problemas de usabilidade distribuídos por impacto

A maior visibilidade dos problemas de impacto 3 no contexto da câmera de documentos é um aspecto relevante uma vez que estes são considerados os problemas mais importantes a serem detectados por serem os mais graves, pois quando eles ocorrem o usuário é incapaz de atingir todos os seus objetivos na tarefa. Desta forma eles estão relacionados aos aspectos da interface que devem ser modificados prioritariamente, antes dos demais problemas. A diferença entre a frequência de ocorrência dos problemas de impacto 3 no contexto do emulador e da câmera sem fio foi significativa, um valor de 113%, indicando que os problemas mais graves aparecem com mais frequência no telefone celular. Para os problemas de impacto 1 e de impacto 2 esta diferença foi de 26% e 18% respectivamente. Entretanto podemos considerar que apesar de aparecerem com menor frequência no contexto do emulador, este contexto foi capaz de identificar 75% dos problemas de maior impacto (impacto 2 e 3) enquanto que os contextos da câmera de documentos e da câmera sem fio identificaram 100% desses problemas.

Razão Benefício/Custo

Considerando como custo as horas despendidas pelo avaliador na realização dos ensaios de interação e na análise das fitas de vídeo, o contexto da câmera de documentos foi o que apresentou a melhor razão benefício/custo. Neste contexto foi possível identificar o maior número de problemas de usabilidade, incluindo os de maior severidade, por hora de avaliação. A análise da razão benefício/custo indicou uma vantagem maior para o contexto da câmera de documentos, sendo 82% maior em relação ao contexto do emulador e 30% maior em relação ao contexto da câmera sem fio. A Tabela 2 apresenta um resumo do desempenho das avaliações nos três contextos de avaliação.

Tabela 2: Desempenho dos contextos de avaliação

	Emulador 3°	Câmara de documentos 1°	Câmara sem fio 2°
Problemas de usabilidade	81,4%	95,35%	88,37%
Grau de severidade	229	363	296
Benefício/custo	7,49	11,38	8,79

CONSIDERAÇÕES FINAIS

De uma maneira geral, os resultados mostraram que os índices de desempenho do usuário foram mais baixos e que os problemas de usabilidade apareceram com maior frequência no contexto da câmera de documentos. Estes resultados não podem ser atribuídos somente à diferença entre as interfaces (emulador X telefone celular) ou ao local de realização dos ensaios de interação (dentro do laboratório X em campo), caso contrário estas diferenças poderiam ter aparecido também quando foram comparados os resultados dos contextos da câmera sem fio e do emulador, o que não ocorreu. Estes dados sugerem que o fato de ter que operar o telefone que estava fixo em um tripé móvel, mas que não podia ser movimentado pelo usuário, sob pena de sair da área de visualização da câmera de documentos, provocou um desconforto e foi uma fonte extra de preocupação para o participante. Esta tarefa cognitiva concorrente acabou por influenciar o resultado dos testes.

O contexto do emulador é a situação de avaliação mais simples de ser executada uma vez que não exige equipamentos especiais para registrar a interação e não há a necessidade de utilizar o equipamento real no qual a aplicação será utilizada. Entretanto o uso do emulador coloca uma restrição na escolha da amostra de participantes, uma vez que é preciso que o participante já seja usuário de computador. O treino no uso do mouse e de outros dispositivos para a interação é importante, caso contrário as dificuldades com o uso do computador serão as principais enfrentadas pelo usuário, impossibilitando a avaliação do diálogo com a interface. De uma maneira geral, o contexto do emulador obteve ótimos índices de desempenho na identificação de problemas de usabilidade, embora neste contexto estes problemas tenham aparecido com uma frequência muito menor em relação aos outros dois contextos de avaliação. Estes dados sugerem, como constatado por [9] que os principais problemas de usabilidade podem ser identificados em ambientes mais simples de avaliação dentro do laboratório. Entretanto a pertinência dos problemas de usabilidade identificados no contexto do emulador depende da similaridade entre a interface do emulador e a interface do telefone real utilizado. As pequenas diferenças existentes nas interfaces escolhidas neste experimento mostraram que uma leve mudança na interface do emulador, como um maior número de linhas por exemplo, pode causar uma repercussão muito grande na usabilidade verificada, para melhor ou para pior.

A solução da minicâmara sem fio para registrar a avaliação no ambiente externo foi extremamente

satisfatória uma vez que foi capaz de gerar imagens de alta qualidade sem prejudicar o desempenho do usuário. De uma maneira geral, os resultados da usabilidade medida no contexto da câmera sem fio estiveram próximos aos resultados obtidos nos outros dois contextos de avaliação, pois não houve diferença significativa na comparação desses resultados. O ambiente externo era bastante barulhento e movimentado, entretanto os fatos do usuário saber que participava de uma avaliação, do avaliador estar presente e da utilização dos equipamentos para registrar a interação, fizeram com que o participante aumentasse a concentração nas tarefas que estava executando e não estivesse tão suscetível às interferências externas como era de se esperar. Estes dados sugerem que a pesquisa de novas técnicas que possibilitem ao usuário utilizar um computador de mão em situações reais, ou seja, em movimento, em vários ambientes diferentes, estando exposto às interferências externas e sem a presença do avaliador, ao mesmo tempo em que permitam registrar a interação são importantes temas para a realização de trabalhos futuros.

Embora este experimento tenha utilizado três abordagens representativas para as avaliações de usabilidade dos computadores de mão seus resultados são limitados em vários aspectos. Primeiramente os usuários não estavam expostos a tarefas concorrentes explícitas. Os resultados obtidos nesta pesquisa mostraram que uma tarefa cognitiva concorrente, por menor que ela seja, como a preocupação em manter o tripé no foco da câmera de documentos, pode impactar negativamente sobre a usabilidade medida. O segundo aspecto se refere à posição dos participantes, que realizaram os ensaios de interação sentados e não em pé, como poderia configurar um outro possível contexto real de uso. Finalmente, os resultados aqui alcançados sugerem que este experimento seja repetido utilizando a câmera sem fio e o telefone celular dentro do laboratório e em campo para melhor avaliar o efeito destes dois contextos nos resultados das avaliações de usabilidade.

REFERÊNCIAS

1. Brewster, S. *Overcoming the Lack of Screen Space on Mobile Computers*. Personal and Ubiquitous Computing 2002; 6 (3):188-205
2. Brooke, J. *SUS – A Quick and Dirty Usability Scale*. In: Jordan, P.W. et al., Usability Evaluation in Industry. Taylor & Francis, London, UK 1996; 189-94
3. Chittaro, L., Cin, P. *Evaluating Interface Design Choices on WAP Phones: Single-Choice List Selection and Navigation Among Cards*. Proc. Mobile HCI 2001
4. Desurvire, H. W. *Faster, Cheaper!! Are Usability Inspection Methods as Effective as Empirical Testing?* In: Nielsen, J., Mack, R.L. Usability Inspection Methods. John Wiley & Sons, USA 1994; 173-202
5. Dumas, J.S., Redish, J.C. *A Practical Guide to Usability Testing*. England: Intellect, 1999

6. ISO 9241-11 – Ergonomic Requirements for Office Work with Visual Display Terminals (VDT)s – Part 11: Guidance on Usability 1998
7. Jeffries, R., Miller, J., Wharton, C., Uyeda, K. *User Interface Evaluation in the Real World: a Comparison of Four Techniques*. Proc. CHI 1991
8. Johnson, P. *Usability and Mobility: Interactions on the Move*. First Workshop on Human Computer Interaction with Mobile Devices 1998
9. Kjeldskov, J., Skov, M. *Creating Realistic Laboratory Settings: Comparative Studies of Three Think-Aloud Usability Evaluations of a Mobile System*. Interact 2003
10. LG Corporate web site: BD4000 Mobile Phone Characteristics. Disponível em: <http://www.lge.com.br/produtos/> (Fevereiro, 2004)
11. Macleod, M., Bowden, R., Bevan, N., Curson, I. *The MUSiC Performance Measurement Method*. Behaviour & Information Technology 1997; 16 (4/5): 279-293
12. NASA TLX - NASA Task Load Index. Human Performance Research Group, NASA Ames Research Center. Moffett Field, California, USA 1986
13. Nyysönen, T., Roto, V., Kaikkonen, A. *Mini-camera for Usability Tests and Demonstrations*. Demonstrated at Mobile HCI 2002
14. Openwave Developer Network web site: Tools and SDK. Disponível em: <http://developer.openwave.com/dvl/> (Fevereiro, 2004)
15. Pearrow, M. *The Wireless Web Usability Handbook*. Hingham, MA: Charles River Media, 2002
16. Petrie, H., Furner, S., Strothotte, T. *Design Lifecycles and Wearable Computers for Users with Disabilities*. First Workshop on Human-Computer Interaction with Mobile Devices, 1998
17. Rubin, J. *Handbook of Usability Testing: how to plan, design, and conduct effective tests*. USA: John Wiley & Sons, Inc., 1994.
18. Waterson, S., Landay, J.A., Matthews, T. *In the Lab and Out in the Wild: Remote Web Usability Testing for Mobile Devices*. Proc. CHI 2002

Avaliação heurística participativa do CollabSS

Marcos Augusto F. Borges
Instituto de Computação – Unicamp
R. Barão de Paranapanema, 400, ap. 112,
Bosque, Campinas – SP – Brasil, 13026-010
marcosborges@ig.com.br

M. Cecilia C. Baranauskas
Instituto de Computação – Unicamp
Caixa Postal 6176 – 13083-970 –
Campinas – SP – Brasil
cecilia@ic.unicamp.br

RESUMO

Ambientes CSCL comumente provêm uma ferramenta para interação síncrona com base em textos (chat), por meio da qual os usuários podem colaborar. A colaboração entre os usuários normalmente é conduzida por um facilitador da colaboração, que possui a responsabilidade de manter o grupo trabalhando em uma atmosfera colaborativa. CollabSS é um sistema desenvolvido para apoiar o facilitador a analisar e promover a colaboração entre os participantes no curso de uma atividade colaborativa. Foi conduzida uma avaliação de usabilidade do CollabSS baseada em uma adaptação da técnica de avaliação heurística participativa. Este trabalho descreve como foi conduzida essa avaliação e como ela se refletiu no sistema.

ABSTRACT

CSCL environments usually provide a chat tool embedded in the interface, through which the users can collaborate. The collaboration among users is commonly conducted by a collaboration facilitator. This facilitator has the responsibility of keeping the group working in a collaborative learning atmosphere. CollabSS is a tool developed to support the facilitator in analyzing and promoting the collaboration among the users during the course of a collaborative activity. A usability evaluation of CollabSS was carried on based on a customization of the participatory heuristic evaluation technique. This paper describes how the evaluation was prepared and conducted and also presents the new features of CollabSS, derived from evaluation.

Palavras chaves

Avaliação heurística participativa.

INTRODUÇÃO

Um facilitador da colaboração pode ser definido como um agente da interação responsável por manter uma atmosfera colaborativa entre os agentes interagindo. CollabSS é um sistema para apoiar o facilitador da colaboração na identificação de momentos em que seja importante agir e para sugerir como deve ser essa ação. O projeto do sistema CollabSS foi elaborado para que ele possa trabalhar em conjunto com qualquer ferramenta para comunicação mediada por computador (CMC)

VI Simpósio sobre Fatores Humanos em Sistemas Computacionais — Mediando e Transformando o Cotidiano. Hotel Bourbon, Curitiba, 17 a 20 de outubro de 2004. UFPR, CEIHC—SBC.

baseada em texto. CollabSS implementa o *framework* FAnC para análise de interações.

De forma a analisar o design do CollabSS, foi conduzida uma Avaliação Heurística Participativa com uma versão prévia do sistema. Com base nos resultados obtidos, houve um *re-design* do CollabSS, resultando em uma nova versão. Como o CollabSS é um sistema que acompanha a interação que está ocorrendo em um sistema colaborativo externo, utilizou-se o Jogo da Fábrica como sistema externo para a condução dessa avaliação.

Este trabalho é organizado da seguinte forma: inicialmente é apresentado o Jogo da Fábrica para contextualizar-se a avaliação. Segue-se uma apresentação resumida do FAnC. Apresenta-se então a versão avaliada do CollabSS. Posteriormente, discute-se a técnica de avaliação heurística participativa utilizada. Após é apresentada uma lista com as oportunidades para melhorias detectadas no CollabSS e discute-se como essas oportunidades influenciaram na última versão do sistema. Por fim, são apresentadas as conclusões do trabalho.

O JOGO DA FÁBRICA

O Jogo da Fábrica é um sistema para aprendizado colaborativo apoiado por computador (*CSCL-computer supported collaborative learning*), que aborda o uso de jogos para aprendizagem em um contexto de uso para formação de profissionais em empresas [1][2]. No Jogo da Fábrica os aprendizes podem simular o processo de produção de uma fábrica hipotética. Esse jogo estende trabalhos prévios do grupo DAFE¹ que estudavam o uso de modelagem e simulação e da estética construcionista de aprendizagem em jogos computacionais para apoio à aprendizagem de conceitos de manufatura por trabalhadores do chão-de-fábrica, baseando-se em atividades individuais onde um sujeito interage com um computador.

O Jogo da Fábrica foi construído para uma investigação sobre o uso de um sistema com estética construcionista para CSCL na formação contínua de profissionais de empresas de manufatura. O Jogo da Fábrica tem como objetivo apoiar o aprendizado de conceitos de manufatura sincronizada, como *lead time*, *just in time*, *Kanban*, identificação de gargalos, etc. Mais importante que

¹ Grupo de Dinamização da Aprendizagem e Formação em Empresas do Núcleo de Informática Aplicada a Educação (Nied) da Unicamp (<http://www.nied.unicamp.br/~dafe>)

viabilizar o aprendizado dos conteúdos de manufatura, o jogo é um sistema para auxiliar o funcionário a aprender como trabalhar em grupo buscando um objetivo comum. O Jogo da Fábrica está alinhado aos objetivos das organizações atuais, incentivando os trabalhadores a serem mais ativos nos processos de produção, a interagirem e a colaborarem entre si, utilizando conhecimento e experiência de um modo criativo para detectar problemas e para sugerir e discutir soluções.

O Jogo da Fábrica estende a arquitetura previamente utilizada em outros projetos do grupo DAFE para um sistema síncrono multi-usuário que utiliza a Internet como mídia de comunicação, tendo a interação e a colaboração à distância como parte essencial do sistema. Todas as interações que ocorrem na ferramenta de interação textual síncrona, bem como as configurações definidas para as jogadas, podem ser gravadas para uso em análises posteriores e para reflexão dos envolvidos. A arquitetura foi construída para o atendimento de requisitos não funcionais como velocidade de comunicação e confiabilidade do sistema.

O Jogo da Fábrica herda dos trabalhos anteriores do DAFE, como o Enxuto [3] e o Jonas [4] as seguintes características:

- explora situações do contexto real da fábrica;
- encoraja os usuários a arriscar em ambiente sendo seguro, uma vez que o usuário pode errar e aprender com os erros, sem risco de causar prejuízos para a empresa;
- o coordenador (que incorpora os papéis do facilitador da colaboração e do aprendizado) confunde-se com os outros usuários como um dos participantes do jogo.

A interface de usuário do Jogo da Fábrica apresenta oito células de produção distribuídas em formato de “U” ao redor da janela. Cada célula tem posições definidas para matéria prima e material produzido. Um fator aleatório insere uma variabilidade na produção. Ao centro, uma ferramenta para interação textual síncrona é disponibilizada para a comunicação entre os usuários. A Figura 1 ilustra o design do sistema.

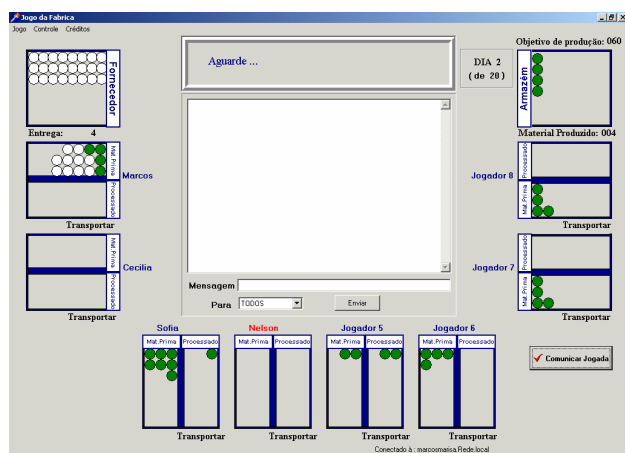


Figura 1. Jogo da Fábrica

Cada célula de produção é controlada por um jogador ou pelo sistema. Cada jogador é responsável por fazer a “produção” e a “transferência” de produtos prontos de sua célula para a próxima. O conjunto de todos os jogadores fazendo essas ações pode ser visto como o funcionamento da linha de produção. É esperado que os jogadores interajam de forma colaborativa enquanto a produção vai sendo simulada.

O Jogo da Fábrica possibilita a participação de usuários que não estejam fisicamente próximos uns dos outros. Uma das vantagens do uso de ambientes virtuais é a possibilidade de poder envolver pessoas de diferentes departamentos ou fábricas, além de possibilitar a presença virtual de um instrutor/facilitador. Além disso, o sistema pode ser usado por um número variável de jogadores (um a oito). Isto é importante porque, para um contexto real de fábrica, nem sempre está disponível um número grande de funcionários da empresa. No caso de menos de oito jogadores envolvidos, as células restantes são controladas automaticamente pelo próprio sistema.

No Jogo da Fábrica, todos os jogadores têm a mesma visualização da linha de produção da fábrica. Com esta visão eles conseguem identificar a situação da linha de produção, analisando visualmente o desempenho e os problemas. Um dos jogadores desempenha o papel de coordenador. O coordenador controla uma célula, similarmente aos outros jogadores, mas também desempenha algumas atividades específicas no jogo. A configuração da fábrica e a definição das metas são feitas previamente pelo coordenador. Entre os parâmetros de configuração definidos pelo coordenador estão o número de dias de produção, a meta de produção, o sistema de produção (“puxar” ou “empurrar”), o tamanho do lote de transferência (*Kanban*) e os limites de capacidade das máquinas (a capacidade em um dia específico é definida aleatoriamente pelo sistema dentro desses limites). O coordenador também é o responsável por definir o início e o fim de cada jogada e o andamento do jogo, além da atribuição de horas extras aos agentes.

O FRAMEWORK FANc

O FANc é um *framework* conceitual para analisar uma interação textual, de forma a possibilitar o estudo de: a natureza dos atos de fala que ocorrem durante a atividade, o papel dos agentes envolvidos e a dinâmica das interações.

Em uma colaboração ocorrendo em um sistema computacional, podem existir diversas formas para os agentes interagirem, que não se limitam a comunicação por meio de ferramentas para interação textual. O estudo de todas as possíveis formas de comunicação pode trazer elementos extras em uma análise, mas o objetivo do FANc é estudar exclusivamente a colaboração que ocorre por meio de ferramentas computacionais para interação textual.

O FANc é composto por um conjunto de definições, um conjunto de representações e um conjunto de heurísticas. As definições conceituam o que é importante em uma análise de interação. As representações ilustram graficamente o que está acontecendo em uma interação,

com base nas definições. As heurísticas propõem formas para obter informações sobre colaboração com base nas interações e para intervir na interação com objetivo de manter o estado colaborativo entre os agentes.

O FAnC define “fala” como cada mensagem individual enviada por um agente. As falas podem ser classificadas em três dimensões:

- emissor: o agente que criou a ação de influência;
- área de influência: os agentes no campo de influência;
- classe: o tipo de informação que ela carrega.

O objetivo de avaliar os agentes que criam e que sofrem a influência é identificar a diferença entre os papéis desempenhados por eles em uma dada interação.

Conforme indicado por [5], o tipo de informação veiculado pela fala é importante para que se possa analisar o tipo de interação em uma atividade e, portanto, para que se possa analisar o quanto a atividade é colaborativa. O FAnC define um padrão para a construção de classificações. Para se adequar às heurísticas e representações do FAnC, uma classificação deve ser feita de forma que cada fala seja identificada como sendo de apenas uma classe (ou seja, as classes devem ser mutuamente exclusivas). A próxima seção apresenta um exemplo de classificação de falas para um determinado contexto.

Com base na definição de falas apresentada anteriormente, define-se “conversa” como uma seqüência de falas relacionadas. Uma conversa pode ser classificada de acordo com as seguintes características:

- iniciador: o agente que a iniciou, ou seja, o emissor da primeira fala;
- área de influência: a união das áreas de influência do conjunto de falas que compõe a conversa;
- classe de origem: a classe da primeira fala;
- classe da conversa: considerada como a classe da maioria das falas;
- tamanho: o número de falas; com base no tamanho da conversa, define-se “conversa unitária” como uma conversa composta por uma única fala, conversa pequena como sendo uma conversa com poucas falas e conversa grande como sendo uma conversa com muitas falas;
- paralelismo: se ocorre em paralelo com outras conversas;
- duração da conversa: tempo entre a primeira e a última fala.

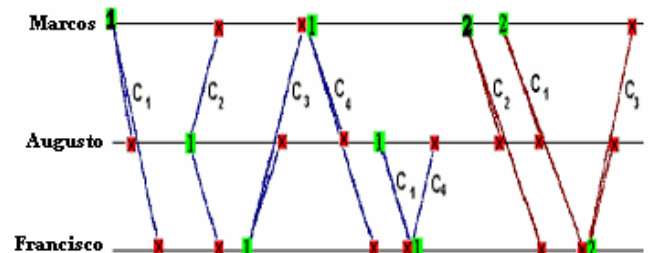
FAnC propõe uma representação básica para servir como instrumento de análise: o Gráfico de Conversas. O Gráfico de Conversas representa o momento que acontece uma fala, o agente que a criou, a área de influência da fala e as conversas. No Gráfico de Conversas, cada usuário é representado em uma linha horizontal. O transcorrer do tempo é representado horizontalmente, da esquerda para a direita. Linhas diagonais representam as falas, iniciando-se à esquerda na linha correspondente ao agente emissor,

na posição correspondente ao tempo em que a fala foi enviada e apontando à direita para uma ou mais linhas que representam a área de influência da fala. Havendo mais de um agente na área de influência, esse fato é representado como várias linhas diagonais que têm o mesmo ponto de origem. No início de cada fala pode-se representar a conversa a que ela pertence com um número, apresentado em negrito na primeira fala da conversa. Ao lado da linha diagonal que representa a fala, pode ser apresentado o código da sua classe.

A Figura 2 apresenta um exemplo de Gráfico de Conversas. Nessa ilustração há três agentes interagindo (Clóvis, Valente e Marcos). O gráfico mostra duas conversas (1 e 2), cada uma composta por um conjunto de falas. A conversa 1 começa com Clóvis enviando uma fala da classe C1 para os outros dois agentes. Em seguida, Valente envia uma fala da classe C2 para os outros dois agentes e assim a interação continua. Nesse exemplo, a conversa 1 é composta por seis falas e a conversa 2 por três.

A estrutura linear do Gráfico de Conversas do FAnC torna possível:

- analisar-se o perfil da interação em um momento de tempo específico, podendo-se deslocar a análise no tempo com facilidade;
- representar interações longas e com densidade alta de mensagens sem aumentar-se a complexidade da representação.



Marcos fala para TODOS: Devemos aumentar a capacidade das máquinas?

Augusto fala para TODOS: Eu acho que a capacidade é o suficiente.

Francisco fala para TODOS: Eu acho que precisamos aumentar.

Marcos fala para TODOS: Também acho que precisamos aumentar.

Augusto fala para Francisco: Por que você acha que precisa aumentar.

Francisco fala para Augusto: Não foi por isso que não atingimos a meta?

Marcos fala para TODOS: Acho que já discutimos o suficiente.

Marcos fala para TODOS: Já está na hora do almoço...

Francisco fala para TODOS: Podemos então terminar a discussão?

Figura 2. Um exemplo de Gráfico de Conversas (acima) e texto da interação correspondente (abaixo)

Com o FAnC, objetivou-se oferecer um ferramental para a análise da interação com o objetivo de identificar a existência ou não de aspectos de colaboração. Indicativos de colaboração podem ser obtidos por meio de heurísticas que podem apoiar a análise da colaboração e apoiar um facilitador no transcorrer da atividade. Um conjunto de heurísticas que independe da classificação das falas adotada na análise é apresentado abaixo. Heurísticas para uma classificação específica também podem ser propostas. As heurísticas propostas, independentes da classificação das falas, são:

- Densidade das falas: número de falas por unidade de tempo. Uma densidade mínima é um índice necessário para a colaboração, mas não suficiente. Para que haja colaboração é realmente necessária uma quantidade mínima de falas por unidade de tempo, mas essas falas, se não fizerem parte de uma conversa, podem ser falas desconectadas que pouco apoiam a colaboração;
- Tamanho das conversas: quanto maiores as conversas, maior é a chance de elas fazerem parte de uma atividade colaborativa. Os agentes que participam de grandes conversas são aqueles com maior probabilidade de estarem trabalhando colaborativamente. Esta heurística é útil também na identificação de agentes que podem desempenhar o papel de facilitador entre aqueles envolvidos em uma interação: agentes que iniciam um certo número de conversas, especialmente conversas grandes, são os que, provavelmente, têm maior potencial para apoiar a colaboração;
- Oportunidade de apoio à colaboração: é um momento onde o facilitador deve intervir buscando aumentar a colaboração no grupo. Quando um agente ou todo o grupo diminui sua participação, pode ser considerado como um exemplo de oportunidade de apoio à colaboração. Um facilitador, tendo identificado uma oportunidade como essa, deve intervir imediatamente no processo, buscando motivar os agentes a ter uma maior participação. Caso contrário, o estado colaborativo pode se deteriorar;
- Iniciador de conversa: é um tipo de fala com grande potencial de gerar conversas longas e colaborativas. Entre esses tipos, por exemplo, estão falas pedindo explicitamente para que um certo agente se envolva mais nas discussões.

Um facilitador deve buscar identificar com precisão oportunidades de apoio à colaboração, enviando à interação um iniciador de conversa que seja adequado a essa oportunidade. Dessa forma, ele poderá criar discussões com maiores possibilidades de gerar colaboração.

COLLABSS: UM SISTEMA DE APOIO À COLABORAÇÃO

CollabSS [6] possui um agente que acompanha a interação e analisa os dados, identificando oportunidades de apoio à colaboração.

Quando uma oportunidade é identificada pelo CollabSS, ele apresenta ao facilitador um botão com uma lâmpada

na janela principal (Figura 3). Pressionando esse botão, o sistema apresenta uma janela para envio de iniciador de conversa (Figura 4). Essa janela apresenta, no campo de mensagens, um iniciador de conversas. O facilitador pode alterar a mensagem, ignorá-la ou enviá-la a todos os agentes da interação ou, quando a oportunidade detectada for relacionada a um agente específico, a este agente.

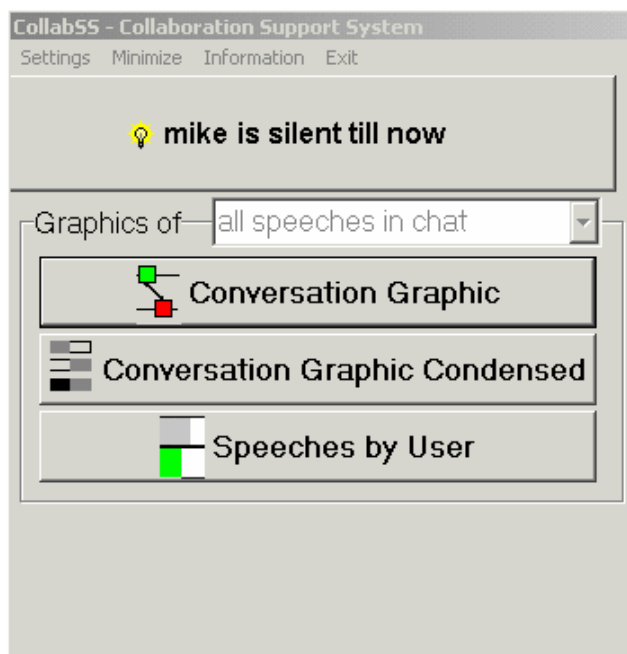


Figura 3. Tela principal do CollabSS com oportunidade de apoio à colaboração identificada.

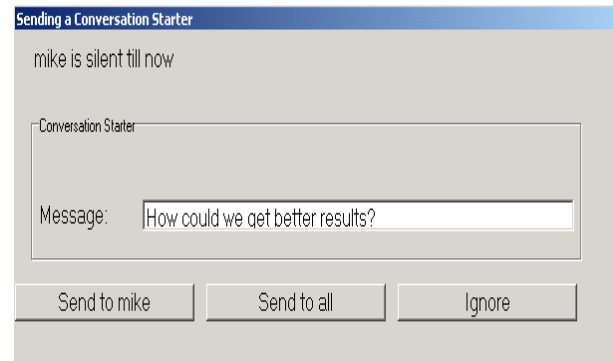


Figura 4. Tela de envio de iniciador de conversa

CollabSS também provê representações da interação alteradas dinamicamente no decorrer da própria interação. Essas representações almejam apoiar o facilitador na análise: do curso das conversas, da forma como os agentes estão interagindo entre si e de como está o número de falas por unidade de tempo, por usuário e por conversa. Essas representações gráficas estão acessíveis a partir da janela principal do CollabSS. O facilitador pode usar o gráfico de conversas (Figura 5), o gráfico condensado de conversas (Figura 6) e o gráfico de barras com o número de falas por usuário (Figura 7).

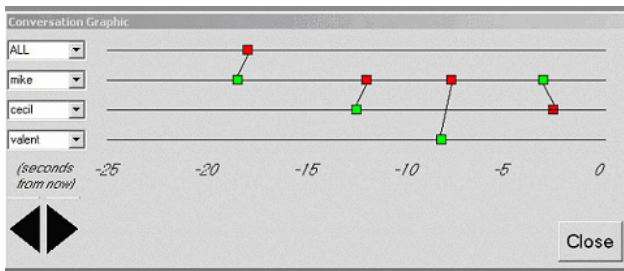


Figura 5. Tela do gráfico de conversas.

Usando o gráfico de conversas (Figura 5) o facilitador pode analisar a participação dos agentes, a densidade de falas em um curto intervalo de tempo e o padrão das conversas. As linhas horizontais representam os agentes. No curso da interação, o facilitador pode selecionar os agentes que serão representados no gráfico. A linha do tempo aumenta horizontalmente da esquerda para a direita: na Figura 5, a extrema direita é o momento em que a representação está sendo apresentada e o gráfico apresenta até 25 segundos antes disso. O gráfico tem a possibilidade de deslocamento horizontal, habilitando uma análise de toda a interação, desde sua primeira fala. Cada fala é apresentada como uma linha diagonal, tendo o iniciador representado pelo quadrado à esquerda e a área de influência pelo(s) quadrado(s) à direita dessa linha.

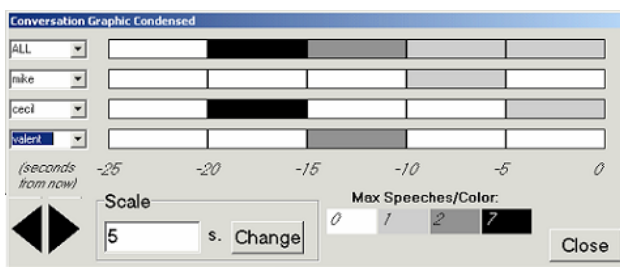


Figura 6. Tela do gráfico condensado de conversas.

Usando o gráfico condensado de conversas (Figura 6) o facilitador é capaz de analisar a participação dos agentes e a densidade das falas em um intervalo maior que no gráfico de conversas. A densidade é representada com cores em retângulos, na linha do agente iniciador, no intervalo de tempo identificado. A escala de tempo e o próprio número de escalas pode ser configurado de forma a representar maiores ou menores intervalos.

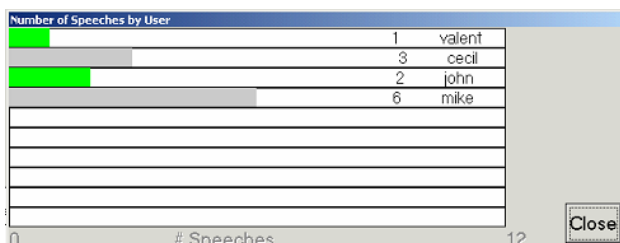


Figura 7. Tela de falas por usuário.

A janela de falas por usuário (Figura 7) mostrou o número de mensagens cada usuário enviou à interação. Ela pode ser útil na avaliação de quem está mais ativo e quem necessita maior apoio por parte do facilitador.

AVALIAÇÃO HEURÍSTICA PARTICIPATIVA

A avaliação heurística participativa (AHP) [7] é uma técnica de inspeção participativa derivada da avaliação heurística (AH) de Nielsen [8]. A AHP estende a AH de dois modos:

- a AHP adiciona heurísticas orientadas ao processo na lista de heurísticas usadas para guiar o inspetor na AH. Estas heurísticas visam inspecionar como o artefato computacional situa-se no contexto em que será usado;
- a AHP adiciona usuários especialistas no domínio na lista de inspetores da AH.

Para se avaliar o CollabSS em uso, é necessário que ele esteja acompanhando uma interação ocorrendo em um sistema externo com interação baseada em ferramenta para interação textual. Por esse motivo, a avaliação do CollabSS foi feita em uma atividade que envolveu o Jogo da Fábrica.

A dinâmica foi conduzida em um laboratório do Nied, em janeiro de 2003, envolvendo oito participantes: professores, alunos de pós-graduação e pesquisadores da área de informática e educação. Os participantes que fizeram o papel de coordenadores do Jogo da Fábrica foram os usuários do CollabSS na atividade. Participaram como inspetores uma professora universitária da área de engenharia, representando um especialista no domínio da aplicação e três estudantes de pós-graduação com conhecimentos em interfaces e sistemas para aprendizado. A professora e um dos estudantes coordenaram individualmente (Figura 8), cada um, uma rodada no Jogo da Fábrica. Os outros dois estudantes trabalharam conjuntamente: enquanto um coordenava a atividade no Jogo da Fábrica, o outro, ao lado, buscava especificamente apoiar a colaboração com o CollabSS (Figura 9).



Figura 8. Um avaliador durante a AHP.



Figura 9. Dois avaliadores durante a AHP.

A avaliação iniciou-se com uma breve explicação do Jogo da Fábrica e dos conceitos envolvidos. Um texto explicativo, que descreve as funcionalidades do Jogo da Fábrica, foi fornecido para apoiar os avaliadores. Antes do início da avaliação, o autor apresentou aos envolvidos que iriam desempenhar o papel de coordenadores do Jogo da Fábrica uma breve introdução ao CollabSS. Os coordenadores do Jogo da Fábrica usaram o CollabSS como sistema de apoio ao papel de facilitadores, assumindo então o papel de avaliadores do CollabSS. Como o Jogo da Fábrica usa uma tela inteira, o CollabSS foi instalado em outro computador, colocado ao lado (Figuras 8 e 9).

Seguiu-se a experimentação com o jogo. Ao final das jogadas, o autor entregou a cada um dos participantes uma versão traduzida das heurísticas propostas pela AHP (apresentada na próxima seção) para inspeção da interface. Cada avaliador teve dez minutos para resumir individualmente os possíveis problemas de usabilidade descobertos. Por fim, todos os avaliadores combinaram seus resultados individuais em uma lista de problemas construída por todos, agregando a ela também um conjunto de recomendações. A lista, as recomendações e seu resultado no *design* do sistema são apresentados na seção “Resultado da Avaliação”.

TEXTO ENTREGUE AOS AVALIADORES

Abaixo é apresentado o texto entregue aos avaliadores.

“Este processo irá fazer parte de uma avaliação heurística participativa da usabilidade dos sistemas. As pessoas que forem coordenadores estarão avaliando o CollabSS. Os outros participantes irão avaliar o Jogo da Fábrica.

Nesta avaliação você é livre para discutir todos os pontos que achar importante. É fornecida abaixo uma lista de heurísticas que podem ser usadas como apoio a sua análise. Não é uma lista completa nem exclusiva: podem ser levantadas questões não citadas por ela. Além disso, podem existir heurísticas que não sejam adequadas para avaliar o sistema em questão (neste caso devem ser ignoradas).

Ao final dos experimentos, cada participante terá 5 minutos para preparar uma lista de questões

individualmente. Depois, os grupos de avaliação devem se unir e construir em 5 minutos uma lista única de problemas de usabilidade descobertos. Pode ser criada também uma lista de recomendações para os desenvolvedores.

Heurísticas

1. Status do sistema: o sistema mantém os usuários informados e no momento adequado sobre o que está acontecendo.
2. Seqüência das tarefas: usuários, quando apropriado, podem selecionar e seqüenciar as tarefas ao invés de o sistema ter o controle sobre as ações dos usuários. Wizards estão disponíveis mas são opcionais e estão sob o controle do usuário.
3. Saídas de emergência: usuários podem
 - a. Encontrar facilmente saídas de emergência quando escolhem uma opção por engano (uma saída de emergência permite que o usuário saia de um estado não desejado sem ter que passar por uma grande quantidade de passos).
 - b. Fazer suas próprias decisões (com informação e feedback claros) com relação aos problemas relacionados com abandonar uma certa atividade.
 - c. Acessar operações de desfazer e refazer
4. Flexibilidade e eficiência de uso: aceleradores estão disponíveis para especialistas, mas são transparentes para novatos. Usuários são capazes de configurar ações freqüentes. São disponibilizados meios alternativos de acesso e operação para usuários que não estão no grupo de usuários típicos (ex. em habilidade física e/ou cognitiva, cultura, língua)
5. Mapeamento entre o sistema e o mundo real: o sistema fala a língua do usuário, com palavras, frases e conceitos familiares para o usuário, ao invés de termos orientados ao sistema. As mensagens são baseadas no mundo real do usuário, aparecendo em uma ordem lógica e natural.
6. Consistência e padronização: cada palavra, frase ou imagem no design é usada com consistência, tendo sempre o mesmo significado. Cada objeto de interface ou operação no computador é sempre designado usando a mesma palavra, frase ou imagem. O sistema segue as convenções da plataforma em que é usado.
7. Reconhecimento e não memorização: os objetos, ações e opções são visíveis. O usuário não precisa memorizar informações enquanto transita de um ponto para outro do sistema. As instruções para uso do sistema estão visíveis ou facilmente recuperáveis quando apropriado.
8. Estética e design minimalista: os diálogos não contêm informações que são irrelevantes ou

raramente necessárias (informações extras em um diálogo competem com unidades de informação relevantes e diminuem a visibilidade relativa das últimas)

9. Ajuda e documentação: o sistema é intuitivo e pode ser usado nas tarefas mais comuns sem documentação. Quando necessário, a documentação é facilmente encontrada, apóia o usuário na tarefa, apresenta os passos que devem ser seguidos e tem o tamanho adequado para a tarefa do usuário. Documentos grandes são suplementados com múltiplos meios de encontrar seus conteúdos (tabela de conteúdos, índices, buscas, etc.)
10. Ajuda no reconhecimento dos usuários, diagnóstico e recuperação de erros: as mensagens de erro indicam precisamente o problema e sugerem uma solução construtivamente. Elas estão expressas em linguagem coloquial (sem códigos). Usuários não são criticados pelos erros.
11. Prevenção de erros: melhor que boas mensagens de erros é um design que as evite. Os erros são antecipados e o sistema trata o erro ou como uma entrada inválida ou como uma entrada ambígua que precisa ser melhor definida.
12. Habilidades: o sistema apóia, estende, suplementa ou melhora as habilidades do usuário, o conhecimento prévio e a experiência. O sistema não substitui o usuário. Wizards apóiam, estendem ou executam decisões tomadas pelo usuário.
13. Interação com usuário respeitosa e agradável: as interações do usuário com o sistema melhoram a qualidade da experiência de usar o sistema. O usuário é tratado com respeito. O design reflete o papel profissional do usuário, a identidade pessoal ou a intenção. O design é esteticamente agradável com um balanço apropriado entre os valores artísticos e funcionais.
14. Trabalho de qualidade: o sistema apóia o usuário na entrega de trabalho de qualidade para seus clientes (quando apropriado). Os atributos de um trabalho de qualidade incluem entrega no prazo, acuracidade, apelo estético e níveis adequados de completude.
15. Privacidade: o sistema ajuda o usuário a proteger informações pessoais ou privadas, que pertencem ao usuário ou a seus clientes.”

RESULTADO DA AVALIAÇÃO

Analisou-se o resultado da avaliação de modo a verificar a possibilidade de solucionar problemas e implementar sugestões. São apresentados a seguir os problemas levantados, o que foi sugerido pelo grupo e como essas informações resultaram em melhorias no sistema:

- indicação: o sistema originalmente apresentava a interface apenas em inglês e a avaliação foi conduzida com agentes brasileiros;

melhoria: foi projetada uma arquitetura de internacionalização e, no momento, a interface está disponível em português e inglês;

- indicação: foi solicitada a implementação de *hints* (dicas que aparecem automaticamente ao parar o apontador sobre um objeto) para os principais objetos das interfaces, de modo a ajudar o usuário no entendimento do sistema;

melhoria: o uso de *hints* foi implementado usando a mesma arquitetura dos textos das janelas, o que viabiliza a sua internacionalização. No momento estão disponíveis textos de *hint* em inglês e português. A Figura 13 ilustra um desses *hints*;

- indicação: o sistema, originalmente, possuía um conjunto de configurações padrão. Na avaliação ficou claro que essas configurações não eram adequadas a todas as situações;

melhoria: as configurações no sistema passaram a ser obtidas de um arquivo de configuração em formato texto que pode ser editado para alteração de seus valores;

- indicação: era bastante trabalhoso selecionar os usuários que seriam representados no Gráfico de Conversas e no Gráfico Condensado de Conversas;

melhoria: foi desenvolvido um *wizard* para selecionar os agentes que o facilitador queira ver representados nos gráficos. O *wizard* automaticamente abre as janelas dos gráficos, distribuindo-as na tela, evitando sobreposição e selecionando os agentes a serem acompanhados. A Figura 10 representa a janela de *wizard*;

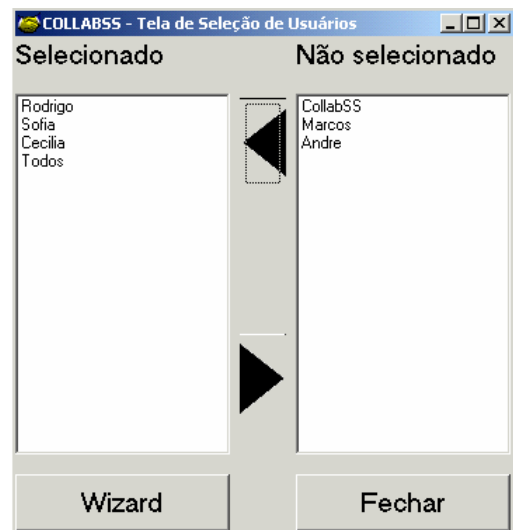


Figura 10. Janela de *wizard* de usuários.

- indicação: o facilitador pode identificar um agente precisando de ajuda ao consultar o Gráfico Condensado de Conversas;

melhoria: quando o facilitador clicar em um retângulo do Gráfico Condensado de Conversas, a janela de envio de iniciadores de conversa é aberta tendo como agente pré-selecionado o representado pela linha do retângulo escolhido;

- indicação: deveria ser possível para o usuário solicitar que o CollabSS não avalie a participação de um agente específico. Isto é importante porque algumas vezes um agente interrompe sua participação no meio de uma atividade e, na versão testada, quando isto ocorria, CollabSS mantinha-se identificando possíveis oportunidades de apoio a colaboração para agentes não ativos naquele momento;

melhoria: criação de uma janela de habilitação/deshabilitação (Figura 11). Apenas usuários habilitados são acompanhados pelo CollabSS para detectar-se oportunidades de apoio à colaboração. A partir dessa indicação foi também implementada a possibilidade de o sistema externo indicar ao CollabSS a desconexão de um agente. CollabSS automaticamente desabilita agentes que tenham se desconectado;



Figura 11. Janela de habilitação de usuários.

- indicação: deveria ser possível chamar a janela de habilitação de usuário diretamente da janela de envio de iniciadores. Isso se deve ao fato de que, algumas vezes, na tela de envio de iniciadores, o facilitador detecta que o agente pré-selecionado não deveria ser mais acompanhado pelo sistema porque, por exemplo, este agente saiu, mas manteve o computador conectado;

melhoria: foi implementada esta chamada. A Figura 12 apresenta a nova versão da janela de envio de iniciador de conversa, com esta opção no menu da janela;

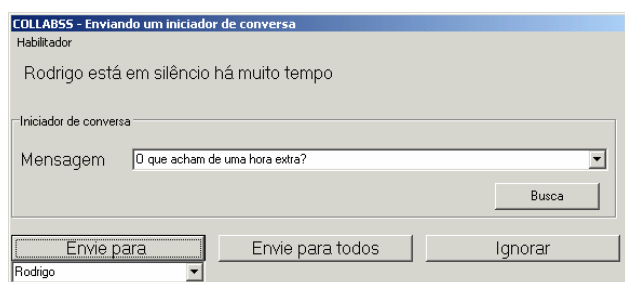


Figura 12. Versão alterada da tela de envio de iniciador de conversa.

- indicação: na janela de envio de iniciadores de conversa, o sistema escolhia um iniciador de conversa e, quando a possível oportunidade de apoio à colaboração fosse relacionada a um agente específico, fixava esse agente como possível destinatário do iniciador de conversa;

melhoria: conforme representado na Figura 12, o sistema não mais impõe um agente para o qual um iniciador deve ser enviado. CollabSS ainda sugere um agente, mas o usuário pode selecionar outro de uma lista apresentada. O sistema apresenta todos os iniciadores em uma lista e mantém selecionado o último iniciador usado. Além disso, provê funcionalidades para apoiar o usuário na seleção desses iniciadores. Caso o usuário queira procurar um iniciador com uma palavra ou texto específicos, basta digitar este texto no campo dos iniciadores e pressionar o botão “Busca”, ilustrado na Figura 10;

- indicação: quando o CollabSS detecta uma possível oportunidade de apoio à colaboração, ele apresenta o botão correspondente na janela inicial (Figura 3). Mas nem sempre os facilitadores tinham interesse em usar a oportunidade detectada. O facilitador precisava, então, clicar no botão e na janela de envio de iniciadores selecionar “ignorar”;

melhoria: basta o usuário pressionar o botão na janela inicial com o botão direito do mouse para que o sistema busque uma nova possível oportunidade de apoio à colaboração;

- indicação: alguns usuários tiveram dificuldades em interpretar os gráficos de conversa;

melhoria: inicialmente as falas eram representadas por linhas sem indicação explícita de orientação. De modo a facilitar o entendimento foram substituídas por setas, conforme representado nas Figura 13;

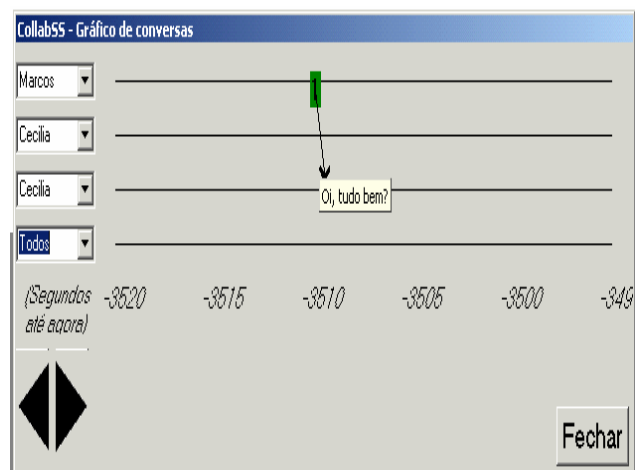


Figura 13. Versão alterada da tela do gráfico de conversas.

- indicação: os facilitadores prefeririam que fossem apresentadas porcentagens e não em valores absolutos nos gráficos de barra;

melhoria: foi implementada a funcionalidade de escolha pelo facilitador de gráficos com números absolutos ou porcentagem, conforme ilustrado na Figura 14.

CollabSS - Falas por usuário		Porcentagem
1	Rodrigo	
1	Sofia	
1	Cecilia	
CollabSS		
3	Marcos	

Falas 3

Figura 14. Versão alterada da tela de falas por usuário.

CONCLUSÃO

Este trabalho apresenta um uso prático da técnica de avaliação heurística participativa. Foi apresentado como a técnica foi adequada e adotada para a avaliação do sistema CollabSS.

Conforme apresentado de forma sucinta, a AHP resultou em um conjunto significativo de melhorias não somente em elementos de interface do usuário, mas também nas funcionalidades do sistema CollabSS. Dessa forma, esta atividade superou as expectativas iniciais com relação aos resultados alcançados para a melhoria do sistema.

Conforme o número de sugestões propostas durante a avaliação, pode-se dizer que a técnica foi eficiente não somente em avaliar o *design*, mas também em detectar oportunidades de melhorias nas funcionalidades do sistema. Outro resultado importante foi que as oportunidades identificadas resultaram em melhorias que foram incorporadas na última versão do CollabSS. Essa versão incorpora um conjunto de novas funcionalidades, sendo também mais flexível.

Apesar do número de melhorias indicadas, todos os avaliadores concordaram que o CollabSS é uma ferramenta útil no apoio à colaboração entre agentes interagindo através de um sistema colaborativo com base em comunicação textual. Foi identificado que cada avaliador utilizou-se o CollabSS de forma distinta. Um avaliador usou o sistema principalmente para prover iniciadores de conversas. Outro não usou os iniciadores de conversa, mas apenas os gráficos para analisar a interação. O terceiro usou as duas funcionalidades conjuntamente. Portanto, é importante o fato de o CollabSS oferecer um conjunto de funcionalidades diferentes de forma a apoiar diferentes estilos de usuários e usos.

Como continuidade do trabalho está sendo proposta a integração do CollabSS com uma ferramenta de ensino a distância e a condução de uma avaliação heurística participativa nesse outro contexto de uso.

AGRADECIMENTOS

Agradecemos ao Núcleo de Informática Aplicada à Educação (Nied)-Unicamp e seus membros pela colaboração com este trabalho.

REFERÊNCIAS

1. Baranauskas, M.C.C., Gomes Neto, N., Borges, M.A. F. Gaming at work: a learning environment for synchronized manufacturing. *Computer Applications in Engineering Education Journal*, Wiley & Sons, v.8, n.3/4, p.162-169, 2000.
2. Baranauskas, M.C.C., Gomes Neto, N., Borges, M.A. F. Learning at work through a multi-user synchronous simulation game. *International Journal of Continuing Engineering Education And Life-long Learning*, RU, v. 11, n. 3, p.251-260, 2001.
3. Borges, M. A. F. O design centrado no aprendiz no sistema Jonas: uma experiência de formação na indústria. Dissertação de mestrado (Ciência da Computação) - Universidade Estadual de Campinas, Baranauskas, M.C.C. (Orientadora), 1997.
4. Borges, E.L. Design de um ambiente computacional de modelagem e simulação para formação de pessoal na industria. Dissertação de mestrado (Ciência da Computação) - Universidade Estadual de Campinas, Baranauskas, M.C.C. (Orientadora), 1998.
5. Guerrero, L.A. , Alarcon, R., Collazos, C., Pino, J.A., Fuller, D.A. Evaluating Cooperation in Group Work. *Proceedings of the Sixth International Workshop on Groupware*, Madeira, Portugal, IEEE Computer Society, p.28- 35, 2000.
6. Borges, M.A.F., Baranauskas, M.C.C. (2003). CollabSS: a Tool to Help the Facilitator in Promoting Collaboration among Learners. *Educational Technology & Society*. Volume 6, No 1, [Online reference, see http://ifets.ieee.org/periodical/vol_1_2003/borges.pdf, referred 18th March, 2003].
7. Muller, M.J., Matheson, L., Page, C., Gallup, R. Participatory Heuristic Evaluation. *Interactions*, september + october, 1998.
8. Nielsen, J., Mack, R.L. *Usability Inspection Methods*, Wiley, New York, EUA, 1994, apud Muller et al., 1998.

Avaliando a Qualidade da Interação de Usuários com Dispositivos Móveis

Milene Silveira

Márcio Pinho

Adriano Gonella

Patrick Calvetti

Projeto *Mobile Museum*
Faculdade de Informática/PUCRS
Avenida Ipiranga, 6681 – Prédio 30
Porto Alegre – RS - Brasil
{milene, pinho, gonella, pcalvetti}@inf.pucrs.br
55 51 33203558

RESUMO

Testes com usuários são tradicionalmente realizados em laboratórios, onde o usuário fica sentado em frente ao computador, e realiza uma série de tarefas solicitadas, enquanto tem sua interação observada (e gravada). Mas, como realizar estes testes quando o usuário necessita se deslocar durante a interação? Este artigo discute o que é feito atualmente nestes casos e apresenta um estudo de caso a fim de coletar mais dados para esta discussão. Neste estudo de caso, o usuário necessita utilizar um dispositivo móvel do tipo Tablet-PC durante a visita a um Museu. Foram realizados testes para verificar como se dá esta interação e, principalmente, que alternativas existem para realizar estes testes em ou extra-laboratório.

ABSTRACT

Usability tests are usually performed inside laboratories, where the user is placed in front of a computer and accomplishes a series of requested tasks, while his interaction is observed and recorded. But, what if the user needs to walk during the tests, how to perform this kind of evaluation? This paper discusses what has been currently done by the researchers to address this problem and presents a case study in order to collect more meaningful data. In the case study the user used a mobile computer while visits a Science Museum. Tests were developed to analyze how this interaction is performed and which alternatives are available to do the tests in or outside laboratories.

Palavras Chaves

Testes com usuários, interação com dispositivos móveis.

INTRODUÇÃO

Na realização de testes com usuários, para avaliar a qualidade de interação de uma aplicação, tipicamente o usuário recebe uma série de tarefas a realizar e as efetua sentado em frente ao computador, enquanto um integrante da equipe de avaliação observa a atividade [11,14]. A forma como esta observação é realizada pode variar desde a observação simples e direta, com o integrante da equipe

de avaliação sentado próximo ao usuário, anotando detalhes da interação, até o uso de câmeras de vídeo ou salas de observação sofisticadas, com vidros espelhados e equipamentos especiais (Figura 1).



Figura 1: Exemplo de laboratório de usabilidade [16]

Estes tipos de testes são tradicionalmente usados para dispositivos do tipo *desktop*, onde a aplicação a ser avaliada é utilizada no dia-a-dia do usuário final, com a pessoa sentada em frente ao computador, em seu lar, local de trabalho ou lazer. Mas, e quando a interação é feita com dispositivos móveis (*handhelds*)? Os mesmos tipos de teste podem ser realizados? A observação de um usuário utilizando o dispositivo móvel sentado em uma sala traz os mesmos resultados que fazê-lo em movimento, em um ambiente real?

Diferentes autores têm tratado deste tema [1,6,7,17], mas ainda não existe um consenso sobre a melhor abordagem a ser seguida para estes testes.

Com o objetivo de aprofundar a pesquisa neste campo, obtendo-se mais dados a respeito de alternativas de testes com usuários com dispositivos móveis em ambientes reais, decidiu-se, como objeto de estudo, o uso destes dispositivos em um ambiente de museu¹.

Este artigo apresenta o que é encontrado atualmente na literatura sobre testes com usuários com dispositivos móveis e o estudo de caso realizado. Neste estudo de caso, são descritos: o ambiente em que o mesmo foi

VI Simpósio sobre Fatores Humanos em Sistemas Computacionais — Mediando e Transformando o Cotidiano. Hotel Bourbon, Curitiba, 17 a 20 de outubro de 2004. UFPR, CEIHC—SBC.

¹ Este trabalho está sendo desenvolvido em colaboração com a HP Brasil P&D.

realizado, os protótipos criados para uso neste ambiente, bem como os testes realizados com o mesmo e os resultados obtidos. A partir destes resultados, são discutidas alternativas para condução de testes com usuários e feitas algumas considerações finais.

ABORDAGENS SOBRE TESTES COM USUÁRIOS DE DISPOSITIVOS MÓVEIS

A maioria dos projetos que testam usabilidade de aplicações para dispositivos móveis, fazem estes testes em laboratório (Figura 2), pois acreditam que, desta forma, é possível capturar praticamente a mesma quantidade de problemas de interação com o usuário [7].



Figura 2: Avaliação de dispositivo móvel em laboratório [6]

Estes testes são realizados com o usuário trabalhando sentado em uma sala, com o dispositivo móvel sobre uma mesa ou apoiado em um tripé (Figura 3), em contraste à situação de um ambiente real, onde este usuário estaria movimentando-se durante a interação, a fim de realizar suas atividades.

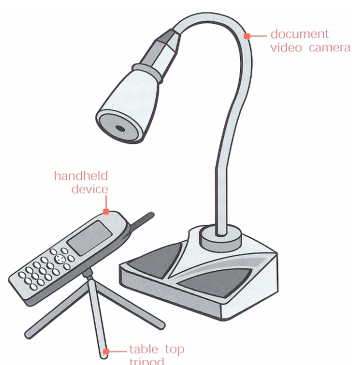


Figura 3: Configuração de vídeo para avaliação de usabilidade de dispositivo móvel em laboratório [17]

Esta abordagem é defendida por alguns autores por julgarem a sua alternativa, os testes em ambiente real, muito complexa, devido a diversos problemas, como [1,6,7]:

- a dificuldade de se ter acesso a usuários reais e a localizações reais;
- a dificuldade de fazer a coleta de dados;
- a limitação das formas de controle dos testes;
- o deslocamento do foco do usuário para elementos do ambiente e não só para a questão de usabilidade do sistema.

A principal questão levantada é o pequeno ganho que se obtém com os testes em campo, que têm elevados custos

em termos de tempo e de infra-estrutura, com equipamentos sofisticados para gravação da interação e grande número de pessoas envolvidas (entre usuários e equipe de observação), em relação aos testes em laboratórios realísticos. Nestes laboratórios tenta-se simular as situações reais o mais fielmente possível, recriando fisicamente os ambientes de uso reais. Em um projeto, por exemplo, foi feita a simulação, em laboratório, de um quarto de hospital, com alunos deitados em camas como se fossem pacientes [7]. Outro, garantia, em laboratório, o desvio de atenção do usuário, através da recomendação de que este realizasse atividades que demandassem cognição, como dirigir ou jogar no computador, enquanto utilizasse o dispositivo móvel [1].

Por mais que estas abordagens busquem simular o ambiente com precisão, a situação que elas apresentam para o usuário nunca será igual à sua realidade. O custo das simulações também pode tornar-se bastante elevado, dependendo do nível de detalhe desejado. Além disso, é fundamental considerar que os testes em laboratório funcionam bem com soluções onde contexto e ambiente são de segundo interesse [5].

O único consenso que se conseguiu obter a partir destes trabalhos foi que todos mostram que ainda não há uma concordância sobre um conjunto de métodos de avaliação de usabilidade e técnicas para a coleta de dados. Com base nisso, foi realizado um estudo de caso, a fim de tentar obter-se uma quantidade maior de dados para esta discussão.

ESTUDO DE CASO

Conforme visto na seção anterior, testes de usabilidade de dispositivos móveis têm sido realizados em laboratório, dado os problemas encontrados nas tentativas de realizá-los em ambiente real.

Estes problemas, bem como questões relacionadas aos mesmos, servem de base para os questionamentos aqui levantados:

1. *É difícil ter acesso a usuários reais e a localizações reais.* Uma alternativa a isto é aumentar o realismo do laboratório, incluindo mobilidade e contexto, como visto anteriormente no caso da simulação do hospital [7]. Em situações como as que envolvem risco ou questões éticas, por exemplo, que não permitem que o teste seja feito em campo, esta é uma alternativa interessante. Mas, será que é uma alternativa interessante sempre? Se dois dos maiores problemas são tempo e custo, será que a reprodução do ambiente não é mais cara e não leva mais tempo para ser realizada?
2. *A dificuldade de fazer a coleta de dados.* São muitos os equipamentos a serem utilizados para capturar os dados da interação, indo desde câmeras de vídeo portáteis carregadas por um observador, que também carrega *receivers* para áudio e vídeo do usuário, dentre outros [7]. Estes equipamentos devem ser carregados enquanto acompanha-se (em movimento) o usuário durante a avaliação, o que leva a uma dificuldade grande de conseguir-se

manter o foco na tela do dispositivo usado pelo usuário, além da quantidade de pessoas necessárias para este acompanhamento [1]. Defende-se que fazer isto em laboratório é mais simples, além de mais efetivo. Mas, será que são realmente necessários todos estes equipamentos para se ter dados suficientes para a análise da qualidade da interação?

3. *A limitação das formas de controle dos testes e O deslocamento do foco do usuário para elementos do ambiente e não só para a questão de usabilidade do sistema.* O usuário pode ter sua atenção voltada para outros elementos do ambiente e, também, efetuar tarefas diferentes das solicitadas, o que pode desviar seu foco dos fatores de usabilidade a serem avaliados. Mas, será que este desvio de atenção não se faz necessário, a fim de mostrar quais elementos e mensagens do sistema precisam ser ressaltados?

Portanto, a fim de obterem-se mais dados para esta discussão, principalmente, no sentido de tentar contornar e/ou gerenciar estes tipos de problemas, foram elaborados e realizados diferentes testes com dispositivos móveis.

Ambiente de Aplicação dos Testes

Para a realização deste trabalho foi escolhido o Museu de Ciências e Tecnologia da PUCRS (MCT) [10], por estar situado dentro do Campus da Universidade e, também, por apresentar duas características relevantes para o estudo²:

- uma variedade enorme de opções de visitação, interativas em sua maioria. O MCT conta, em seu acervo, com mais de 5.000 peças e 700 experiências interativas nas áreas de Física, Matemática, Ecologia, Astronomia, Medicina, Arqueologia, Computação e outras, em uma área de exposição 12.500 m², dividida em três andares;
- o grande volume de pessoas que visitam o MCT. Ele recebe, por dia, cerca de 1.300 visitantes do Brasil e do exterior, sendo que 80% destes visitantes são provenientes de escolas de ensino médio e de ensino fundamental.

A quantidade de experimentos e de visitantes, bem como a diversidade de perfis destes visitantes, tais como alunos, professores, pesquisadores, famílias e público em geral, favorece este estudo. Em um período normal de visitação, existem dezenas de pessoas circulando pelo MCT em um mesmo momento, podendo este número atingir a casa das (várias) centenas (Figura 4). A atenção dos usuários é freqüentemente deslocada para um novo experimento, aviso sonoro ou acontecimento dentro do ambiente. Uma pessoa utilizando um dispositivo móvel neste contexto pode inclusive ser confundida com mais uma atração do Museu.



Figura 4: Visão parcial do MCT

Este cenário provê não somente um ambiente propício ao uso de dispositivos móveis, mas também permite a realização de testes com usuários dos mais diferentes níveis de conhecimento (alfabetização) computacional, o que traz achados também sobre as facilidades e/ou dificuldades de uso deste tipo de dispositivo pela população em geral.

Protótipo Desenvolvido

Atualmente, como ferramenta de auxílio à visitação dentro do MCT, o visitante dispõe de totens (quiosques com computador), espalhados pelo Museu, com descrições de cada experimento, e, fora do Museu, ele dispõe do sistema SAGRES [2,8], acessível através da Internet [15].

O SAGRES permite que o usuário procure informações sobre as experiências e pesquisas em desenvolvimento nos laboratórios do Museu, podendo, assim, ter informações para planejar sua visita.

Este sistema é utilizado, pelos visitantes, em suas casas ou escolas, por exemplo, mas não durante a visitação. Para melhor estudar a interação durante a visitação, foi desenvolvido um protótipo de sistema para auxiliar professores-visitantes do MCT a elaborarem roteiros de visitação para seus alunos.

A idéia central é que, com o uso de um dispositivo do tipo Tablet-PC (Figura 5), um professor possa ter acesso a informações detalhadas sobre artigos ou experimentos do acervo do MCT à medida que os mesmos são visitados e, a partir disto, elabore os roteiros a serem usados por seus alunos em uma visita futura.



Figura 5: Dispositivo móvel do tipo Tablet-PC

Este recurso personaliza a interação do professor no Museu e permite este faça as pesquisas desejadas, no

² Dados de 2003, fornecidos pelo MCT.

momento e no nível de profundidade que julgar adequado, além de proporcionar um instrumento que leve seus alunos a ter maior foco durante a visitação. A mobilidade permitida por este tipo de equipamento possibilita que o professor não tenha que deslocar-se até os totens ou recorrer a guias especializados, quando sentir necessidade, podendo aprofundar seus conhecimentos, durante a visita, de acordo com seus desejos e necessidades.

Processo de Design do Protótipo

A primeira idéia para o protótipo foi a de fazer um guia eletrônico do MCT, transpondo para o Tablet-PC o que já havia no SAGRES, mudando-se a forma de apresentação das informações para o novo dispositivo e acrescentando-se novas funcionalidades para apoio a visitação.

Mas, após uma série de reuniões realizadas com a coordenação do MCT, a fim de descobrir qual a aplicação que mais se adequaria às necessidades dos mesmos, para apoio à visitação, surgiram duas idéias:

- sistemas de exibição de roteiros: os professores criariam roteiros que seriam acessados via sistema no Tablet-PC e seus alunos usariam um Tablet-PC para explorar estes roteiros durante a visita; e,
- sistema para criação dos roteiros: o professor usaria o Tablet-PC, para criar roteiros de visita para seus alunos, em visita anterior destes ao Museu. Os alunos não usariam o dispositivo móvel durante a visita e, sim, o roteiro criado pelo professor, disponibilizado aos alunos em papel, por exemplo.

A partir de cenários de interação desenvolvidos para estas duas alternativas, em novas reuniões com a coordenação do Museu, a escolha ficou com a segunda opção, dada, principalmente, a característica principal de interatividade total do Museu. O MCT é um museu interativo, ou seja, o visitante precisa de grande liberdade para explorar os experimentos e, esta seria cerceada, se houvesse a necessidade de o mesmo carregar o equipamento (e cuidar de) durante a visita.

Foi definido, então, que o professor utilizaria o sistema, no Tablet-PC, para construir roteiros de visita para seus alunos.

Um roteiro seria composto por uma seqüência de perguntas sobre determinados experimentos, a fim de guiar a visita do aluno ao Museu. Com o roteiro em mãos – de forma impressa, por exemplo - o aluno deveria tentar buscar os experimentos que se relacionassem à pergunta encontrada no roteiro e interagir com o mesmo a fim de conseguir responder a pergunta. Desta forma o aluno poderia ter uma parte de sua visita bastante focada no assunto que o professor estivesse trabalhando em aula.

A partir desta definição de foco do sistema, foi feita a modelagem de tarefas e de interação do mesmo [8], em conjunto com a equipe do Museu. Este processo sempre teve como foco principal seu público-alvo, professores de ensino fundamental e médio, com variados níveis de alfabetização computacional.

Durante o período de prototipação, além de acompanhamento da equipe do Museu, foram feitas avaliações heurísticas [11] e refinamentos sucessivos nos protótipos.

No final, foram criados dois protótipos: um para apoio à equipe do MCT, no cadastro das informações dos experimentos, perguntas e roteiros, e outro para uso pelo professor, durante a visita.

A descrição funcional de ambos, com *softwares* usados para seu desenvolvimento e funcionamento, bem como ambos se relacionam, estão detalhados no esquema a seguir (Figura 6). A seguir, cada um dos protótipos será detalhado.

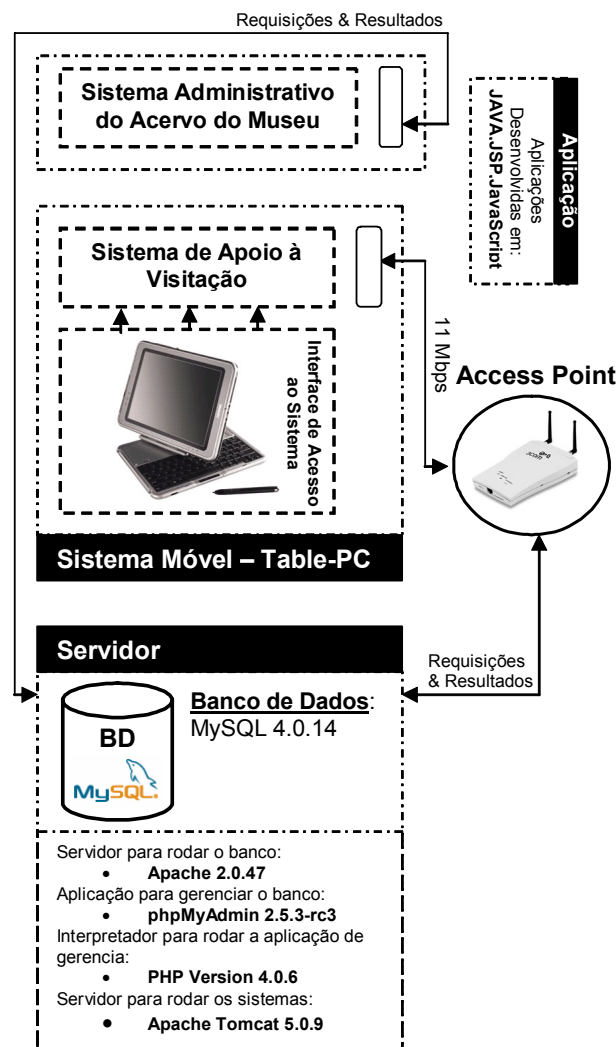


Figura 6: Esquema de Funcionamento dos Protótipos

Sistema Administrativo do Acervo do Museu

O Sistema Administrativo do Acervo do Museu (Figura 7) foi projetado para possibilitar o gerenciamento da base de dados que é acessada pelo Sistema de Apoio à Visitação.



Figura 7: Sistema Administrativo - Cadastro de Experimentos e Consulta a Perguntas

Através dele, funcionários do MCT podem cadastrar não só os experimentos existentes do Museu e os dados relativos aos mesmos, mas também, perguntas relacionadas a estes, classificadas de acordo com níveis de escolaridade e matéria. Também é permitido, através deste sistema, que funcionários do Museu criem roteiros de visitação, a partir das perguntas pré-cadastradas.

A idéia é que, quando o professor vá utilizar o sistema de visitação, ele já tenha diversas perguntas cadastradas, com as quais ele poderá montar seus roteiros ou, até, utilizar roteiros anteriormente criados pelo Museu (ou, porque não, por outros colegas que já tenham utilizado o sistema).

Sistema de Apoio à Visitação

O Sistema de Apoio à Visitação (Figura 8) foi projetado para apoiar os professores no planejamento de roteiros de visitas de seus alunos ao MCT, através da utilização de um dispositivo móvel do tipo Tablet-PC.

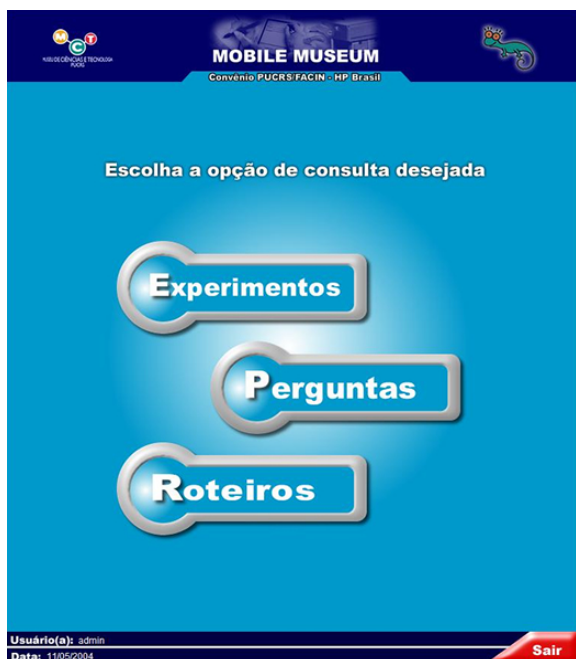


Figura 8: Tela Principal do Sistema de Apoio à Visitação

Para definir seus roteiros, o professor primeiro poderá consultar dados sobre os experimentos existentes no Museu e verificar perguntas relacionadas a cada um destes experimentos. Quando encontrar uma pergunta adequada ao assunto que ele esteja trabalhando com seus alunos, ele poderá incluí-la em seu roteiro. Além de criar roteiros personalizados, o professor poderá analisar e selecionar algum dos roteiros previamente cadastrados pela equipe do Museu ou por outros professores.

A idéia é que o professor possa utilizar o sistema via Tablet-PC para construir o roteiro e, após a finalização do mesmo, leve uma cópia impressa (ou envie para seu endereço de *email*), a fim de disponibilizá-lo a seus alunos.

Para tanto, o sistema permite que o professor pesquise sobre um determinado experimento encontrado fisicamente no MCT (Figura 9), através de sua localização (o Museu é dividido em Pavimentos e estes em Localizações bem demarcadas visualmente).



Figura 9: Sistema de Apoio à Visitação – Consulta a Experimentos

Determinado o experimento, o professor pode então consultar as perguntas relacionadas ao mesmo (Figura 10) e, caso desejado, proceder a inclusão destas em seu roteiro particular.

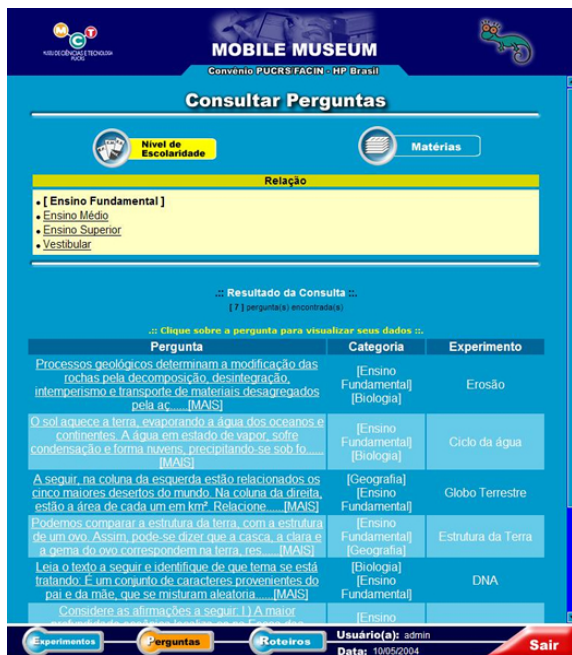


Figura 10: Sistema de Apoio à Visitação – Consulta a Perguntas

Da mesma forma, é possível iniciar a elaboração do roteiro através da consulta de perguntas e respostas, a partir do nível de escolaridade ou da matéria desejados. Além disto, o sistema permite consultar roteiros previamente elaborados.

Para o desenvolvimento deste sistema, especificamente, houve uma série de desafios, dentre eles (e principalmente):

- a grande diversidade de níveis de alfabetização computacional do público alvo;
- a entrada principal de dados via caneta;
- a grande quantidade de informações a ser apresentada a cada nova consulta;
- a grande diversidade de acontecimentos – simultâneos – no Museu, enquanto o usuário tenta obter informações através do sistema.

As duas primeiras questões foram tratadas em conjunto. A interação é toda feita através da seleção de opções. Nesta primeira versão do sistema, não existe necessidade de entrada de dados, a não ser para o *login* no sistema, opção esta que é feita uma única vez durante a interação, utilizando-se o teclado virtual do dispositivo.

Foram projetados botões grandes – para melhor acesso via caneta – e diferenciados das demais informações por cor e formato, para uma melhor visualização (Figura 11).



Figura 11: Exemplos de Botões para Acesso aos Pavimentos

Já para a grande quantidade de informações apresentadas como resultado da consulta, fez-se uma via de acesso alternativo, através das iniciais de cada resultado (experimentos, perguntas, etc), na tentativa de diminuir a

quantidade de rolagem de telas necessária para o acesso a informação desejada (Figura 12).



Figura 12: Acesso ao Resultado da Consulta

Em relação à diversidade de acontecimentos simultâneos no Museu, surgiu uma outra questão importante: a limitada capacidade de atenção do usuário enquanto este está se movendo. Em [12], há uma interessante comparação sobre isto. Seus autores dizem que um dispositivo móvel é visto tipicamente como uma ferramenta para o domínio de negócios, onde o usuário não está se movendo todo o tempo: ou o dispositivo está sobre a mesa, ou no colo, por exemplo, e o maior impacto de *design* está na miniaturização das informações. Em contraste, eles citam os trabalhadores de campo, que estão em constante movimento e, principalmente, têm uma capacidade limitada de atenção, pois seu foco principal é o trabalho que estão realizando em campo.

O público-alvo da pesquisa aqui discutida assemelha-se em muito com os trabalhadores de campo. Assim como estes, os professores (e demais visitantes do Museu) têm sua atenção constantemente desviada para diferentes pontos – que não o que está sendo mostrado no dispositivo: são 700 experimentos e centenas de pessoas se deslocando por entre eles e interagindo com eles.

Então, como dispor a informação de forma que o usuário, além de conseguir encontrá-la facilmente, possa se lembrar do que estava tentando fazer e onde estava, após ter sua atenção distraída por algum evento ao seu redor?

A principal decisão de *design* a este respeito foi manter o usuário constantemente informado, por meio de cores contrastantes, destaques em palavras ou expressões, e mensagens diretas, de onde ele está no sistema e o que ele deve fazer para prosseguir.

A Figura 13, por exemplo, mostra, em (a), o destaque de onde o mesmo se encontra (Pavimento e Localização) em relação ao Museu, e, em (b), em relação ao sistema.



Figura 13: Destaques na Interface

Testes

Em paralelo a finalização dos protótipos anteriormente descritos, foi feita a elaboração dos testes a serem efetuados com os usuários finais do mesmo. As etapas seguidas na realização dos mesmos estão descritas em detalhe a seguir.

Preparação do ambiente

Para que fossem definidas as tarefas a serem solicitadas aos usuários durante os testes, primeiro teve que ser definida a área do Museu onde os mesmos ocorreriam.

O Museu tem três pavimentos e, para esta primeira bateria de testes não se pretendia, por questões de duração do teste e coleta de dados via gravação de interação, que a mesma abrangesse o Museu inteiro.

A fim de delimitar-se a área de abrangência, primeiro foram feitos testes com a instalação de um *Access Point*, dado que o Sistema de Apoio à Visitação, conforme visto na Figura 6, funciona via Internet, em uma rede sem-fio. Além da abrangência do mesmo, foi feita uma análise das perguntas já contidas na base de dados do sistema, para ver qual a área mais favorecida (com maior número de questões disponíveis) para os testes.

A área selecionada foi o Terceiro Pavimento, onde se encontram os experimentos de Física, matéria esta que tem o maior número de perguntas cadastradas até o momento.

Escolha dos Usuários

O público alvo deste trabalho é composto por professores que estejam indo ao Museu para preparar uma posterior visita de seus alunos. Para os primeiros testes realizados – os quais serão aqui relatados – foram escolhidos – com auxílio da coordenação do Museu – professores que já realizam algum tipo de atividade em conjunto com o Museu.

Foram constituídos três grupos de usuários:

- **grupo 1:** formandos do Curso de Licenciatura em Biologia, alunos da disciplina de Prática de Ensino, que elaboram atividades didático-pedagógicas junto ao Museu (**4 usuários**);
- **grupo 2:** professores de ensino fundamental, médio e universitário, alunos de Mestrado, que elaboram atividades didático-pedagógicas junto ao Museu (**5 usuários**);
- **grupo 3:** professores da PUCRS, que elaboram atividades didático-pedagógicas junto ao Museu (**4 usuários**).

Além dos usuários pertencentes a estes grupos ainda foi feito, antes de começarem estes testes, um **teste piloto**, para verificação da adequação das tarefas e tempo estimado, com uma aluna do curso de Pedagogia (**1 usuária**).

Todos os usuários desta primeira etapa tinham conhecimentos básicos em computação e já haviam ido, no mínimo uma vez, ao Museu (exceção feita à aluna do teste piloto, que fez um teste simulado em laboratório).

Elaboração da tarefa

De acordo com a área delimitada e as perguntas disponíveis foi desenvolvido um conjunto de sete atividades, onde o usuário, depois de conhecer os propósitos do sistema, deveria:

1. encontrar determinado experimento no Museu e posteriormente no sistema;
2. verificar a quantidade de perguntas existentes no sistema referentes a determinado experimento e os detalhes de cada uma delas;
3. verificar uma determinada pergunta e incluí-la em seu roteiro de visitação;
4. verificar a quantidade de experimentos cadastrados de acordo com determinada localização;
5. verificar a quantidade de perguntas cadastradas de acordo com determinada categoria;
6. verificar os detalhes de um dado experimento e a pergunta relacionada ao mesmo, incluindo esta pergunta em seu roteiro;
7. verificar a quantidade de perguntas cadastradas em seu roteiro.

Para cada uma destas tarefas, foi elaborado um cenário com o qual o usuário pudesse se identificar e realizar a tarefa. Por exemplo, para a realização da tarefa 4, descrita acima, era entregue o seguinte cenário:

Uma colega sua da escola pediu que você verificasse quantos experimentos existem no Sistema, relacionados à localização **Fluídos**, do **terceiro pavimento**.

Como esta foi uma primeira aproximação com o sistema e, mais ainda, com um novo tipo de dispositivo, o Tablet-PC, para esta seqüência de testes, foram elaboradas

tarefas simples, que necessitassem apenas a seleção de elementos e leitura de informações na tela.

Tempo Previsto para Duração das Tarefas

As tarefas foram elaboradas e testadas de forma que a execução de todas elas levasse no máximo 30 minutos.

Equipe e Equipamentos Utilizados

A equipe de observação foi composta por cinco pessoas, as quais se revezavam nas tarefas de observação dos usuários e restauração do estado inicial do sistema, para cada novo teste.

Para a realização dos testes foram utilizados dois dispositivos móveis do tipo Tablet-PC, com o sistema de visitação e com *software* de captura de interação instalados.

Realização dos Testes

Os grupos de usuários tiveram um mesmo procedimento de teste (mesmo conjunto de tarefas e mesmos observadores), mas executaram os testes em locais diferentes.

Procedimento Comum a Todos os Testes

A cada usuário, ao iniciar o teste, era apresentada a idéia do projeto e o objetivo do testes. Após o preenchimento dos formulários iniciais e questionários pré-teste, para definição do perfil dos usuários, os mesmos recebiam uma pequena instrução sobre como carregar o dispositivo e como manusear a caneta. Após alguns momentos para o usuário se familiarizar com os novos dispositivos, era dado início ao teste.

O usuário ao receber o primeiro cenário de tarefa, começava a deslocar-se pelo ambiente do terceiro pavimento, a fim de encontrar o experimento solicitado na tarefa. A cada tarefa cumprida um novo cenário de tarefa era entregue ao usuário, e, assim, sucessivamente, até a tarefa final. Todo o teste era acompanhado de perto por um observador, para anotação dos dados relevantes (principalmente aqueles que provavelmente não seriam capturados via *software* de captura de interação) e entrega das novas tarefas ao usuário, quando solicitado.

Todos os observadores estavam cientes de que não podiam ajudar o usuário a resolver nenhuma dúvida do sistema (a menos que fosse alguma “pane” no mesmo). No caso de dúvidas em relação ao Museu – localização de determinado experimento, por exemplo - e não em relação ao sistema, o usuário era informado de que poderia tirar suas dúvidas com funcionários do Museu, o que faria normalmente se estivesse usando o dispositivo em uma situação real.

Após a realização de todas as tarefas, o observador fazia uma série de questionamentos sobre o teste ao usuário, os quais serão discutidos posteriormente.

Locais de Aplicação dos Testes

O teste **piloto** foi feito em laboratório, com o usuário se deslocando durante a execução das tarefas, mas em localizações simuladas: cartazes dispostos pelo prédio, determinando as localizações e experimentos. A idéia

deste teste era somente equilibrar o tempo de execução de cada tarefa e determinar se o usuário conseguiria entender o que deveria ser feito em cada uma.

Já o **grupo 1** e o **grupo 2** efetuaram suas tarefas no próprio Museu, no terceiro pavimento, deslocando-se pelo mesmo durante a execução das tarefas. Para este teste foram utilizados dois grupos, por ter sido detectado um perfil muito homogêneo no primeiro: professores da “nova geração”, que utilizam o computador diariamente para suas atividades acadêmicas, de trabalho e lazer. Com isto foi selecionado um segundo grupo, com professores com conhecimentos básicos de informática, mas utilização menos freqüente deste tipo de tecnologia.

Com o **grupo 3**, utilizou-se uma abordagem de teste diferente, onde os mesmos – conhecedores mais a fundo do Museu – deveriam apenas utilizar o sistema e efetuar as tarefas, sem se deslocar pelo Museu. Estes usuários efetuaram os testes sentados em uma sala do Museu. Nas partes do teste que pediam o deslocamento até certa área do Museu, eles foram informados que não precisavam realizar esta movimentação.

Primeiras Impressões dos Resultados

Os resultados obtidos ainda necessitam de uma análise mais aprofundada, mas já foi possível obter algumas impressões iniciais sobre os testes efetuados.

Quanto ao **relato dos usuários**, no pós-teste (ou durante o teste através de verbalizações espontâneas dos mesmos), todos disseram gostar muito do sistema e que ele agilizaria muito o trabalho dos professores que viessem preparar a visita de seus alunos.

Muitos tiveram dificuldades iniciais em utilizar a caneta como dispositivo de entrada, mas após algumas interações tornaram-se mais fluentes (um verbalizou que, “depois de familiarizado, é tranquilo”). Os usuários citaram também o peso do equipamento, o que piorava quanto mais demorada era a interação, e, também, a necessidade de alguma espécie de bolsa para carregá-lo durante a visita o que possibilitaria uma maior interação com os experimentos e um maior conforto para quem o estivesse portando, além de maior segurança. A preocupação de muitos era de “cuidar” do equipamento, por ser um dispositivo a que não estavam acostumados e de ser de valor financeiro elevado.

Quanto à **observação da realização das tarefas**, durante os testes, pode-se observar problemas principalmente nas tarefas que requeriam que o usuário identificasse o número de experimentos ou de perguntas de determinado experimento. A mensagem do sistema que apresenta esta informação, logo no início do resultado da busca, está pouco ressaltada e, muitos usuários, literalmente, contaram quantos itens voltaram da busca, sem notar que ali havia uma mensagem do sistema. Outro item pouco destacado – e que levou a dificuldades de alguns usuários – foi o de perguntas relacionadas a experimentos. Muitos – em uma primeira “olhada” - não o conseguiram identificar, acabando por procurar outro caminho, mais visível na tela, como o da categoria Perguntas (na parte

inferior da tela), que levava as perguntas como um todo e não às perguntas desejadas.

DISCUSSÃO SOBRE A EXECUÇÃO DOS TESTES E SUAS FORMAS ALTERNATIVAS

Para a discussão das questões levantadas neste artigo, foram realizados testes de diferentes formas. Apesar de ainda serem necessários mais testes para uma visão melhor das alternativas existentes, algumas questões já podem ser discutidas:

1. *É difícil ter acesso a usuários reais e a localizações reais.* Uma alternativa a isto é aumentar o realismo do laboratório.

No estudo de caso aqui descrito, o público alvo é bastante grande (como visto, cerca de 200 mil pessoas visitam o Museu anualmente). Têm-se acesso a estes usuários e a localização real de uso do sistema. Apesar disto fez-se também uma série de testes em laboratório, para verificar se o mesmo cobriria todos os problemas que o teste no Museu apresentou.

Uma hipótese levantada neste caso – e que precisa ainda de mais alguns testes para ser comprovada realmente de fato – é que alguns problemas somente irão aparecer com o uso no ambiente real. A mensagem de quantidade de experimentos, citada anteriormente, foi um problema apresentado por quase todos usuários dos grupos 1 e 2. Já nos usuários do grupo 3, que estavam com maior foco sobre sistema, por se encontrarem sentados em frente ao mesmo, o percentual foi bem menor. Este estudo ainda precisa ser aprofundado, mas já dá indícios de que – por maior o número de problemas que se encontrem em um teste em laboratório – é difícil encontrar todos os problemas que se encontrariam no contexto real, onde a distração é maior. Neste caso, como as informações não estavam muito ressaltadas, muitos usuários dos grupos 1 e 2 nem as notaram. Se os testes tivessem sido feitos apenas em laboratório, talvez este problema – grave – tivesse passado despercebido.

A alternativa descrita ao uso em ambiente real, sua reprodução em laboratório, para este teste, seria bastante complicada. Além da necessidade de prover alguma espécie de experimentos ao usuário durante o teste, seriam necessárias várias pessoas circulando durante o teste, o que seria difícil de fazer em outro ambiente.

2. *A dificuldade de fazer a coleta de dados.* São muitos os equipamentos a serem utilizados para capturar os dados da interação (...).

Os testes foram efetuados utilizando-se apenas um *software* para gravação da interação e um observador acompanhando o usuário, a fim de anotar aqueles dados que não iriam constar na gravação. No caso de uso a caneta, o papel do observador ainda é maior, pois nos testes em geral, com o *software* de captura normalmente utilizado,

é possível notar se o usuário está com o *mouse* parado, perguntando-se o que fazer (“E agora?” [13]), é possível acompanhar a “leitura da tela” pelo *mouse*, e assim por diante. Com o uso da caneta, o cursor não fica se deslocando na tela. O usuário pode estar com a caneta na mão e “varrendo” a tela com os olhos. Isto não é possível capturar via *software*.

Uma idéia, com custo não muito alto e sem requerer uma grande quantidade de equipamentos a serem carregados durante a interação é acoplar-se ao dispositivo móvel uma pequena câmera de vídeo que capture não só a tela como a mão do usuário com a caneta também, o que pode dar uma visão melhor da interação. Mesmo com o observador junto, alguns dados importantes escapam a observação, porque – também – não se pode ficar muito próximo ao usuário para não atrapalhá-lo ou intimidá-lo.

Uma dúvida surgida durante a pesquisa, foi se a presença de um observador durante os testes, acompanhando o usuário (caminhando com ele pelo Museu), não acabaria por constrangê-lo. Esta pergunta foi feita literalmente aos usuários dos grupos 2 e 3, que, em unanimidade responderam que não só não atrapalhava, como lhes dava segurança durante este primeiro contato com um dispositivo não usual para eles.

3. *A limitação das formas de controle dos testes e O deslocamento do foco do usuário para elementos do ambiente e não só para a questão de usabilidade do sistema.*

Volta-se aqui à discussão já desenvolvida no item 1, onde se defende que o usuário deve, sim, ter sua atenção deslocada para outros elementos pois só assim será possível saber se a comunicação (visual, textual ou sonora) do sistema estará surtindo efeito.

CONSIDERAÇÕES FINAIS

Ainda há muito a ser pesquisado. Através dos testes realizados conseguiu-se uma visão melhor deste tema ainda em constante discussão: *como conseguir realizar testes com usuários com dispositivos móveis?*

Para esta pesquisa, o objetivo é ainda mais amplo: *como conseguir realizar testes com usuários com dispositivos móveis, em ambiente real, e a baixo custo?* A idéia é propor alternativas que permitam aos projetistas verificarem se suas idéias estão de acordo com as necessidades e desejos de seus usuários, sem que para isto se precise de equipamentos e/ou laboratórios de teste altamente especializados. E ressaltar a idéia de que o ambiente em que o dispositivo móvel estiver sendo usado, bem como o público deste ambiente, influirá em muito na interação do usuário.

Foi possível realizar os testes aqui mencionados a custo baixo e em ambiente real. O uso de equipamentos como a câmera de vídeo ajudaria a equipe de observação, mas sem ela os testes também foram possíveis.

A fim de continuar esta discussão, pretende-se agora buscar usuários que não estejam acostumados a visitas ao Museu. Ou seja, buscar usuários que estejam vindo ao Museu preparar suas visitas pela primeira vez. E alterar as tarefas de teste para que estes possam criar seus próprios roteiros – situação de uso real – e esta criação seja observada. Para isto terá que ser analisada novamente a forma de captura dos dados, se por vídeo – o que, pela duração maior do trabalho, ficará muito grande – ou apenas por observação direta, por exemplo.

Além disto, como trabalho futuro, pretende-se explorar a questão da influência do contexto na interação, onde o dispositivo obtém do ambiente sua localização, que agregada à atividade atual do usuário, pode acrescentar mais recursos ao processo de interação [3,4].

Estão previstos, também, estudos para migração do sistema de visitação para dispositivos do tipo PDA (*Personal Digital Assistant*), na tentativa de facilitar o trânsito e o trabalho do professor-visitante dentro do Museu.

AGRADECIMENTOS

A coordenação do Museu de Ciências e Tecnologia da PUCRS e todos seus funcionários. A todos os usuários que participaram dos testes e contribuíram para a execução deste trabalho. Aos professores que não só disponibilizaram tempo de seus alunos para a execução dos testes, mas também os incentivaram a efetuá-los.

REFERÊNCIAS BIBLIOGRÁFICAS

1. Beck, E.T., Christiansen, M.K., Kjeldskov, J., Kolbe, N., Stage, J. Experimental Evaluation of Techniques for Usability Testing of Mobile Systems in a Laboratory Setting. *Proceedings of OzCHI 2003, Brisbane, Australia*.
2. Bertolotti, A.C., Moraes, M.C., Costa, A.C.R. Providing Personal Assistance in the SAGRES Virtual Museum. *Proceedings of International Conference on Museums and the Web, 2001*.
3. Burrell, J., Treadwell, P.; Gay, G.K. Designing for Context: Usability in a Ubiquitous Environment. *Proc. 2000 Conference on Universal Usability, nov.2000*
4. Fogli, D., Pittarello, F., Celentano, A., Mussio, P. Context-Aware Interaction in a Mobile Environment. *Mobile HCI 2003 conference, Lecture Notes in Computer Science (LNCS 2795), Springer-Verlag Berlin Heidelberg 2003, pp.434-439*.
5. Johnson, P. Usability and Mobility; Interactions on the move. *First Workshop on Human Computer Interaction with Mobile Devices. GIST Technical Report G98-1. 21-23rd May 1998. Department of Computing Science, University of Glasgow, Scotland*.
6. Kjeldskov, J., Skov, M.B. Evaluating the Usability of Mobile Systems: Exploring Different Laboratory Approaches. *Proceedings of 10th International Conference on Human-Computer Interaction 2003, Crete, Greece. Lawrence Erlbaum Associates*.
7. Kjeldskov, J., Skov, M.B., Als, B.S., Høegh, R.T. Is it Worth the Hassle? Exploring the Added Value of Evaluating the Usability of Context-Aware Mobile Systems in the field. *Accepted for publication in Proceedings of the 6th International Mobile HCI 2004 conference, Glasgow, Scotland. Lecture Notes in Computer Science, Berlin, Springer-Verlag*.
8. Mobile Museum. Modelagem de IHC do Sistema de Apoio à Visitação. *Relatório Interno. Dezembro, 2003*.
9. Moraes, M.C., Bertolotti, A.C.; Costa, A.C.R. Evaluating Usability of SAGRES Virtual Museum Considering Ergonomics Aspects and Virtual Guides. *Proceedings of 7th World Conference on Computers in Education: Networking the Learner, 2001*.
10. Museu de Ciências e Tecnologia da PUCRS. <http://www.mct.pucrs.br/>
11. Nielsen, J. *Usability Engineering*. AP Professional, London, 1993.
12. Pascoe, J., Ryan, N., Morse, D. Usin while moving: HCI issues in fieldwork environments. *ACM Transactions on Computer-Human interaction, Vol.7, No.3, September 2000, pages 417-437*.
13. Prates, R.O., de Souza, C.S., Barbosa, S.D.J. A method for evaluating the communicability of user interfaces. *Interactions, Vol.7, No.1, pp.31-38. 2000*.
14. Rubin, J. *Handbook of Usability Testing: How to Plan, Design e Conduct Effective Tests*. John Wiley & Sons, New York, 1994.
15. SAGRES – Museu Virtual. <http://sagres.mct.pucrs.br/>
16. The UEGroup. <http://www.theuegroup.com/usability/usability.htm>
17. Weiss, S. *Handheld Usability*. John Wiley & Sons, Ltd, West Sussex, England, 2002.

Como conhecer usuários através do Método de Explicitação do Discurso Subjacente (MEDS)

Ana Maria Nicolaci-da-Costa^{1,2}
anicol@psi.puc-rio.br

Carla Faria Leitão^{1,2}
cfaria@inf.puc-rio.br

Daniela Romão-Dias¹
danirom@globocom.com

1. Departamento de Psicologia da PUC-Rio

Rua Marques de São Vicente, 225, Gávea, Rio de Janeiro – RJ, CEP: 22453-900
(21) 31141183

2. SERG – Departamento de Informática da PUC-Rio

Rua Marques de São Vicente, 225, Gávea, Rio de Janeiro – RJ, CEP: 22453-900
(21) 31141500, ramal 3323

RESUMO

Este artigo tem dois objetivos principais. Em primeiro lugar, visa apresentar ao público de IHC uma breve revisão das principais características dos chamados métodos qualitativos de pesquisa nas ciências humanas e sociais. Esta revisão serve de pano de fundo para o segundo objetivo, ou seja, para a apresentação dos pressupostos teóricos bem como das etapas de operacionalização de um método qualitativo específico: o Método de Explicitação do Discurso Subjacente – MEDS. Este método foi concebido na área de psicologia clínica com o intuito de tornar visíveis aspectos invisíveis da “configuração interna” de homens, mulheres e crianças contemporâneos. Vem, também, sendo empregado em IHC para identificar características de usuários que são de grande interesse para o desenvolvimento de software e interfaces, tais como gostos, aversões, desejos, expectativas, dificuldades, etc., principalmente quando estes são de difícil explicitação verbal. Por último, à guisa de sugestão para trabalhos futuros, é discutida a possibilidade, ainda sob estudo, de complementação quantitativa dos resultados gerados pelo MEDS.

Palavras-chave

Métodos qualitativos, MEDS, usuários.

ABSTRACT

The present paper has two main objectives. Firstly, it aims at presenting the HCI public with a brief review of the main characteristics of the so-called qualitative research methods in the human and social sciences. Such a review provides the background for the second objective, that is, the presentation of the theoretical presuppositions as well as of the operationalization stages of a specific qualitative method the Underlying Discourse Unveiling Method – UDUM. This method was conceived in clinical psychology with the purpose of giving visibility to invisible aspects of the “inner configuration” of contemporary men, women and children. It has also been used in HCI in order to identify users’ characteristics which are of great interest to the

development of software and interfaces, such as likes, dislikes, desires, expectations, difficulties, etc., mainly when these are hard to verbalize explicitly. Finally, as a suggestion for future work, an eventual and still under study quantitative complementation of the results generated by UDUM is discussed.

INTRODUÇÃO

O esforço constante de melhor contemplar as necessidades dos usuários de sistemas computacionais vem fazendo com que muitos profissionais de IHC se interessem por conhecer métodos que os ajudem a definir os perfis dos usuários das interfaces que projetam e/ou avaliam.

Neste artigo, em linha com a característica interdisciplinar da área de IHC, apresentamos uma contribuição da Psicologia à difícil tarefa de identificar e analisar características, expectativas e opiniões daqueles que hoje fazem uso de diferentes tipos de ferramentas computacionais. Trata-se do Método de Explicitação do Discurso Subjacente (MEDS), um método qualitativo de pesquisa desenvolvido por Nicolaci-da-Costa [15] [16] a partir de seus estudos no campo da psicologia clínica. Com este objetivo, primeiramente definiremos de modo breve e genérico o que vêm a ser os métodos qualitativos. Em seguida, descreveremos os principais pressupostos, características e etapas do MEDS propriamente dito. Finalmente, deixaremos indicada a possibilidade, ainda em estudo, de complementação quantitativa dos resultados gerados por esta metodologia.

MÉTODOS QUALITATIVOS:

ALGUMAS DE SUAS PRINCIPAIS CARACTERÍSTICAS

Nas ciências exatas, a atividade de pesquisa está hegemonicamente ligada aos métodos quantitativos de investigação. A maior parte destes pauta-se na elaboração de hipóteses mensuráveis de trabalho, na coleta de dados a partir de procedimentos rigorosos e pré-estabelecidos, no tratamento estatístico destes dados e na busca de regularidades que permitam a generalização de suas conclusões.

Nas ciências humanas e sociais, no entanto, esta hegemonia não se faz presente. Ao longo da história dessas ciências, disciplinas como a Antropologia, a Sociologia e a Psicologia reagiram a essa metodologia e propuseram métodos alternativos considerados mais

VI Simpósio sobre Fatores Humanos em Sistemas Computacionais — Mediando e Transformando o Cotidiano. Hotel Bourbon, Curitiba, 17 a 20 de outubro de 2004. UFPR, CEIHC—SBC.

adequados às suas investigações [3] [6] [8] [22]. Essa reação aconteceu em função de um ou mais dos seguintes fatores: (a) uma oposição radical à concepção de um ser humano com características universais e, conseqüentemente, abstraídas de seu contexto social de existência, (b) a alegação de que as formas de ser, agir, sentir, interagir, se organizar em grupos, etc. dos seres humanos são por demais complexas e sutis para serem reduzidas a números, (c) o desejo de atingir uma profundidade maior do que aquela que pode ser alcançada a partir de grandes amostras que sejam representativas da população sob estudo, (d) uma crítica à incapacidade dos métodos quantitativos de realizar explorações daquilo que é novo e desconhecido na medida em que trabalham a partir de hipóteses.

Esses métodos foram chamados de *qualitativos* por haverem surgido *em oposição aos métodos quantitativos* e não, como pensam alguns, por fornecerem subsídios para julgar o que é bom ou mau ou para ganhar acesso a qualidades intrínsecas ao ser humano.

É, contudo, somente a partir do final do século XX, que a metodologia qualitativa vem ganhando maior evidência em diferentes contextos científicos. (Isso porque as teorias universalistas que dominaram grande parte do cenário das ciências humanas e sociais daquele século vêm perdendo força.) Desde então, trabalhos norte-americanos [19] [23], ingleses [24] [25], franceses [8] e brasileiros [3] [22] vêm apresentando discussões sistemáticas dos métodos qualitativos de investigação. Essas discussões são realizadas a partir de diferentes perspectivas (a origem disciplinar leva cada autor a enfatizar determinados aspectos da metodologia em detrimento de outros) e com diferentes graus de profundidade (dependendo do objetivo de cada trabalho, a análise é mais voltada para uma visão panorâmica da abordagem qualitativa ou para a definição particularizada de uma ou mais de suas vertentes de pesquisa).

A despeito dessas diferenças, a literatura recente sobre a metodologia qualitativa revela algumas das principais características desta modalidade de investigação. São elas: (a) a postura de abertura e de desconhecimento do objeto de estudo adotada pelos pesquisadores que dela fazem uso, (b) a contextualização desses objetos de estudo, (c) a investigação em profundidade, (d) o tamanho reduzido das amostras utilizadas, e, finalmente, (e) a flexibilidade dos procedimentos e técnicas. Vejamos um pouco mais detalhadamente cada uma destas características.

A primeira delas diz respeito à forma por meio da qual o pesquisador se aproxima daquilo que visa estudar. Na maior parte das pesquisas qualitativas, o pesquisador adota uma *postura de abertura e de desconhecimento* frente a seu objeto de estudo. Os métodos qualitativos geralmente deixam de lado quaisquer hipóteses prévias [3] [19] [20] e, desse modo, tornam-se sensíveis ao que de novo e imprevisível o trabalho de campo pode vir a apresentar. Tão livre quanto possível de pressuposições, preconceitos e juízos de valor (a liberdade total nunca pode ser alcançada), o pesquisador busca compreender as experiências de outras pessoas com atenção e curiosidade

[8]. Seus objetivos centrais são a identificação, descrição e análise destas experiências e dos significados a elas atribuídos, de forma livre e espontânea, pelos participantes da pesquisa.

Já a segunda característica refere-se à *contextualização do objeto de estudo*, ou seja, à investigação de experiências humanas no contexto específico de sua ocorrência [3] [20] [22]. No mais das vezes, as pesquisas qualitativas partem do pressuposto de que as formas de ser, de agir e de pensar dos seres humanos, além de serem múltiplas e heterogêneas, são também referidas a contextos sociais, culturais e históricos específicos. Assim sendo, o conhecimento obtido neste tipo de pesquisa é reconhecidamente parcial, situado e relativo, ao invés de universalista, como é o caso de grande parte do saber gerado pela metodologia quantitativa [20]. Em outras palavras, ao invés de almejam identificar atributos universais de seres humanos abstraídos de seus contextos de vida, as pesquisas qualitativas se caracterizam por investigar seres humanos reais situados em seus contextos sociais, econômicos, históricos e culturais de existência.

Nestes contextos, entra em cena a terceira característica que grande parte das pesquisas qualitativas têm em comum: a *profundidade da investigação*. Diferentemente das pesquisas quantitativas, que buscam um conhecimento amplo e genérico (e, por conseqüência, forçosamente superficial e panorâmico), as pesquisas qualitativas tendem a privilegiar a minúcia do conhecimento contextualizado.

A profundidade e contextualização almejadas, por sua vez resultam na quarta característica que as pesquisas qualitativas têm em comum: o *tamanho reduzido das amostras*. As pesquisas qualitativas trabalham com *amostras pequenas*, se comparadas às amostras estatisticamente representativas utilizadas nos estudos quantitativos [6] [20] [22] [23]. Isso porque, entre outros fatores, as pesquisas qualitativas são, em geral, extremamente trabalhosas (“*labour-intensive*”) e de lenta execução [20]. Cada um de seus passos – a delimitação precisa de seus objetivos, o conhecimento do contexto no qual a questão de estudo se insere, a coleta minuciosa dos dados, a análise artesanal, aprofundada e iterativa do material coletado, etc. – envolve freqüentes tomadas de decisão e denso trabalho intelectual. Por essas razões, as pesquisas qualitativas costumam trabalhar intensivamente com poucos participantes ao invés de extensivamente com grandes amostras.

A *flexibilidade dos procedimentos e técnicas utilizados* na pesquisas qualitativas é a quinta e última característica comum a ser detalhada. Significa que, ao invés de adotar um padrão único de pesquisa estabelecido a priori (o que é usualmente feito na metodologia quantitativa), a abordagem qualitativa abriga diferentes procedimentos e técnicas, definidos e adaptados em função dos objetivos de cada estudo [3] [8] [20] [22]. No que diz respeito ao levantamento de dados, por exemplo, dentro do conjunto heterogêneo de modalidades de pesquisa qualitativa, destacam-se os seguintes procedimentos e técnicas: (a) a *observação participante*, que implica a inserção ativa do pesquisador no contexto natural de ocorrência do

fenômeno estudado [3] [8] [22], (b) a *análise documental*, que estuda de forma sistemática uma seleção de materiais publicados em livros, jornais, revistas, na Internet, etc. [3] [8] [20], (c) a *história de vida*, que faz uso da autobiografia para identificar processos sociais e psicológicos representativos de uma época ou de um determinado grupo [2] [3], (d) a *pesquisa-ação*, que faz coincidir a análise de um determinado problema com um conjunto de intervenções que visam a sua solução [3] [21] e (e) as *entrevistas semi-estruturadas*, nas quais o entrevistador faz uma série de perguntas abertas que o entrevistado pode responder livremente como lhe aprouver [8] [19] [20] [22] [23]. Esta última técnica – a das entrevistas semi-estruturadas – é uma das mais freqüentemente utilizada nas pesquisas qualitativas no campo das ciências humanas e sociais (ver a seguir).

Tendo em vista todo esse conjunto de características, os métodos qualitativos mostram-se especialmente sensíveis à identificação e descrição de novos comportamentos e experiências humanas. Por isso mesmo, vêm se revelando particularmente úteis para gerar conhecimento sobre mudanças radicais como as que enfrentamos no mundo atual. Como o próprio tema deste Simpósio revela, homens, mulheres e crianças contemporâneos estão alterando seus modos de ser, de agir e de pensar a partir das profundas transformações ocorridas em seu cotidiano (muitas delas desencadeadas pela difusão das tecnologias computacionais) [10]. No que se segue, apresentaremos uma modalidade de investigação qualitativa que vem sendo utilizada para registrar e analisar sua nova “configuração interna”. Trata-se do Método de Explicitação do Discurso Subjacente (MEDS).

O MÉTODO DE EXPLICITAÇÃO DO DISCURSO SUBJACENTE (MEDS)

Características gerais

O MEDS é um método *exploratório* desenvolvido para a pesquisa em psicologia clínica. Tem em comum com os demais métodos qualitativos a abertura, a investigação aprofundada em contexto, as amostras pequenas e a flexibilidade de procedimentos e técnicas.

A principal diferença entre o MEDS e grande parte dos demais métodos qualitativos é o registro explícito de que trabalha com *material discursivo* (característica que divide com alguns outros métodos de análise de discurso [24]), aliado a uma concepção de que a *língua em contexto* – o discurso – desempenha um importante papel na *construção social da esfera do psicológico*.

É um método fortemente interdisciplinar. Seu desenvolvimento sofreu uma grande influência da lingüística, das teorias da construção social da subjetividade, bem como de alguns pressupostos e técnicas de investigação clínica oriundos da psicanálise.

Da lingüística, o MEDS absorveu os preceitos de: (a) trabalhar com uma concepção de linguagem explícita [16] [20], (b) coletar discursos em *settings* naturais e informais [9], e (c) considerar as transcrições do discurso coletado (quando oral) como uma forma de seleção do material a ser analisado [15] [20] e (d) realizar uma análise sistemática do material coletado tendo plena consciência que se trata de material discursivo.

Das teorias da construção social da subjetividade [1] [7] incorporou um pressuposto que se opõe ao do universalismo predominante nas pesquisas quantitativas: o de que *a forma de percebermos o mundo bem como os modos de pensarmos, agirmos e sentirmos são socialmente construídos* (estando, portanto, sempre referidos a um contexto social específico).

Por último, embora não adotando uma perspectiva psicanalítica, o MEDS inspirou-se em algumas contribuições da psicanálise: (a) nas técnicas clínicas da *livre escuta* que, por não ser intrusiva, capta aquilo que é importante para o *outro*, (b) no pressuposto de que geralmente há algo não verbalizado e/ou não consciente por trás do que dizemos (um significado subjacente), (c) na crença de que aquilo que é importante para alguém aparece no seu discurso espontâneo.

Tentando resumir, o MEDS usa a língua em contexto, ou seja, o discurso, como *via de acesso* às características internas de homens, mulheres e crianças. Esta via de acesso é legitimada pela concepção de que, ao internalizarmos uma língua nos contextos em que ela é naturalmente usada, internalizamos todo um conjunto de conceitos, regras, valores, etc. que caracterizam uma *determinada sociedade ou grupo social*. O corolário dessa afirmação é o de que, se o discurso constrói, o discurso também revela. Para que isso aconteça, é, no entanto, necessária a aplicação *sistemática* de todo um conjunto de procedimentos, guiados pelos princípios gerais ditados pelo método, a ser estabelecido em função do objetivo da investigação. (Esses procedimentos serão discutidos em seguida.)

Dentro da psicologia, o MEDS tem sido utilizado para trazer à tona as transformações e os conflitos de cunho interno resultantes de mudanças sociais geradas por fatores econômicos, políticos, tecnológicos, etc. [13] [14].

Já em IHC, seu principal objetivo tem sido o de tornar visíveis aspectos da configuração interna de homens, mulheres e crianças que são importantes para o desenvolvimento de software e interfaces: gostos, preferências, aversões, dificuldades, aspirações, desejos, etc. Nesta área, sua maior utilidade reside no fato de poder captar aquilo que, por ser invisível inclusive para os próprios homens, mulheres e crianças em questão, muitas vezes passa despercebido por outros métodos.

Operacionalização

O MEDS faz uso de procedimentos ao mesmo tempo flexíveis e rigorosos. Esses procedimentos são flexíveis porque, como veremos a seguir, são adaptáveis a diferentes objetos de estudo mediante a introdução de alterações que preservam as características gerais do método. São também rigorosos porque, uma vez estabelecidos e explicitados, esses procedimentos não permitem outras alterações bem como devem ser aplicados sistematicamente.

Tal como outros métodos qualitativos, o MEDS não parte de hipóteses ou categorias pré-estabelecidas. Muito pelo contrário, é um método *exploratório* que parte de *perguntas abertas*. Por essa razão, é particularmente adequado à investigação daquilo que é novo e/ou

desconhecido. No que se segue serão discutidos os princípios gerais de suas diversas etapas.

1 - Delineamento do objetivo:

Esta é uma etapa fundamental porque dela dependerão várias das decisões a serem tomadas no planejamento das outras etapas da pesquisa. De modo a possibilitar a investigação em profundidade da questão proposta, este objetivo deve ser nítido e compacto. Sugere-se que seja colocado sob a forma de uma pergunta aberta. Damos um exemplo.

Um pesquisador está diante da tarefa de desenvolver um sistema de apoio à decisão para uma determinada empresa, mas desconhece as necessidades e expectativas da equipe de gerentes que, no futuro, fará uso desta ferramenta. Para conhecer essas necessidades e expectativas, irá realizar uma pesquisa e enuncia seu objetivo por meio da seguinte questão: *Como você imagina uma ferramenta computacional que ajude você no processo de tomada de decisões de sua empresa?* Essa pergunta delimita a área de investigação da pesquisa, evitando que o pesquisador se perca na coleta de informações que não são relevantes para a mesma.

Uma vez delimitado o foco de sua investigação, o pesquisador poderá planejar as etapas subsequentes de modo a coletar dados a respeito das expectativas da equipe gerencial *exclusivamente naquilo que diz respeito a um sistema de apoio à decisão*. Ou seja, com essa única pergunta em mente, o pesquisador evita uma tentação muito freqüente em estudos com usuários: a de coletar o maior número de informações possíveis sobre o que esses usuários têm a dizer sobre suas interações com quaisquer tipos de ferramentas computacionais. Vale, aqui, voltar a enfatizar: um objetivo nítido estabelece os limites claros que priorizam a profundidade dos resultados a serem obtidos, ao invés de ambicionar a obtenção de dados superficiais a respeito de um conjunto heterogêneo de fatores. Quando se almeja a profundidade, perde-se em amplitude, dado que quanto mais amplo o escopo, necessariamente mais superficiais são os resultados.

2 - Recrutamento dos participantes:

O recrutamento dos participantes está intimamente vinculado ao objetivo da pesquisa. Dado que, numa pesquisa em profundidade, o número de participantes é invariavelmente pequeno, as estratégias de recrutamento requerem muita atenção. Weiss [23], por exemplo, adota o que chama de “snow ball sampling”, que consiste em recrutar sujeitos para a pesquisa, que indicam outros sujeitos e assim sucessivamente. Já Seidman [19] usa uma estratégia diferente, que denomina de “amostra de máxima variação”. Nesta, o pesquisador escolhe, em uma determinada população, tipos heterogêneos de sujeitos de modo a permitir que um número maior de pessoas possa se identificar com a pesquisa, já que ela é composta de tipos de sujeitos diferentes entre si.

No MEDS, sugere-se que seja adotada uma estratégia radicalmente oposta à de máxima variação de Seidman. Preferimos usar o que chamamos de um “perfil de alta definição” [17], ou seja, realizamos um tipo de recrutamento que maximiza a homogeneidade do grupo de participantes escolhido a partir de critérios relevantes

para a investigação proposta. Para este tipo de recrutamento, é útil fazer uso de uma rede de conhecidos que possam dar indicações de pessoas que tenham o perfil escolhido (o que não deixa de ser uma forma de utilização da estratégia de bola de neve de Weiss [23]).

Alguns exemplos podem ajudar o leitor visualizar o processo de delineamento do “perfil de alta definição” nas pesquisas que fazem uso do MEDS. Retomemos, primeiramente, a pesquisa do sistema de apoio à decisão anteriormente citada. Nesta, os critérios de recrutamento foram estabelecidos *a priori* e de modo praticamente automático. Isto porque, uma vez que o sistema a ser projetado deveria ser altamente customizado para um grupo de gerentes de uma empresa específica, os participantes poderiam ser recrutados somente a partir desse grupo de gerentes.

Mais freqüentemente, no entanto, esses critérios não são tão óbvios. Este é o caso, por exemplo, de uma pesquisa cujo objetivo é avaliar, da perspectiva dos usuários, sites de compras online em supermercados. Diferentes perfis de usuários compõem a clientela desses sites (homens jovens e solteiros, mulheres maduras, casadas e com filhos, homens divorciados cujos filhos os visitam somente no fim de semana, etc.). É muito provável que cada um desses grupos de usuários tenha necessidades, hábitos e preferências também específicas. Em busca de um foco nítido, o pesquisador deve optar por um dos vários perfis de usuário. É somente ao assumir explicitamente esta opção que ele pode conhecer em profundidade como os membros de *um determinado grupo com características bem definidas* fazem compras de supermercados online, o que procuram, aquilo de que gostam nos sites, aquilo de que não gostam, etc.

3 - Preparação para a coleta de dados:

A coleta de dados é uma das mais importantes etapas da pesquisa. Como já mencionado no início do artigo, as pesquisas qualitativas tendem a adotar uma variedade de técnicas para essa coleta: observação participante, entrevistas semi-estruturadas ou abertas, seleção de material publicado em mídia digital (internet ou intranet) ou impressa (livros, jornais ou revistas), etc.

No caso do MEDS, tradicionalmente é dada preferência (mas não exclusividade) às entrevistas semi-estruturadas (caracterizadas por uma estrutura flexível, como veremos abaixo) de perguntas abertas – presenciais ou via qualquer programa interativo em tempo real – realizadas a partir de um roteiro construído com base no objetivo da pesquisa. O MEDS possibilita, no entanto, a utilização de outros instrumentos de coleta de dados. Este método já foi utilizado, por exemplo, em pesquisas que fizeram uso de questionários abertos, enviados e respondidos via e-mail por diferentes tipos de usuários, como no caso de Nicolaci-da-Costa [14] e da Silva et al [4]. Foi, ainda, empregado para analisar material publicado online (no caso, o discurso dos blogs) em, por exemplo, Prange [18].

Em todos os casos, a escolha do instrumento de coleta de dados deve atender aos objetivos da pesquisa e levar em conta os princípios gerais do MEDS, como, por exemplo, os de que os discursos devem ser coletados em *settings*

naturais e informais para que possam ser espontâneos (ao menos tanto quanto possível).

Embora várias técnicas possam ser utilizadas, no que se segue, privilegiaremos as entrevistas (tanto presenciais quanto via programas interativos síncronos). Isso porque, além de muito planejamento e treinamento do entrevistador, qualquer entrevista requer uma série de escolhas e decisões cujos princípios devem ser explicitados.

3.1 - Entrevistas presenciais:

As entrevistas presenciais devem ter como modelo uma *conversa natural* entre duas pessoas (a presença de mais de um entrevistador pode inibir o entrevistado e dois ou mais entrevistados podem querer se mostrar um para o outro, entrar em competição, etc.). Para que uma conversa desse tipo gere as informações das quais o pesquisador necessita, ela tem que ser preparada com esse intuito. Para tanto, é necessário que seja elaborado um roteiro tendo em mente o objetivo da pesquisa.

3.1.1 - Elaboração do roteiro:

Muitas técnicas de entrevista, a exemplo da técnica de história de vida, na qual pode-se simplesmente pedir ao entrevistado que conte sua vida [2], dispensam o roteiro. Outras usam-no de forma pouco sistemática, permitindo que novas perguntas sejam nele incluídas uma vez começadas as entrevistas [19] [23] (o que pode gerar sérios problemas, pois perguntas diferentes não permitem as necessárias comparações durante a fase de análise dos dados).

O MEDS, no entanto, prevê a elaboração de um roteiro de *itens abertos* (e não de perguntas prontas). Os itens são simplesmente lembretes para o entrevistador. Lembrem-no de, *durante a entrevista*, fazer perguntas sobre os tópicos do roteiro sem que estas perguntas sejam lidas e, em consequência, soem formais e artificiais para os entrevistados. *Todos os tópicos previstos no roteiro devem ser abordados em todas as entrevistas*. Este procedimento deve ser sistemático e rigoroso porque dele dependerá a qualidade da análise a ser realizada posteriormente.

Um pequeno extrato de um roteiro de entrevista pode tornar visível a estrutura de itens abertos a ser preparada pelo pesquisador. No estudo em questão, os pesquisadores tinham como objetivo investigar o processo de adesão a um bulletin board (BB) em uma empresa de pequeno porte. O sistema implantado era bastante simples, tendo em vista o perfil de seus usuários (funcionários com pouca ou nenhuma familiaridade com tecnologias computacionais). O roteiro construído para este fim era composto de duas partes distintas. A primeira consistia na coleta de dados de identificação do entrevistado (no caso, idade, setor em que trabalhava e cargo), enquanto que a segunda era composta por 6 blocos temáticos, cada um deles subdividido em vários itens. No quadro que se segue, estão listados todos os blocos temáticos e *alguns* itens destes blocos, de forma a permitir a visualização das principais características do roteiro de entrevista comumente utilizado no MEDS.

Quadro 1 – Roteiro de Entrevista sobre Adesão ao Bulletin Board (BB)

BLOCOS TEMÁTICOS	PRINCIPAIS ITENS
I – Sobre a divulgação de notícias antes do BB...	<ul style="list-style-type: none"> • Antes do BB, como recebia e divulgava informações sobre a empresa (informações de que tipo, através de quais meios).
II – Sobre as expectativas em relação ao BB...	<ul style="list-style-type: none"> • Quando e como ouviu falar do BB pela primeira vez. • O que imaginou a partir disto. <ol style="list-style-type: none"> a. A respeito da ferramenta. b. A respeito das facilidades e dificuldades que o BB apresentaria. c. A respeito dos tipos de notícia que esperava encontrar e divulgar.
III – Começando a usar o BB...	<ul style="list-style-type: none"> • Como começou a usar o BB (iniciativa própria ou após treinamento), por quê. • Caso tenha participado do treinamento, quais os pontos positivos e negativos deste. • Caso tenha sido por iniciativa própria, verificar se precisou de alguma ajuda e, em caso positivo, como a procurou.
IV – Comparando as expectativas com o que encontrou...	<ul style="list-style-type: none"> • que no BB atendeu suas expectativas, por quê. • que não atendeu suas expectativas, por quê. • Comparar as notícias que esperava encontrar com as que encontrou.
V – Usando no dia-a-dia...	<ul style="list-style-type: none"> • Frequência de acesso. • que procura e como procura. • Facilidades e dificuldades encontradas. • Mudanças na comunicação da empresa a partir do uso do BB.
VI – Olhando para o futuro...	<ul style="list-style-type: none"> • Como imagina o BB daqui a alguns meses. • Pontos a aprimorar na ferramenta.

O roteiro acima foi elaborado adotando uma ordenação dos itens que procura seguir o fluxo de um bate-papo informal. Esse fluxo pode, no entanto, seguir diferentes lógicas, como, na realidade, acontece nos nossos bate-papos cotidianos. No roteiro acima, o fluxo segue uma ordem cronológica, ou seja, parte de um período anterior à implantação do sistema e acompanha gradualmente as experiências do entrevistado ao longo das etapas desta implantação. Em outros roteiros, a seqüência poderá seguir outra lógica. Poderá, por exemplo, ir do genérico ao específico. Em todos os casos, no entanto, apesar de os itens serem dispostos em uma seqüência lógica, esta seqüência não necessariamente será aquela adotada durante a entrevista. Isso porque toda conversa tem um fluxo característico e esse fluxo natural deve ser respeitado. Um entrevistado pode, por exemplo, dar respostas a vários itens do roteiro sem que lhe sejam feitas as perguntas correspondentes. Neste caso, as perguntas não lhe devem ser colocadas de novo. (Para isso, o entrevistador tem que estar atento ao que o entrevistado diz!). Outro entrevistado pode, ao responder a uma pergunta que está no início de um bloco temático, incluir a resposta a outra prevista para o final da entrevista e o entrevistador deverá saber acompanhá-lo.

3.1.2 - Elaboração do Termo de Consentimento:

Além do roteiro de entrevista, o pesquisador deve preparar o Termo de Consentimento, documento que contempla as diretrizes éticas da pesquisa e que deve ser assinado por cada um dos entrevistados e pelo entrevistador. Deste termo, devem constar os objetivos gerais da pesquisa, os procedimentos envolvidos (a realização da entrevista, sua gravação em áudio, etc.) e as garantias dadas ao entrevistado (de respeito à sua participação voluntária, de anonimato e de confidencialidade). O Termo de Consentimento deverá ser fornecido a todos os participantes antes do início das entrevistas [11].

3.1.2 - Realização de entrevistas-piloto:

Este procedimento é adotado por vários pesquisadores [19] [22] [23]. Se o grupo a partir do qual podem ser recrutados os participantes não for muito pequeno (como poderia acontecer no caso dos gerentes citado anteriormente), o ideal é testar o roteiro em entrevistas-piloto, que não farão parte do conjunto de entrevistas a ser analisado. Nas entrevistas-piloto, é possível fazer várias coisas ao mesmo tempo: (a) treinar o entrevistador inexperiente, (b) testar a adequação do roteiro ao objetivo da pesquisa (muitas vezes julgamos que o roteiro está adequado e este teste mostra que não está), e (c) testar a sua fluidez (um bom roteiro é aquele que flui de um item a outro numa lógica análoga à de uma conversa informal).

O número de entrevistas-piloto necessárias vai depender do quanto o entrevistador está treinado bem como da qualidade do roteiro. Quando as entrevistas começarem a fluir bem é, em geral, a hora de encerrar esta etapa.

3.2 – Entrevistas online

As entrevistas online pressupõem as mesmas etapas de planejamento, preparação e construção de roteiro que as entrevistas presenciais. Para que tenham o tom de

espontaneidade e informalidade previstos pelo MEDS, essas entrevistas devem também levar em conta a familiaridade que o pesquisador e os participantes têm com o instrumento adotado (ICQ, Messenger, chats, etc.). Tanto o entrevistado quanto o entrevistador têm que estar à vontade com a mediação tecnológica escolhida. O entrevistador deve também ter consciência das especificidades deste tipo de entrevista.

A título de exemplo, a condução de uma entrevista online costuma exigir mais tempo disponível (digitar é uma ação mais lenta do que a fala), maior esforço de concentração dos envolvidos (de modo a evitar a realização de outras tarefas simultaneamente) e uma atenção especial à privacidade (a escolha do ambiente computacional deve considerar cuidadosamente a confidencialidade das comunicações entre entrevistado e entrevistador).

4 - Coleta de dados:

Uma vez definidos o perfil, o ambiente (físico ou virtual), a forma de recrutamento dos participantes bem como construído e testado o roteiro, a coleta de dados pode ter início. Esta também pode ser subdividida em etapas.

4.1 - Convite para participação:

O convite deve ser feito da forma mais natural possível. Face a face, se houver esse tipo de contato, por e-mail, por telefone, etc. Nesta fase, o entrevistador deve ter o cuidado de dar ao futuro entrevistado todas as informações sobre a pesquisa que este desejar, enfatizando os procedimentos éticos do estudo (a participação voluntária, a preservação do anonimato dos participantes, bem como a possibilidade de interrupção da participação em qualquer momento da entrevista).

No caso das entrevistas face a face, o pesquisador deve, também, perguntar ao entrevistado onde este gostaria que fosse realizada a entrevista. Neste caso, suas preocupações devem ser as de realizar a entrevista onde o entrevistado se sentir à vontade e, ao mesmo tempo, evitar lugares muito ruidosos por conta da necessária gravação em áudio.

Já no caso das entrevistas online, as preocupações do entrevistador devem se concentrar na questão dos horários mais convenientes para os participantes. O entrevistado deve poder escolher o horário no qual deseja conceder a entrevista (mesmo que seja muito tarde à noite ou até de madrugada).

4.2 - Realização das entrevistas:

Quer sejam face a face quer sejam online, as entrevistas devem ser sempre realizadas em contextos que sejam naturais tanto para o entrevistador quanto para o entrevistado.

As entrevistas presenciais da pesquisa sobre o processo de adesão a um bulletin board cujo roteiro apresentamos anteriormente foram, por exemplo, realizadas em locais diversos, por escolha dos diferentes entrevistados. Alguns quiseram realizá-las em suas próprias salas de trabalho, enquanto outros preferiram locais mais reservados (um parque, um restaurante tranquilo, etc.) para que se sentissem livres para falar sobre situações mais sigilosas de seu cotidiano profissional.

No caso das entrevistas face a face, a escolha dos locais, a flexibilidade da seqüência dos itens do roteiro e a formulação das perguntas no momento da entrevista, vale repetir, têm um único intuito: o de fazer com que a entrevista se assemelhe tanto quanto possível a uma conversa natural em um contexto informal [9].

A diferença fundamental entre a entrevista (presencial ou online) e uma conversa (presencial ou online) reside no fato de que o entrevistador deve se restringir a fazer perguntas e escutar (ou ler), solicitando apenas informações e exemplos que complementem, ilustrem ou aprofundem os depoimentos dos participantes. O entrevistador deve, sobretudo, evitar sugerir ou completar respostas, direcioná-las e colocar palavras na boca (ou nos dedos) do entrevistado. Deve também evitar interrupções desnecessárias. Sugere-se ainda que, sempre que possível, reduza-se a um mínimo a presença de recursos tecnológicos, pois estes podem deixar o entrevistado desconfortável. Registros de vídeo, por exemplo, costumam inibir o entrevistado desnecessariamente, pois acrescentam informações que não podem ser analisadas com os recursos oferecidos pelo método (ou seja, com os recursos da análise de discurso que serão discutidos a seguir).

No MEDS, prevê-se a realização de uma única entrevista com cada entrevistado. (Antes de seu início, deve-se proceder à assinatura do Termo de Consentimento.) Essas entrevistas tendem a ter, em média, cerca de uma hora de duração, no caso de serem realizadas face a face, e uma hora e meia ou duas, no caso de serem feitas online. Como já foi dito, as entrevistas devem seguir o modelo um-a-um, um entrevistador e um entrevistado, na medida em que se quer evitar que a presença de mais de um entrevistador intimide o entrevistado ou que este não se pronuncie espontaneamente por conta da presença de outro entrevistado. Todas as entrevistas devem ser gravadas (o gravador rapidamente se torna invisível), a não ser que o entrevistado não permita a gravação (o que raramente acontece). O entrevistador também deve registrar – no papel – tudo o que de inesperado e não passível de registro em áudio acontecer ao longo da entrevista (sorrisos e expressões faciais significativas, gestos indicativos de ansiedade, etc.).

5 - Preparação para a análise dos dados:

Os discursos coletados nas entrevistas face a face devem ser transformados em texto. Para tanto devem ser transcritos. (As entrevistas realizadas online são atraentes porque evitam o trabalho de transcrição!) A transcrição de uma entrevista não é, no entanto, uma atividade automática ou automatizável. Ela requer que sejam estabelecidas prioridades, novamente de acordo com os objetivos da pesquisa [20].

Para ilustrar o tipo de cuidado que a transcrição envolve, podemos citar o registro da ocorrência de uma hesitação ou de uma pausa mais longa na fala do entrevistado. Ocorrências como essas, podem ser, na etapa da análise de dados, importantes indicadores de conflitos, dúvidas, defesas, etc. Já outras minúcias podem atrapalhar mais do que ajudar. Menos muitas vezes é mais e, por isso, detalhes em excesso devem ser evitados. Uma transcrição

é sempre longa e fazer sentido de muitas transcrições simultaneamente requer muito trabalho de limpeza de detalhes superficiais do texto para que se possa atingir as entrelinhas do discurso, ou melhor, aquilo que está por trás dele.

6 - Análise dos dados

Praticamente todas as formas de coleta de dados usadas nas pesquisas qualitativas geram material discursivo. Raramente, no entanto, os pesquisadores por elas responsáveis revelam ter consciência de que estão trabalhando com material discursivo [15] [16]. Isso, por sua vez, impede que prestem atenção a certas características do discurso dos entrevistados que podem ser importantes. Damos um exemplo. Na maior parte das vezes, esses pesquisadores [19] [22] [23] fazem análises temáticas ou de conteúdo sem se darem conta de que a *forma* (lingüística) também carrega significado. A pergunta que fica sem resposta é o que fazem com certos tipos de recorrências discursivas como, por exemplo: (a) novas formas de falar ou escrever como os textos abreviados dos chats (que podem trazer à tona novas identidades de grupo) ou o uso de letras maiúsculas em emails (que significam que a pessoa está gritando); (b) expressões inusitadas como, por exemplo, a recente “Estou no celular” (que pode ser indicativa de alterações nas concepções de lugar e de espaço); (c) usos pronominais como a insistência em usar o pronome “eu” (que pode indicar um discurso excessivamente auto-referido, a exemplo do discurso de alguns blogs [18] ou o uso recorrente do pronome “nós” para se referir a preferências individuais logo após um divórcio (uso que pode ser usado como indicador da dificuldade de o entrevistado se ver como um simples “eu”); (d) os de um mesmo significado a despeito do uso de diferentes palavras (ou outras unidades lingüísticas), como no caso do emprego dos adjetivos “produtivo”, “qualificado”, “bom”, “correto”, “sadio”, para indicar o que os entrevistados vêem como o mesmo uso *ideal* da Internet (o que revela a existência de um valor social [12]). Todas essas formas e unidades lingüísticas carregam tanto significado que é possível tomá-las como base de análises muito interessantes e reveladoras.

Diferentemente do caso daqueles pesquisadores que não registram que estão trabalhando com material lingüístico e por isso fazem o que poderia ser chamado de uma *análise implícita de discurso*, o MEDS se caracteriza por fazer uma *análise explícita de discurso*. Por esse motivo, pode mais facilmente levar em consideração quaisquer aspectos lingüísticos e paralingüísticos recorrentes nos discursos coletados – forma, conteúdo, sintaxe, entonação, pausas, hesitações, etc. – desde que sejam relevantes para os objetivos da pesquisa.

O MEDS parte do pressuposto de que qualquer característica lingüística ou paralingüística que seja recorrente nos discursos dos participantes de uma pesquisa pode ser uma importante via de acesso a aspectos de nossa configuração interna – desejos, aspirações, conflitos, etc. – que nós próprios muitas vezes desconhecemos. Tal como a febre é um indicador de algo invisível (uma infecção), essas características recorrentes

são *indicadores* de algo invisível que se quer tornar visível.

Assim sendo, no MEDS, todo o trabalho executado nas etapas que antecedem à análise do discurso coletado serve de base para um trabalho análogo ao de um detetive, o de usar as pistas – os indicadores – disponíveis para trazer à tona o discurso subjacente e seu significado. Essa análise é realizada em duas grandes etapas.

A primeira é a da análise inter-sujeitos, na qual são reunidas *todas* as respostas de *todos* os sujeitos a *cada um* dos itens do roteiro (esta análise deve ser feita *somente quando todas as transcrições já estiverem disponíveis* na medida em que é importante se tenha uma noção do conjunto de respostas). Embora a análise inter-sujeitos seja muitas vezes a única a ser utilizada por muitos pesquisadores [19] [20] [23], a análise inter-sujeitos realizada no MEDS tem características particulares. E isso acontece porque, no MEDS, diferentemente do que acontece em outros métodos [22], *todas* as respostas de *todos* os sujeitos a *todas* as perguntas são analisadas sistematicamente e rigorosamente, o que é possível somente porque *todos* os sujeitos deram

suas respostas a *todos* os itens do roteiro durante as entrevistas.

No caso do MEDS, mas não no de outros métodos, este procedimento, que propicia uma visão panorâmica dos depoimentos gerados por cada um desses itens, somente nos aponta os primeiros resultados – ou seja, revela as tendências centrais das respostas dadas pelo grupo como um todo –, embora estes ainda sejam inconclusivos. São inconclusivos porque o MEDS prevê a realização de uma segunda etapa: a da análise intra-sujeitos. Nesta, tomam-se as respostas de cada um dos sujeitos como um único conjunto dentro do qual são analisados possíveis conflitos de opiniões, inconsistências entre respostas, sentimentos contraditórios, etc. Com o *insight* ganho nesta segunda etapa, retorna-se à primeira e reanalisam-se os conjuntos das respostas dadas por todos os sujeitos a cada uma das perguntas. Este vai-e-vem (que pode ser repetido quantas vezes necessário) permite que o material coletado seja dominado a fundo e torna possível detectar, além de vários não-ditos (pois o que não é verbalizado em uma resposta pode ser claramente sugerido em outra), os pormenores de sentimentos, conflitos internos e porquês que a pesquisa se propõe a revelar.

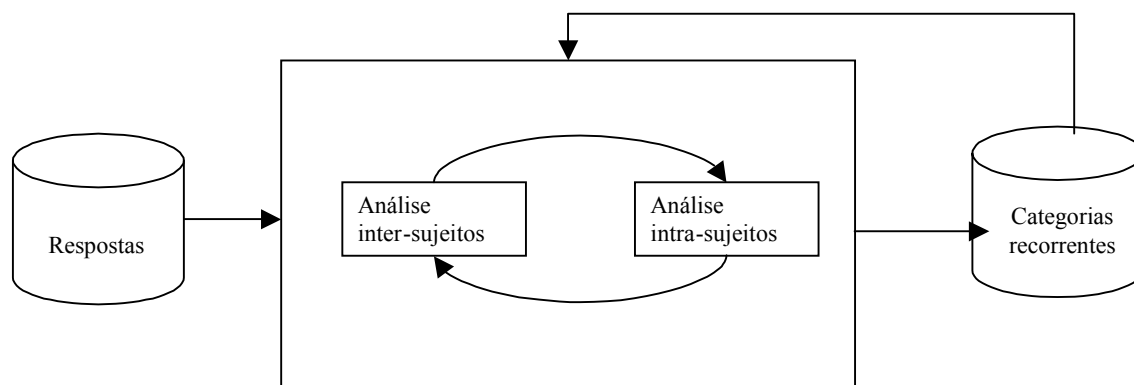


Figura 1: Uma visão esquemática dos procedimentos do MEDS [5]

7 - Interpretação dos resultados

Como indica a figura 1, no MEDS, as recorrências nos depoimentos dos participantes levam o pesquisador a propor categorias de análise para os seus resultados. Ou seja, diferentemente do caso de outras metodologias que partem de categorias de análise previamente estabelecidas, no MEDS essas categorias emergem dos discursos dos entrevistados. Segue-se um exemplo.

Uma pesquisa teve como objetivo conhecer as expectativas dos participantes de um grupo de discussão acadêmica presencial frente à possibilidade de seus encontros migrarem para o ambiente online [4]. Da análise do material gerado pelos participantes, emergiram várias categorias de análise. A título de ilustração, citamos duas delas: a preocupação com a apresentação do material das discussões (o grupo espontaneamente trouxe à tona o problema do excesso das informações comumente disponibilizadas em ambientes online) e o controle da privacidade (os participantes temiam que a ferramenta computacional interferisse no controle

individual e coletivo da privacidade das discussões do grupo).

O significado dessas categorias obviamente vai depender da pesquisa. Algumas linhas gerais para a interpretação das recorrências discursivas, de acordo com o MEDS, podem, no entanto, ser explicitadas. Todas estão intimamente vinculadas aos pressupostos teóricos do método, anteriormente discutidos:

1) As recorrências no conjunto de depoimentos dos participantes (etapa inter-sujeitos) permitem tornar visíveis as características internas geradas pela experiência em um determinado grupo social. Quanto mais estiverem presentes mais serão indicativas de que foram socialmente construídas.

No caso do estudo sobre a migração de um grupo de discussão para o ambiente online, a categoria do controle de privacidade é um ótimo exemplo de como determinadas experiências sociais geram impactos sobre o mundo interno das pessoas. Os participantes da

pesquisa revelaram recorrentemente o quanto temiam que o controle que mantinham sobre os diferentes níveis de privacidade de suas vidas fosse afetado por questões tecnológicas.

2) As recorrências no discurso de cada um dos sujeitos (etapa intra-sujeitos), por sua vez, levam à identificação daquilo que é importante individualmente.

No exemplo que estamos utilizando, recorrências da categoria controle de privacidade em diferentes momentos do depoimento de um determinado participante enfatizam a importância que esta questão assumiu para ele e, também (como vimos na etapa inter-sujeitos), para o grupo.

3) Já as contradições no discurso de cada um dos sujeitos (etapa intra-sujeitos) frequentemente revelam conflitos internos entre discurso e prática, bem como entre diferentes valores, visões de mundo, gostos, preferências, sentimentos, etc. Em tempos de mudança, esses tipos de conflito são muito comuns, na medida em que, valores mais recentemente adquiridos, muitas vezes contradizem aqueles tradicionais que foram absorvidos em épocas *discursos que, mediante cuidadosa análise, revelem indicadores daquilo que não nos é possível ver, perceber e muito menos verbalizar a respeito de nós mesmos*: nossas opiniões, gostos, preferências, aversões e conflitos gerados pela experiência com uma *nova* realidade. Esta realidade é multifacetada e deve ser investigada complementarmente – de preferência interdisciplinarmente – por pesquisadores inseridos em diferentes áreas de saber. Na psicologia, por exemplo, o MEDS vem sendo utilizado com propósitos clínicos. Já em IHC ele tem se provado útil para gerar perfis de usuários e conhecer o que estes pensam e sentem a respeito de diferentes interfaces, quais características destas são do seu agrado ou desagradado, quais as dificuldades que essas interfaces apresentam, etc.

O MEDS, como qualquer outro método de pesquisa, tem, no entanto, limitações. Do ponto de vista de IHC, uma dessas limitações parece ser importante: o fato de os resultados do MEDS não comportarem grandes generalizações (quando é usado um “perfil de alta definição”, as generalizações são possíveis somente para aqueles que tenham exatamente o mesmo perfil e, conseqüentemente, a mesma inserção social [17]).

Isto, porém, não quer necessariamente dizer que essas generalizações não possam eventualmente ser levadas a cabo. Muitos autores [3] [22] são da opinião de que os métodos qualitativos e os quantitativos podem ser usados complementarmente. Também do nosso ponto de vista, essa complementaridade, *se realizada criteriosamente*, é em princípio possível (e, no caso de IHC, talvez desejável). Citamos, a título de exemplo, uma possibilidade, *ainda sob estudo*.

Suponhamos que o MEDS fosse usado como uma etapa exploratória destinada a gerar conhecimento das características de um determinado conjunto de usuários, características essas que o pesquisador deseja verificar se estão presentes em uma amostra representativa da população.

anteriores da vida. Algumas vezes essas próprias contradições são também recorrentes entre sujeitos [12], o que indica mudanças sociais bruscas que geram conflitos generalizados.

Este parece ter sido o caso do grupo de discussão acadêmica ao qual vimos nos referindo. Seus participantes estavam, por um lado, atraídos pelo novo, a saber, pela possibilidade de expandir suas interações para o ambiente online. Por outro, a possibilidade das relações do grupo serem mediadas por uma ferramenta computacional também despertava temores. O modelo de discussão presencial lhes era um velho conhecido, e as mudanças que a mediação de uma ferramenta computacional poderia gerar nas relações do grupo foram foco de conflitos e preocupações.

CONCLUSÃO

Como foi visto ao longo desta apresentação, o MEDS é um método qualitativo que procura *explorar em profundidade*, as características internas de homens, mulheres e crianças. Mais especificamente é um método que se propõe a *coletar*

Neste caso, poderia ser recrutada uma amostra diferente daquela do “perfil de alta definição” descrito acima. Poderia, por exemplo, ser adotado o recrutamento de uma “amostra variação máxima”, tal como proposta por Seidman [19] e discutida anteriormente. Isso porque esse tipo de amostra maximiza a heterogeneidade dos participantes da pesquisa e pode coletar – por meio do mesmo tipo de entrevistas exposto anteriormente – respostas com maior grau de diversidade.

Uma vez obtidas essas respostas, um questionário, desta feita fechado (do tipo múltipla escolha) poderia ser montado abordando os mesmos tópicos das entrevistas e usando os depoimentos obtidos na fase exploratória como alternativas de resposta. Do ponto de vista operacional, esta seria uma solução simples para obter um maior poder de generalização dos resultados, tendo em vista que um questionário pode ser facilmente respondido por uma ampla amostra de respondentes e também facilmente analisado de forma *quantitativa*.

Antes, porém, de propormos esse tipo de complementação, somos da opinião que ele deve ser objeto de cuidadosa reflexão. E a razão para isso é simples: ao escolhermos um método de pesquisa, assumimos um compromisso com as posições epistemológicas sobre as quais este método se apóia e temos que respeitar as restrições ditadas por essas posições. Apresentamos um exemplo.

Como já foi dito, o MEDS tem um compromisso explícito com o pressuposto de que a subjetividade é socialmente construída. Isso implica dizer que qualquer generalização possível a partir de resultados quantitativos que usem o MEDS como ponto de partida também teria suas limitações (embora não tão rígidas quanto as do MEDS). Sob pena de incorrerem em sérias inconsistências e erros, não podemos almejar fazer afirmações de cunho *universalista* partindo de um método que tem como pressuposto a *construção social* de características individuais!

Para evitar este e outros possíveis tipos de contradição e inconsistência que poderiam colocar a perder nossos resultados, formas de complementação quantitativa da fase exploratória do MEDS estão sendo cuidadosamente estudadas. Os resultados desses estudos serão tornados públicos tão logo seja possível.

AGRADECIMENTOS

Carla Leitão agradece o apoio financeiro fornecido pelo CNPq por meio do “Programa Sociedade da Informação”.

REFERÊNCIAS

- Berger, P.L. e Luckmann, T. *A Construção Social da Realidade*. Petrópolis, Vozes, 1973.
- Bertaux, D. From the life-history approach to the transformation of sociological practice. In D. Bertaux, (ed.), *Biography and Society*, Beverly Hills, Sage Publications, 1981, p. 29 - 45.
- Chizzotti, A. *Pesquisa em ciências humanas e sociais*. São Paulo, Cortez Editora, 1991.
- da Silva, E.J., de Souza, C.S., Prates, R.O., Nicolaci-da-Costa, A.M. What they want and what they get: a study of light-weight technologies for online communities. Anais do Congresso Latino-Americano de Interação Humano-computador, 2003, *Latin American Conference on Human-Computer Interaction*, (2003), p. 135 - 146.
- de Souza, C.S., Nicolaci-da-Costa, A.M., Prates, R. de O., da Silva, E.J. Compulsory institutionalization: investigating the paradox of computer-supported informal social processes. *Interacting with Computers*, v.16, n.4, 2003, p. prelo.
- Denzin, N. K. and Lincoln, Y.S. *Handbook of Qualitative Research*. Thousand Oaks, Sage Publications, 2000, 2nd edition.
- Foucault, M. *Vigiar e Punir*. Petrópolis, Vozes, 1996. Jodelet, D. Aperçus sur les méthodologies qualitatives. In Moscovici, S. et Buschini, F. (org), *Les Méthodes des Sciences Humaines*, Paris, Presses Universitaire de France, 2003, p. 139 - 162.
- Labov, W. The logic of non-standard English. In W. Labov, *Language in the Inner City*, Philadelphia, University of Pennsylvania Press, 1972, p. 201 - 240.
- Leitão, C.F. e Nicolaci-da-Costa, A.M. Psicologia clínica e informática: por que essa inusitada aproximação?. *Psicologia Clínica Pós-Graduação e Pesquisa (PUC/RJ)*, 2001, p. 189 - 205.
- Leitão, C.F. e Romão-Dias, D. Pesquisas em IHC: um debate interdisciplinar sobre a ética. *Latin American Conference on Human-Computer Interaction*, 2003, Dimap-UFRN, (2003). [<http://www.dimap.ufrn.br/~jair/wiic/wiic.html>, visitado em maio de 2004]
- Nicolaci-da-Costa, A.M. Quem disse que é proibido ter prazer online? Identificando o positivo no quadro de mudanças atual. *Psicologia Ciência e Profissão*, v.22, n.2, 2002, p.12 - 21.
- Nicolaci-da-Costa, A.M. A tecnologia da intimidade. III Workshop de Fatores Humanos em Sistemas Computacionais, 2000, *Anais do III Workshop de Fatores Humanos em Sistemas Computacionais*, SBC, (2000), p.3 - 11.
- Nicolaci-da-Costa, A.M. *Na Malha da Rede: os impactos íntimos da Internet*. Rio de Janeiro, Editora Campus, 1998.
- Nicolaci-da-Costa A. M. A análise de discurso em questão. *Psicologia: Teoria e Pesquisa*, v.10, n.2, 1994, p. 317 - 331.
- Nicolaci-da-Costa, A.M. Questões metodológicas sobre a análise de discurso. *Psicologia Reflexão e Crítica*, v.4, n.1/2, 1989, p. 103 - 108. Nicolaci-da-Costa, A.M.; Leitão, C.F.; Romão-Dias, D. Gerando conhecimento sobre os homens, mulheres e crianças que usam computadores algumas contribuições da psicologia clínica. IV Workshop de Fatores Humanos em Sistemas Computacionais, 2001, *Anais do IV Workshop de Fatores Humanos em Sistemas Computacionais*, SBC, (2001), p.120 - 131.
- Prange, A.P.L. *Da literatura aos Blogs: um passeio pelo território da escrita de si*. Dissertação (Mestrado em Psicologia Clínica), PUC-Rio, 2003.
- Seidman, I. *Interviewing as Qualitative Research: a guide for researchers in education and social sciences*. New York, Teachers College Press, 1998.
- Taylor, S. Locating and conducting discourse analytic research. In M. Wetherell, S. Taylor and S.J.Yates (ed), *Discourse as Data: a guide for analysis*, London, Sages Publications, 2001, p. 5-48.
- Thiollent, M. *Metodologia da Pesquisa-Ação*. São Paulo, Cortez Editora, 1996.
- Turato, E. R. *Tratado da Metodologia de Pesquisa Clínico-qualitativa*. Petrópolis, Vozes, 2003.
- Weiss, R.S. *Learning from Strangers: the art and method of qualitative interview studies*. New York, The Free Press, 1995.
- Wetherell, M.; Taylor, S.; Yates, S.J. (ed) *Discourse as Data: a guide for analysis*. London, Sage Publications, 2001a.
- Wetherell, M.; Taylor, S.; Yates, S.J. (ed) *Discourse Theory and Practice: a reader*. London, Sage Publications, 2001b

Contribuições de Visualização de Informação para a Área de Educação a Distância

Celmar Guimarães da Silva

Heloísa Vieira da Rocha

Instituto de Computação – Universidade Estadual de Campinas
Campinas – São Paulo – Brasil
{celmar, heloisa}@ic.unicamp.br

+55 19 3788-5866

RESUMO

Ambientes de Educação a Distância armazenam uma grande quantidade de dados sobre seus participantes, como quais ferramentas do sistema foram usadas e como ocorreu a interação entre participantes nessas ferramentas. Essas informações, embora de interesse dos participantes, nem sempre estão disponíveis a eles; e quando disponíveis, nem sempre são apresentadas de maneira adequada às análises que esses participantes desejam efetuar. Técnicas de Visualização de Informação podem ser utilizadas para prover essa adequação, utilizando representações visuais e interativas que auxiliem o usuário a compreender esses dados e a extrair deles informações relevantes. Este artigo apresenta como o ambiente TelEduc usa essas técnicas para representar a interação entre participantes na ferramenta InterMap, e algumas formas em estudo para aprimorar e ampliar as funcionalidades dessa ferramenta. Além disso, o artigo apresenta a ferramenta Acessos e discute como técnicas de Visualização de Informação podem ser aplicadas nessa ferramenta para adequá-la às necessidades de usuários de ambientes de Educação a Distância.

ABSTRACT

Distance learning environments accumulate a big quantity of data about their courses' participants, e.g. which system tools were utilized and how the interaction among these participants happened on these tools. Although the participants are interested on this data, it is not always available to them; and when it is, its presentation is not always suitable to the analysis they want to do. Information Visualization techniques can provide this suitability, providing interactive visual presentations which help users to understand data and extract relevant information from it. This paper presents how the TelEduc environment uses these techniques to represent interaction among participants on the InterMap tool, and some ongoing ways to improve this tool. Also, the paper presents the Access tool and discusses how Information Visualization techniques can be applied on this tool in order to fulfill the distance learning users' needs.

Palavras-chaves

Ambientes de Educação a Distância, Visualização de Informação, TelEduc.

INTRODUÇÃO

Ambientes de Educação a Distância (EaD) são sistemas que reúnem diversos recursos comuns da Web – como ferramentas de correio eletrônico, fóruns, salas de bate-papo, porta-arquivos, entre outros – sob um gerenciamento voltado para propiciar o ensino e a aprendizagem a distância. Um exemplo de ambiente de EaD é o TelEduc [7], criado pelo Núcleo de Informática Aplicada à Educação e pelo Instituto de Computação da Universidade Estadual de Campinas.

Possuindo um conjunto amplo de ferramentas, o TelEduc armazena grande quantidade de dados relacionados a cursos, professores (também chamados de formadores), alunos e suas respectivas ações no ambiente. Contudo, muitos desses dados não eram (ou não são) mostrados de acordo com as necessidades dos participantes e com as tarefas que eles precisam efetuar. No escopo do Projeto TelEduc, a primeira dificuldade percebida nesse sentido foi mostrar a presença dos participantes e as interações entre eles; isto deu origem a duas ferramentas do TelEduc – Acessos e InterMap – a serem exploradas em detalhe neste artigo.

A ferramenta InterMap representa graficamente interações entre participantes de cursos do ambiente, usando para isso teorias e técnicas da área de Visualização de Informação. Essa área de Interfaces Humano-Computador utiliza o computador como meio para representar dados abstratos de maneira visual e interativa, com o objetivo de ampliar o uso e a aquisição de conhecimento [2]. As técnicas e conceitos de Visualização de Informação procuram otimizar o uso das habilidades visuais do ser humano, tentando revelar a ele visões breves e esclarecedoras sobre um fenômeno abstrato para o qual não existe uma representação visual espacial inerente [4] – e este é exatamente o caso da maioria dos dados de um ambiente de EaD.

O InterMap representou a primeira tentativa de se aplicar técnicas de Visualização de Informação no TelEduc. Desde sua integração ao ambiente em 2000, seu uso vem apontando a necessidade cada vez maior de reforçar ainda mais as técnicas já aplicadas, em especial no que diz respeito à interação do usuário com as representações visuais apresentadas. Observações dessa interação deram origem a uma nova estrutura para representação de grafos de interação – a estrutura de Grafo Polar – mostrada neste artigo.

Além disso, estudos sobre a possibilidade de uso dessas técnicas em todo o ambiente TelEduc vêm sendo efetuados, iniciando pela ferramenta Acessos, que registra o acesso dos usuários a todas as ferramentas do ambiente.

VI Simpósio sobre Fatores Humanos em Sistemas Computacionais — Mediando e Transformando o Cotidiano. Hotel Bourbon, Curitiba, 17 a 20 de outubro de 2004. UFPR, CEIHC—SBC.

Esses estudos tomaram Acessos como ponto de partida por diversos motivos, dentre os quais se destacam: (1) a generalidade dessa ferramenta, que pode facilmente fazer parte de qualquer ambiente de EaD; (2) o importante papel que essa ferramenta tem no acompanhamento dos alunos dentro do ambiente, tornando-os “visíveis” aos demais participantes; (3) a interligação natural dessa ferramenta com todas as demais do ambiente; e (4) o grande esforço cognitivo que os usuários atualmente apresentam para entender os dados apresentados por essa ferramenta, especificamente no ambiente TelEduc. Assim, a representação gráfica dos dados de acessos tem por objetivo auxiliar os participantes de um curso a entenderem como os demais participantes usam as ferramentas do ambiente, o que pode ser útil tanto para professores quanto para alunos desse curso.

Este artigo relata algumas melhorias que vêm sendo estudadas e implementadas nas ferramentas InterMap e Acessos, relacionadas ao uso de técnicas de Visualização de Informação. O artigo está organizado da seguinte forma: a próxima seção introduz alguns conceitos de Visualização de Informação necessários para o entendimento das próximas seções. Em seguida, são apresentadas a ferramenta InterMap e suas diversas representações visuais. Na seção seguinte, é mostrada a estrutura de grafos chamada Grafo Polar, desenvolvida para melhorar a representação de grafos de interação no InterMap. As duas últimas seções se relacionam à ferramenta Acessos, apontando suas características e problemas e apresentando novas abordagens para solucioná-los.

CONCEITOS DE VISUALIZAÇÃO DE INFORMAÇÃO

Antes de apresentar as melhorias sendo estudadas e implementadas nas ferramentas InterMap e Acessos, é preciso fazer uma breve introdução aos conceitos de Visualização de Informação que vêm conduzindo esse processo de (re)design dessas ferramentas.

De maneira geral, as técnicas de Visualização de Informação têm como objetivo transformar um conjunto de dados brutos, nos quais se procura informação, em uma forma visual compreensível e manipulável pelo usuário. As etapas dessa transformação podem ser acompanhadas pelo modelo de referência proposto por [2] (Figura 1), que permite comparar e contrastar técnicas de Visualização de Informação.

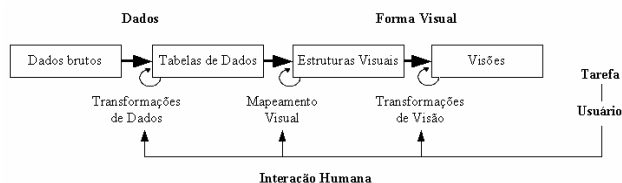


Figura 1 - Modelo de referência para visualização.
Adaptado de [2] (p. 17, figura 1.23).

Nesse modelo, os dados brutos passam por uma transformação que organiza suas inter-relações, gerando assim uma tabela de dados (similar à organização de uma tabela em um banco de dados). Essa tabela é organizada em *variáveis* (dimensões), que caracterizam o conjunto de dados, e em *casos*, conjuntos (tuplas) com um valor para cada variável da tabela. Cada variável pode ser

classificada em numérica/quantitativa, ordinal, ou nominal/categórica. [2, 13]. Em uma comparação com funções matemáticas, cada variável também pode ser classificada como variável de *entrada* (se seu valor determina o valor de outras variáveis) ou de *saída* (se seu valor é determinado por outras variáveis). Em outras etapas do modelo, essas classificações auxiliam a definir qual tipo de representação utilizar para representar visualmente os dados.

Em algumas situações os dados organizados em uma tabela precisam ser reorganizados, dando origem a outra tabela de dados. Uma situação típica em que isso acontece é a de *agrupamento*, na qual dados de várias tuplas são reunidos em uma única e nova tupla; por exemplo, quando se quer reunir um conjunto de pessoas de uma tabela em um grupo.

Tendo organizado os dados que se deseja representar, é necessário apresentá-los na tela sob algum formato. Para isso, pode-se fazer um *mapeamento visual* desses dados no espaço, representando-os por meio de marcas e propriedades gráficas diversas associadas a elas (cores, formas, texturas, orientação, etc.). É criada, assim, uma *estrutura visual* que os representa. Para o uso do *espaço*, podem ser definidos eixos que representem algumas das variáveis envolvidas. Esses eixos podem ser nominais, ordinais, quantitativos ou não estruturados, de acordo com os tipos das variáveis por eles representadas. *Marcas* (como retângulos, círculos, linhas, etc.) podem representar informação pela sua presença ou ausência, e por meio de suas *propriedades gráficas*. Por exemplo, o tamanho de um círculo pode representar a intensidade de uma característica dos dados originais. Uma tabela apresentando a efetividade de cada tipo de marca para representar cada tipo de variável [2] auxilia a determinar com qual marca se deve representar qual variável. Para se representar estruturas mais complexas como árvores (hierarquias) e grafos, pode-se usar linhas para conectar marcas ou encapsular marcas umas dentro das outras.

Gráficos de dispersão (conhecidos como “*scatterplots*” ou “*starfields*”) são exemplos típicos de estruturas visuais. Um gráfico de dispersão baseado em dois eixos ortogonais X e Y pode facilmente representar três variáveis: cada eixo fica responsável por uma variável, enquanto a cor das marcas representa a terceira variável.

Tendo sido definida uma estrutura visual estática, pode-se criar possibilidades para o usuário interagir com ela e modificá-la. Com isso, o usuário pode extrair da visualização mais informação do que seria possível se ela se apresentasse sempre da mesma forma. Estabelecendo parâmetros gráficos, o usuário pode efetuar *transformações de visão*, criando assim *visões* de estruturas visuais de acordo com sua necessidade. Três tipos comuns de transformações de visão são: (1) analisar a estrutura visual por outros *pontos de vista*, usando aproximação, afastamento, deslocamento, seleção de uma parte a ser examinada, ou apresentando detalhes de uma região selecionada (*detalhes-sob-demanda*) [12]; (2) efetuar *distorções*, que permitem apresentar foco e contexto simultaneamente na mesma estrutura; e (3) efetuar *investigações de local*, como solicitar detalhes a respeito de um dado específico de uma estrutura.

Como ponto crucial do modelo, o usuário é quem efetua todas as transformações nele citadas, de acordo com a tarefa que estiver executando. Desse modo, é interessante tornar efetiva essa *interação humana*, provendo rapidez ao efetuar o feedback relacionado a uma ação do usuário. Preferencialmente, é desejado que o efeito de uma ação do usuário apareça na tela em menos de 0,1s, causando uma sensação de causa e efeito. Esse tipo de interação, chamada de interação reativa ou “*responsive interaction*” [13], modifica fundamentalmente o processo de compreensão dos dados, permitindo que (1) o usuário explore mais possibilidades em um determinado período de tempo; e (2) que ele desloque parte do esforço para a máquina, observando o que acontece à medida que os controles da estrutura visual são modificados [2]. Ao ser aplicado à relação entre uma representação visual e os controles que a modificam, essa técnica é também denominada acoplamento forte ou “*tight coupling*” [1].

Outro exemplo importante de técnica de interação é a técnica de *manipulação direta* [1], baseada em quatro princípios: (1) representar visualmente o mundo de ação, incluindo tanto objetos quanto ações; (2) possibilitar ações rápidas, incrementais e reversíveis; (3) possibilitar seleção por apontamento (não por digitação); e (4) efetuar exibição imediata e contínua dos resultados.

Reunindo todos esses conceitos, é possível gerar verdadeiras ferramentas de análise de dados em diversas áreas de conhecimento. Nas próximas seções, serão mostradas as iniciativas de aplicação desses conceitos em ambientes de EaD, mais especificamente nas ferramentas InterMap e Acessos do ambiente TelEduc.

INTERMAP

Antes da criação da ferramenta InterMap, o TelEduc não possuía mecanismos de os professores saberem se os participantes estavam interagindo entre si, exceto no caso de mensagens serem enviadas para todos os participantes do curso. Da mesma forma que na maioria dos ambientes de EaD, cada participante não “enxergava” outras pessoas compartilhando o mesmo espaço e se comunicando umas com as outras, mas via apenas um conjunto de ferramentas que apresentava e organizava a informação disponível nos cursos, geralmente representada de forma seqüencial e textual [8, 11]. Logo, não se conseguia distinguir o aluno “calado e presente” do aluno realmente “ausente”, e essa diferenciação é extremamente importante no acompanhamento de um curso [8].

Considerando essa necessidade de tornar visíveis as interações entre os participantes de cursos, Romani desenvolveu a ferramenta InterMap (*Interaction Map* – Mapa de Interação) [10], que representa graficamente as interações ocorridas em algumas das ferramentas de comunicação do ambiente (Correio, Bate-Papo e Fóruns de Discussão).

A ferramenta é dividida em duas partes principais: as telas de formulação da consulta e a tela de apresentação da informação. Nas telas de consulta, como a mostrada na Figura 2, o usuário escolhe características relacionadas aos dados que deseja analisar, dentre as seguintes: ferramenta em que deseja fazer a análise (Correio, Fóruns de Discussão ou Bate-Papo); tipo principal de representação (Mapa de Interação, Mensagens por Período, Mensagens por Participante ou Fluxo de

Conversação); subtipo de representação (Tabela textual ou Gráfico); período a ser consultado; agrupamento de dados a ser utilizado (por dia, por semana ou por mês); participantes a serem considerados (todos, somente alunos, somente formadores, ou algum participante específico); sessão de bate-papo ou fórum a ser mostrado, se for o caso.

Após o usuário informar dados relativos a essas características, o sistema mostra em uma nova tela uma representação visual correspondente a eles. Essa representação pode ser basicamente dos seguintes tipos: grafo, gráfico de barras, gráfico de dispersão, tabela textual e linhas de tempo paralelas.

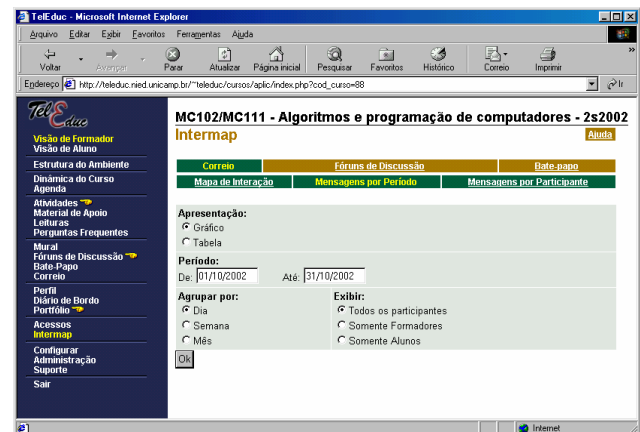


Figura 2 – Exemplo de tela de consulta da ferramenta InterMap.

Na Figura 3 são mostrados exemplos de algumas dessas representações. A Figura 3a mostra um mapa de interação, representando graficamente a troca de mensagens entre os participantes de um curso via ferramenta Correio; os participantes são mostrados como vértices de um grafo, e há aresta entre dois vértices se um dos participantes enviou mensagem para o outro. Nesta representação, existem também alguns controles para efetuarem simplificações e filtros. A Figura 3b exibe um gráfico de barras que representa a quantidade de mensagens enviadas por participante por dia a um fórum específico da ferramenta Fóruns de Discussão. A Figura 3c apresenta em formato tabular e textual os mesmos dados da Figura 3b. Na Figura 3d, é visto um exemplo de gráfico de dispersão utilizado para representar a quantidade diária de mensagens enviadas por participante ao Correio; essa representação relaciona cores a quantidades de acessos, e com isso três variáveis podem ser exibidas simultaneamente: dias, participantes e quantidade de acessos. Na Figura 3e, o mapeamento em cores é novamente utilizado, dessa vez para diferenciar formadores e alunos num diagrama de intervenção do professor nas mensagens de um fórum. Por fim, a Figura 3f apresenta linhas de tempo paralelas para indicar os períodos em que cada pessoa estava presente em uma sessão de bate-papo, marcando com traços horizontais os momentos em que cada uma delas falou.

Tendo sido a primeira ferramenta do TelEduc a contar com o apoio de teorias e técnicas de Visualização de Informação, o InterMap atende diretamente a preocupação dos usuários em tornar visíveis no ambiente as pessoas e as suas interações, que poderiam ser mais

facilmente percebidas em um ambiente presencial de ensino. No entanto, o InterMap peca pela falta de recursos que possibilitem ao usuário explorar melhor as representações apresentadas, pois a maioria delas não lhe permite reorganizar os elementos apresentados na tela ou modificar rapidamente características da representação.

Desse modo, é preciso prover formas melhores de interação do usuário com a ferramenta, preferencialmente possibilitando uma melhor análise dos dados por ela apresentados. Algumas melhorias vêm sendo propostas e implementadas para a ferramenta, como será exposto na próxima seção.

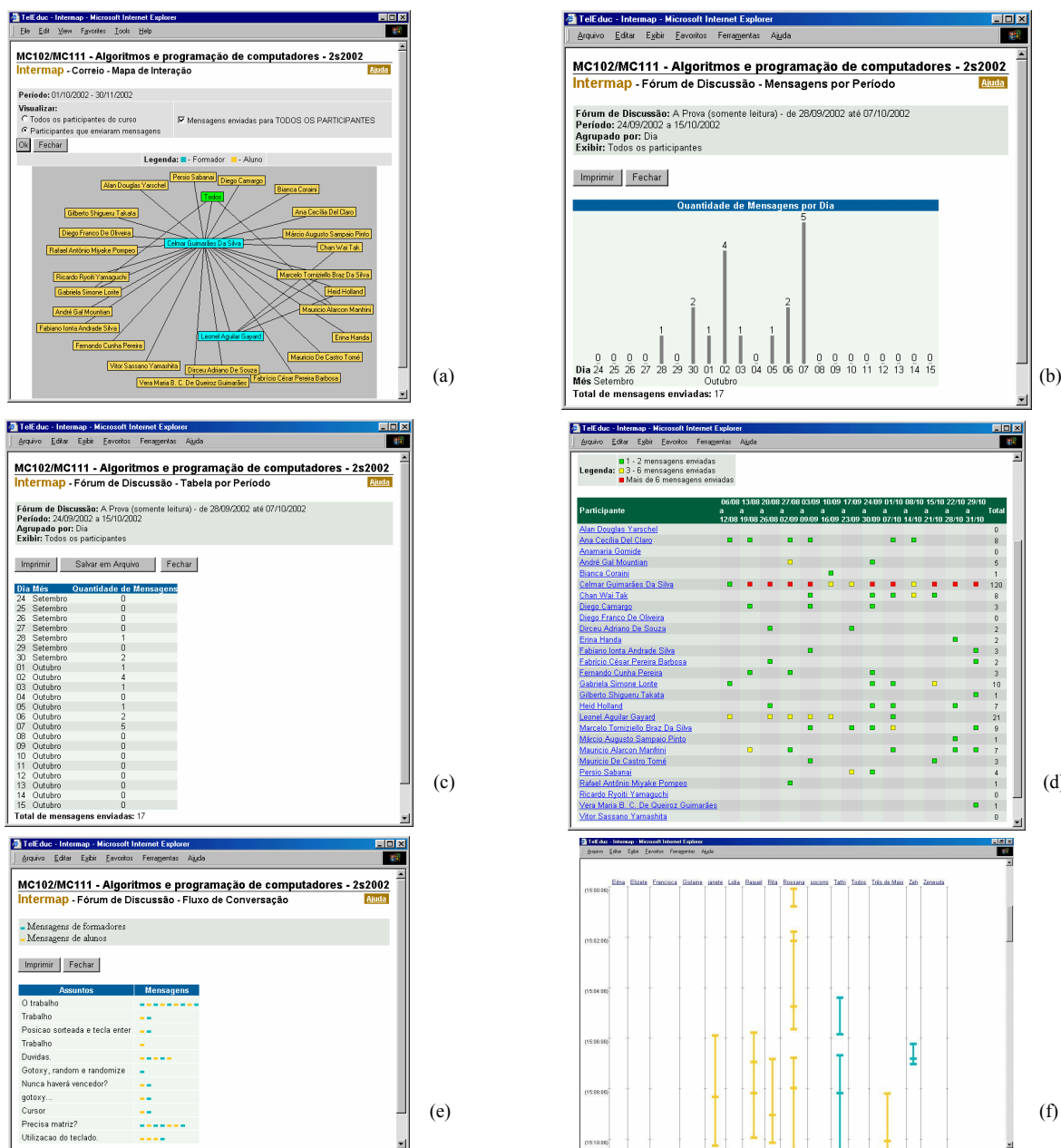


Figura 3 – Exemplos de representações da ferramenta InterMap: (a) mapa de interação da ferramenta Correio; (b) gráfico de barras de um fórum da ferramenta Fóruns de Discussão; (c) tabela numérica com os mesmos dados da figura anterior; (d) gráfico de dispersão das mensagens do Correio; (e) intervenção do professor em um fórum nos Fóruns de Discussão; (f) fluxo de conversação de uma sessão de bate-papo.

GRAFO POLAR

Como mostrado na Figura 3a, uma das formas adotadas pelo InterMap para representar a interação entre participantes é por meio de grafos. A disposição desses elementos é feita por um método orientado a força (“force-directed method”) que iterativamente aproxima na tela os participantes que trocaram mais mensagens entre si. No entanto, observou-se que esse método pode facilmente tornar confuso o grafo de interação, levando o

usuário a intervir na disposição de alguns vértices para eliminar a oclusão de outros elementos do grafo. Vários usuários do InterMap, não satisfeitos com a disposição de vértices oferecida pelo método, interrompem a aproximação automática desses vértices para então posicioná-los manualmente. Também foi possível perceber que nessa redistribuição manual os usuários dispunham os vértices de uma maneira elíptica, trazendo

em um segundo instante para o centro da tela aqueles que desejavam analisar com mais detalhes.

Pela observação desses comportamentos, foi proposta uma estrutura circular para a disposição dos vértices do grafo. Essa estrutura, nomeada de Grafo Polar [14], permite que o usuário disponha os participantes em dois anéis concêntricos (Figura 4), dos quais o mais interno (central) representa seu foco de interesse. Os vértices podem ser reposicionados livremente nos anéis (via manipulação direta), que podem ser redimensionados e rotacionados pelo usuário por controles específicos. Participantes (vértices do grafo) podem ainda ser destacados ou removidos da representação, caso necessário.

Para facilitar a análise das interações representadas na região central do grafo, foi desenvolvida uma técnica de distorção para diminuir o cruzamento de arestas nessa região. Essa técnica cria no grafo uma região denominada área proibida, pela qual somente podem passar arestas relacionadas aos nós do anel central; as demais arestas são distorcidas, tornando-se tangentes ao extremo dessa área, como pode ser visto na Figura 5.

A integração da estrutura do Grafo Polar com o InterMap (Figura 4) abriu precedentes para que a tela que apresenta o mapa de interação passasse por melhorias relacionadas à interação do usuário. Para evitar que o applet responsável pelo Grafo Polar fosse recarregado a cada novo parâmetro ajustado na seção de controles, foi criado um acoplamento forte entre controles e visualização, de tal modo que qualquer modificação nos filtros afeta imediatamente o grafo.

Como visto, o Grafo Polar utiliza técnicas de manipulação direta e acoplamento rígido para permitir uma exploração mais efetiva dos dados apresentados pelo mapa de interação do InterMap. As demais representações dessa ferramenta também podem se beneficiar dessas técnicas, possibilitando ao usuário filtrar e reorganizar dados na tela de acordo com suas necessidades.

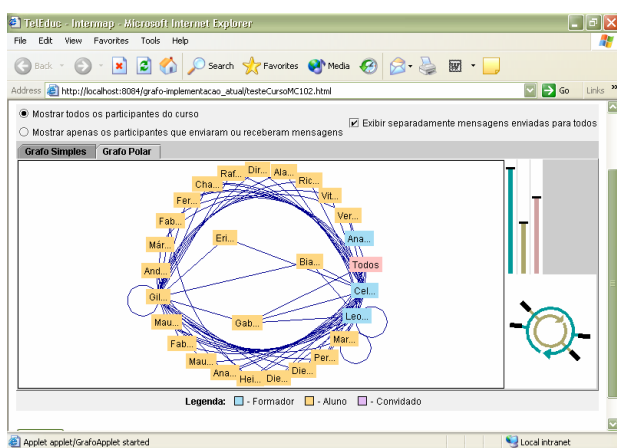


Figura 4 – Estrutura de Grafo Polar representando a troca de mensagens entre participantes de um curso. À esquerda, o grafo em si, apresentando dois anéis concêntricos (o interno com 3 vértices e o externo com 25).

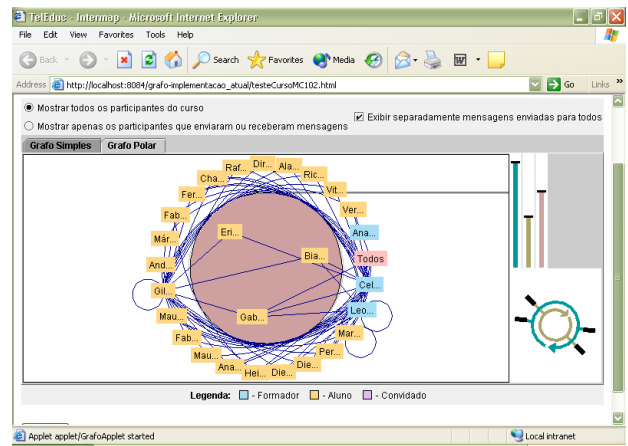


Figura 5 – Grafo Polar mostrando sua área proibida e a distorção de arestas.

ACESSOS

Uma preocupação semelhante à que deu origem ao InterMap era a necessidade de registrar e tornar visíveis os acessos dos participantes ao TelEduc e a suas ferramentas. Isso porque somente se tinha noção se um participante tinha ou não “comparecido” ao ambiente se ele se manifestasse enviando mensagens por alguma ferramenta de comunicação. Essa aparente solidão era incômoda tanto para os alunos, que se sentiam desestimulados a continuar o curso “sozinhos”, quanto para os professores, que não sabiam se seus alunos estavam ou não acessando o ambiente. Novamente, não se podia diferenciar participantes “calados e presentes” daqueles realmente “ausentes”.

Como forma inicial de solucionar esse problema, a ferramenta Acessos foi desenvolvida para registrar o acesso ao ambiente, que acontecia pela ferramenta Agenda. Esse acesso era computado por participante, gerando um gráfico de barras com a quantidade de acessos de cada pessoa desde o início do curso. Junto aos nomes dos participantes, podiam ser exibidos dados como cidade, estado e local de trabalho, de acordo com os quais os participantes podiam ser agrupados. Esse formato de relatório, mostrado em [10], é chamado atualmente de Relatório de Acessos, e pode ser visto na Figura 6. A partir dele, os formadores podiam ter noção da presença dos alunos dentro do ambiente, embora não soubessem quais as ferramentas que eles haviam utilizado.

Com a evolução da ferramenta, novos tipos de relatórios foram incluídos no ambiente, os quais atualmente compõem o Relatório de Frequência (Figura 7). Nesse relatório, o usuário seleciona em uma tela de consulta a ferramenta da qual deseja obter o relatório e também o período de dias a ser exibido (Figura 7a). Em vez de uma ferramenta, pode também ser escolhida a opção “entrada no ambiente”, relacionada aos acessos dos usuários ao ambiente e não a uma ferramenta específica. Após o preenchimento dos dados pelo usuário, o sistema exhibe numa segunda tela o relatório, contendo uma tabela Participante X Tempo cujas células indicam a quantidade de acessos de um determinado participante em um dia específico (Figura 7b). Ao clicar em uma célula não vazia, o usuário pode ver em uma terceira janela (Relatório de Acessos às Ferramentas) os horários dos

acessos por ela indicados (Figura 7c). Essa última janela é uma aplicação direta da técnica de detalhes-sob-demanda apresentada anteriormente.

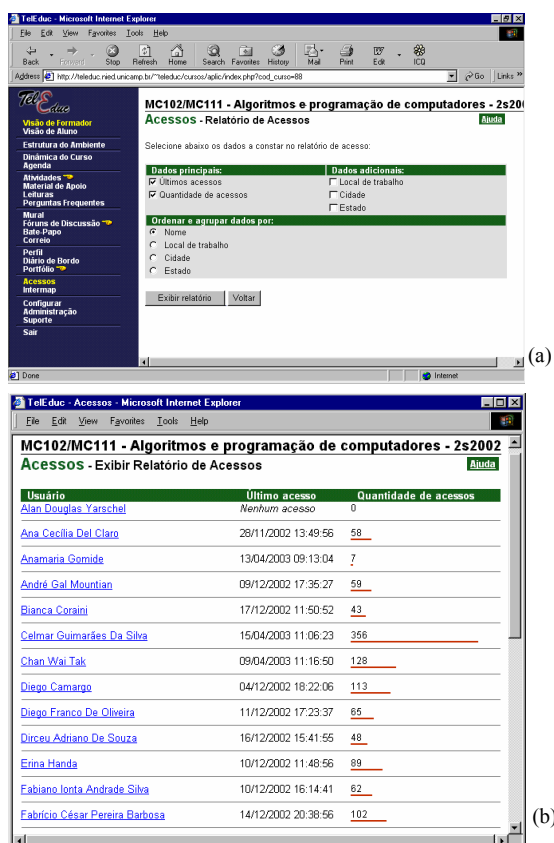


Figura 6 – Relatório de Acessos: (a) tela de consulta; (b) relatório contendo o gráfico de barras resultante.

Com essas variedades de relatórios, a ferramenta Acessos consegue exibir dados relativos aos acessos de todos os participantes ao ambiente, permitindo assim que os formadores efetuem um melhor acompanhamento do uso das ferramentas pelos alunos. É importante ressaltar que alguns usuários, no entanto, utilizam a ferramenta Acessos para efetuar tarefas diferentes das definidas pelos seus desenvolvedores, fato denominado (re)significação de uma ferramenta [9]. Como exemplo de caso real, um participante utilizou Acessos para verificar a co-presença de seus colegas de curso, um uso inusitado dessa ferramenta.

Necessidades e principais problemas

Ao se observar ao longo do tempo como a ferramenta Acessos é utilizada, pôde-se coletar uma série de necessidades relacionadas a esses usos. Uma delas é que a ferramenta deve permitir verificar rapidamente e sem muito esforço cognitivo quando aconteceram os últimos acessos dos participantes ao TelEduc e às suas ferramentas. Sabendo a última vez em que cada participante acessou o TelEduc, formadores podem “cobrar” deles comparecimento ao ambiente e, assim, manter o ritmo do curso. Além disso, possíveis problemas de cada participante podem ser analisados pela última vez em que eles acessaram uma determinada ferramenta. Por exemplo, ao perceber que um aluno resolveu mal uma atividade nesta semana, pode-se analisar seus acessos e

descobrir que seu último acesso a ferramentas que contém material de consulta aconteceu há três semanas.

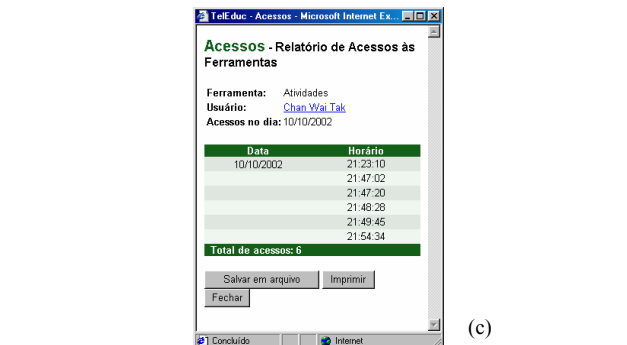
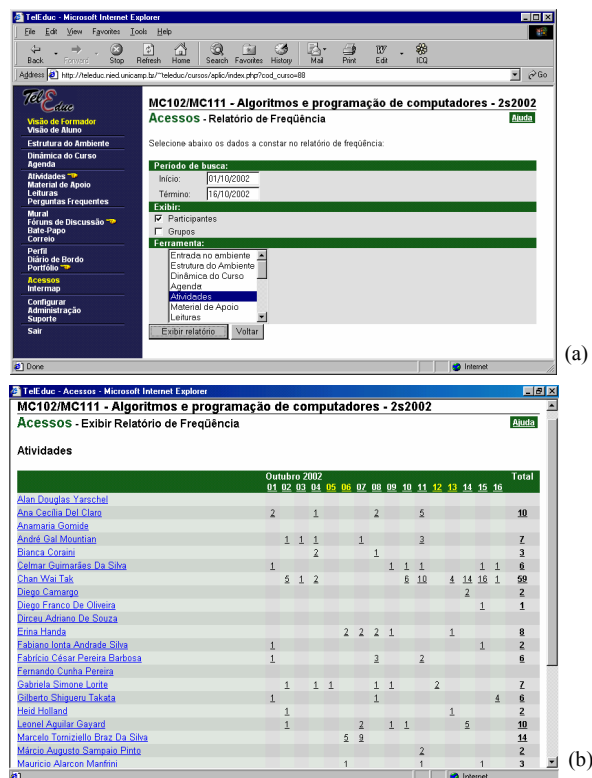


Figura 7 – Telas do Relatório de Frequência da ferramenta Acessos: (a) tela de consulta; (b) relatório da ferramenta Atividades; (c) janela exibindo detalhes sobre os acessos de um participante em um dia específico.

Para participantes de uma maneira geral, a observação dos últimos acessos de outros participantes pode ser útil para verificar a presença simultânea (co-presença) de outros usuários no ambiente. Também é necessário permitir uma análise organizada por grupos de usuários; afinal, dependendo da metodologia de trabalho em grupo adotada, pode haver usuários cujo papel no grupo os incentiva a acessar em maior ou menor quantidade determinadas ferramentas do ambiente.

O mesmo tipo de análise feito com os últimos acessos do participante pode ser feito com todo o seu histórico de acessos, como analisar possíveis problemas acontecidos anteriormente com o participante, ou agendar atividades síncronas entre um grupo de participantes. Neste último caso, uma análise dos dias e horários que essas pessoas costumam acessar o ambiente pode auxiliar na marcação de uma data e horário comum para todos. Nesse histórico,

é importante prover tanto visões simplificadas dos acessos – como mostrar apenas que usuários estiveram ou não no ambiente em um determinado dia – quanto visões mais detalhadas, mostrando a quantidade de acessos efetuada em cada dia por participante, ou ainda apresentando a ordem e o momento exato dos acessos de um participante específico.

Também é importante que o formador consiga utilizar Acessos para avaliar a metodologia usada no curso. Ao fornecer dados para que a ferramenta responda a perguntas como “Quais as ferramentas utilizadas por um determinado participante no último mês?”, o formador pode perceber a que tipo de recursos do ambiente esse participante recorre em maior ou menor grau, e se isso está de acordo com a metodologia planejada para o curso. Esse tipo de informação sugere uma representação cujo foco principal não seja uma ferramenta, mas sim um participante. Essa representação vai ao encontro da requisição de alguns usuários, que solicitam cada vez mais que seja dado foco ao usuário e não à tecnologia (ferramenta) que ele usa.

É necessário também que a ferramenta auxilie o formador a descobrir qual forma de mediação pedagógica está sendo mais efetiva no decorrer do curso. Para tanto, o sistema deve fornecer dados que respondam a perguntas como “Quais as ferramentas que cada usuário mais utilizou ao longo do tempo?”. Isso aponta para a necessidade de representar simultaneamente todos os usuários e todas as ferramentas.

Por fim, existe a necessidade de que as representações usadas permitam alterar facilmente o conjunto de dados em exibição, de modo que o usuário possa navegar no espaço de informação composto pelos dados de acessos, seja alterando o período de tempo considerado, a ferramenta ou o participante em questão. Essa flexibilidade facilitaria a comparação entre visões distintas desse espaço de informação, levando o usuário a um maior entendimento da situação descrita pela representação.

Embora algumas dessas várias necessidades sejam atendidas pelas estruturas atuais da ferramenta, outras ainda estão longe de serem abordadas por ela. Essa inadequação da ferramenta para algumas tarefas é determinada por uma série de características de interface que dificultam a execução dessas tarefas e, assim, desestimulam sua utilização pelos participantes de cursos do TelEduc. Algumas dessas características são relatadas a seguir.

- *Os controles de navegação e de configuração da representação visual ficam em uma janela de configuração, que é separada da janela de visualização.* Com isso, caso o usuário deseje alterar alguma característica da representação exibida (por exemplo, intervalo de tempo a ser considerado), é necessário que ele feche a janela de visualização, volte à janela de configuração, modifique os controles dessa janela, e então mande gerar novamente a janela de visualização. A carga cognitiva desse processo é elevada, e a reversibilidade de ações, embora possível, é dispendiosa para o usuário. A comparação entre

diferentes visualizações também é prejudicada nesse processo.

- *Os controles de navegação exigem entrada de dados via teclado.* Isso acontece especialmente porque o usuário é obrigado a digitar datas indicando o início e o fim do intervalo de tempo a ser considerado na visualização. Embora inicialmente esses campos já estejam *preenchidos* com um valor padrão (do início do curso até a data atual), o usuário pode necessitar modificá-los, seja para analisar outro período de tempo ou para corrigir dados entrados erroneamente; essas situações obrigam o usuário a clicar em cada campo, apagar as datas anteriores e escolher novos valores, processo que atrapalha a rápida comparação do comportamento dos dados em diferentes intervalos de tempo. Além disso, essa forma de entrada de dados pode levar facilmente a erros como, por exemplo, digitar datas fora do formato esperado pelo sistema, ou deixar um campo obrigatório vazio. O uso de controles que possam ser manipulados diretamente via mouse por meio de cliques e arrastes evita esse tipo de situação, simplificando e agilizando a entrada do usuário na maioria dos casos.
- *Falta de orientação para navegação.* Dependendo do intervalo de tempo selecionado pelo usuário, pode ser mostrado a ele, por exemplo, um Relatório de Frequência no qual não consta nenhum acesso, como é mostrado na Figura 8. Nesta situação, denominada *desert fog* (neblina do deserto) [5], o usuário não sabe em que período deve procurar os acessos, pois a interface não lhe fornece nenhuma pista de onde se encontram esses dados. São necessárias formas de tornar visível esse tipo de orientação ao usuário.
- *Dados representados apenas de maneira numérica.* Datas, horários e quantidades de acesso são representados a todo o momento por números, o que nem sempre é uma característica desejada. Por exemplo, as datas de último acesso representadas numericamente no Relatório de Acessos são de difícil comparação, tornando a representação pouco apropriada para verificar de maneira rápida quais participantes acessaram recentemente o TelEduc ou mesmo quais não o estão acessando. No Relatório de Frequência, por sua vez, é mais fácil comparar a existência ou ausência de acessos dos participantes do que a quantidade de acessos feita por participante; isso porque a carga cognitiva exigida para a comparação de conjuntos extensos de números é bem maior do que a de comparar a presença ou ausência de dígitos.
- *Poucas visões estão disponíveis.* Sabe-se que os dados da ferramenta Acessos são tridimensionais, pois estão relacionados a: (1) quem acessou uma ferramenta, (2) qual ferramenta foi acessada, e (3) quando ocorreu o acesso. Observa-se também que os relatórios disponíveis atualmente possuem como foco uma determinada ferramenta escolhida (ou a entrada no ambiente), mostrando como os demais dados relacionados a ela se distribuem ao longo do tempo. Contudo, usuários do TelEduc relatam a necessidade de outras formas de se mostrar os dados de acessos, seja com foco em um participante específico, seja

apresentando uma visão mais geral do que está acontecendo no ambiente em um período específico de tempo.

Dessa forma, é preciso adotar novas abordagens para melhorar a estrutura da ferramenta, tornando-a mais interativa e propícia à exploração. A próxima seção apresenta uma abordagem que procura contemplar as necessidades apresentadas e ao mesmo tempo solucionar os problemas observados em Acessos.

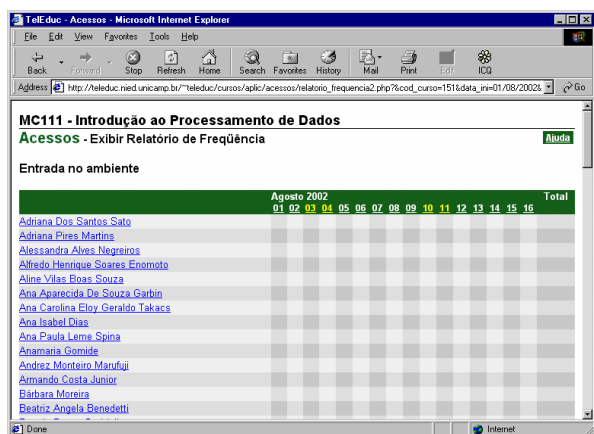


Figura 8 – Desert fog no Relatório de Freqüência: sem dados de acessos, e sem orientação de onde procurá-los.

NOVA ABORDAGEM PARA ACESSOS

Considerando o que foi exposto na seção anterior, deve-se adotar em Acessos uma nova abordagem que atenda melhor às necessidades dos usuários. Serão apresentadas a seguir algumas características já definidas para essa nova abordagem, bem como outras propriedades ainda em estudo.

Introdução

Na ferramenta Acessos, existe uma tabela de dados interna que armazena o momento em que cada usuário acessou cada ferramenta do TelEduc. Para tanto, essa tabela apresenta três colunas: *participantes*, *ferramentas* e *tempo* (com precisão de segundos). Essas três colunas correspondem a três variáveis do modelo de referência apresentado na seção de conceitos de Visualização de Informação no início deste artigo.

Ao analisar as características levantadas na seção anterior, pode-se perceber a necessidade de se representar basicamente dois tipos de informação relacionadas aos acessos dos participantes: a *quantidade de acessos* ao longo de um período, e quando aconteceu o *último acesso* em alguma ferramenta ou no ambiente. Esses dois tipos de informação são os objetivos finais das representações a serem propostas para a ferramenta. Mas, como se pode perceber, eles não são armazenados pela tabela de dados existente. Assim, cada vez que se solicita uma representação, eles precisam ser calculados, gerando uma nova tabela a partir da original (transformação de dados). Portanto, *quantidade de acessos* e *último acesso* podem ser considerados como novas variáveis, geradas por agrupamento de tuplas da tabela de Acessos.

Temos, assim, cinco variáveis (dimensões) que precisam ser representadas. No entanto, a tela apresenta apenas duas dimensões para representá-las, o que se torna um problema. Um dos desafios de Visualização de

Informação está justamente em lidar tanto com essa sobrecarga de variáveis quanto com a sobrecarga de casos armazenados na tabela de dados.

A idéia proposta neste artigo para lidar com esse problema pode ser resumida nos seguintes princípios: (1) possibilitar o acesso a várias estruturas visuais dos dados de acessos, cada uma relativa a um subconjunto das variáveis apresentadas; (2) romper com a abordagem antiga de preenchimento de formulários, mostrando uma estrutura inicial tão logo o usuário entre na ferramenta Acessos; e (3) permitir que, a partir de uma estrutura visual, o usuário possa navegar para outras estruturas semelhantes.

Sem formulários

Como mostrado na seção que apresentou a ferramenta Acessos, a versão original dessa ferramenta apresentava um problema de separação entre a tela de controles e a tela de visualização de dados. Para superar esse problema, essas telas devem ser unificadas, mantendo os controles necessários às estruturas visuais que forem exibidas. Com isso, é possível prover entre controles e visualização um acoplamento forte e que seja percebido pelo usuário.

Além disso, para evitar a obrigatoriedade de se entrar dados exclusivamente via teclado, pode-se complementar controles de digitação (como os campos de datas da Figura 7a) com controles de manipulação direta a eles relacionados.

Com essa unificação das telas, o usuário de início já estará vendo a tela contendo a representação visual. Logo, algum conjunto de dados deve ser mostrado inicialmente, tão logo o usuário acesse a ferramenta. Mostrar esse conjunto inicial orienta os usuários sobre as possibilidades oferecidas pelo espaço de informação exibido, sendo melhor que uma tela com formulário ou uma tela em branco [1]. Para definir qual estrutura inicial mostrar, é necessário inicialmente saber quais as estruturas visuais com as quais se pretende trabalhar.

Estruturas visuais e variáveis

Para definir o conjunto de possíveis estruturas visuais, foram consideradas algumas características importantes desejadas para esse conjunto. Primeiramente, as representações devem ser simples e, conseqüentemente, o esforço cognitivo para compreendê-las deve ser pequeno. Para tanto, cada estrutura visual deve conter apenas duas ou três variáveis, das cinco apresentadas, já que esse número de dimensões é de fácil representação e compreensão. Isso implica que filtragens e agrupamentos devem ser feitos na tabela de dados até reduzi-la a duas ou três dimensões.

Além disso, esse subconjunto de variáveis deve obrigatoriamente conter ou *quantidade de acessos* ou *último acesso*, já que essas variáveis são os dados desejados pelos usuários. Também se deve observar que somente faz sentido falar sobre últimos acessos se for considerado todo o período do curso, e não um intervalo de tempo específico. Assim, as variáveis *tempo* e *último acesso* são mutuamente exclusivas.

Considerando esse conjunto de regras, a Tabela 1 apresenta quais variáveis irão compor cada estrutura visual de Acessos.

Tabela 1 – Estruturas visuais escolhidas, e suas respectivas variáveis.

Variáveis		Estruturas visuais (uma por coluna)								
Entrada	Participantes	X			X	X		X		X
	Ferramentas		X		X		X		X	X
	Tempo			X		X	X			
Saída	Quantidade de Acessos	X	X	X	X	X	X			
	Último Acesso							X	X	X

Definidas as variáveis que irão compor essas estruturas, é preciso definir quais as estruturas que utilizarão essas variáveis. Para isso, foram definidas as seguintes regras:

- Para estruturas de duas variáveis: se a variável de saída for *quantidade de acessos*, a estrutura deve ser um gráfico de barras; se for *último acesso*, deve ser um gráfico de dispersão.
- Para estruturas de três variáveis: são usados gráficos de dispersão, nos quais as variáveis de entrada definem os eixos, e as cores das marcas no espaço são definidas pela variável de saída.

Com isso, as estruturas visuais foram inicialmente limitadas em dois tipos: gráficos de dispersão e gráficos de barras. Isso não significa que outras estruturas visuais (grafos, por exemplo) não possam ser utilizadas para representar acessos, mas que em um primeiro momento elas parecem complicar mais do que auxiliar os usuários nessa situação.

Estrutura visual inicial

Tendo definido o conjunto de estruturas visuais com o qual se irá trabalhar, é preciso definir qual delas deve ser mostrada inicialmente para o usuário. Considerando o mantra de Shneiderman – “Visão geral primeiro, zoom e filtragem, depois detalhes sob demanda” [12] –, torna-se necessário que a visão inicial mostrada seja o mais geral possível. Para isso, ela deve abranger todos os participantes e ferramentas do ambiente, bem como todo o intervalo do curso. Logo, deve ser escolhida uma estrutura visual contendo as variáveis *participantes* e *ferramentas*. Das duas opções de variável de saída disponíveis, a mais genérica é aquela que agrupa todos os acessos do curso, ou seja, *quantidade de acessos*. Assim, essas são as três variáveis que foram definidas para a estrutura visual inicial, que é um gráfico de dispersão colorido de acordo com a quantidade de acessos de cada participante a cada ferramenta (Figura 9).

Navegando em uma mesma estrutura

Como visto na seção sobre conceitos de Visualização de Informação, é interessante prover formas do usuário alterar seu ponto de vista no espaço de informação sendo exibido. Em termos das estruturas visuais de Acessos, isso significa que o usuário deve ter à sua disposição controles que permitam facilmente alternar entre diferentes intervalos de tempo, ferramentas e participantes.

		Ferramentas			
		Atividades	Portfólio	Correio	Bate-papo
Participantes	Cássia				
	Celmar				
	Lara				
	Pedro				

Figura 9 – Gráfico de dispersão representando a quantidade de acessos de alguns participantes a algumas ferramentas. Cores mais escuras representam maior quantidade de acessos, enquanto branco representa nenhum acesso.

Para tanto, ele precisa ter controles adequados na interface da ferramenta que lhe possibilitem fazer isso. Para trocar um participante ou uma ferramenta, podem ser usadas listas de seleção. Já para a escolha de intervalos de tempo, podem ser utilizados *range-sliders* [3], que são controles específicos para a seleção de intervalos de uma lista seqüencial de elementos.

Navegando entre estruturas visuais

Dado o conjunto de estruturas visuais disponíveis, o usuário deve ter meios de chegar a elas. Um desses meios é a seleção direta da estrutura desejada, o que pode ser feito escolhendo a estrutura em uma ou mais listas de seleção, por exemplo. Contudo, implementar somente essa solução para escolher estruturas privilegiaria apenas estratégias de exploração planejada do espaço de informação, comuns para usuários experientes que sabem exatamente o tipo de informação desejada e onde ela se encontra. Estratégias oportunistas e exploratórias, mais comuns para usuários iniciantes (usuários alvo de ambientes de EaD), seriam desestimuladas, e esses usuários poderiam perder o interesse no uso da ferramenta.

Desse modo, é preciso prover formas práticas do usuário acessar outras estruturas visuais, a partir de elementos da estrutura que estiver sendo mostrada a ele. Analisando como exemplo a estrutura inicial escolhida, mostrada na Figura 9, verificamos que ela apresenta (1) uma lista de nomes de participantes; (2) uma lista de ferramentas; (3) um conjunto de células coloridas (neste artigo, em tons de cinza), no qual cada célula representa a quantidade de acessos de um participante a uma ferramenta. Propõe-se que o clique em cada um desses tipos de elementos conduza a uma estrutura visual diferente, da seguinte forma:

- Ao clicar em um nome na lista de participantes, o usuário estará solicitando detalhes sobre os acessos desse participante; ou seja, as informações desse participante, que estavam agrupadas em uma linha da estrutura, devem ser expandidas. Neste caso, pode-se mostrar a estrutura que contém as variáveis *ferramentas*, *tempo* e *quantidade de acessos*, detalhando temporalmente as quantidades de acessos do participante mostradas na estrutura anterior. Deve-se perceber que essa nova estrutura visual não pode conter participantes, pois uma vez que o participante foi definido e é único, ele não é mais uma variável da estrutura.
- De forma semelhante, ao clicar em um nome na lista de ferramentas, o usuário estará solicitando detalhes sobre os acessos ocorridos nessa ferramenta, que não

deve aparecer na estrutura. Pode-se exibir neste caso a estrutura que contém as variáveis *participantes*, *tempo* e *quantidade de acessos*.

- Ao clicar em uma das células coloridas, o usuário estará solicitando detalhes sobre o dado representado por aquela célula. No entanto, ela se refere tanto a um participante quanto a uma ferramenta. Com isso, ele está definindo tanto um participante quanto uma ferramenta, e assim nem um nem outro poderão fazer parte da nova estrutura visual. Neste caso, deve-se mostrar uma estrutura contendo apenas as variáveis *tempo* e *quantidade de acessos*.

Esse comportamento pode ser acompanhado na Figura 10, pelas setas que saem da estrutura visual inicial, destacada no centro da figura. Nessa abordagem, o clique em uma célula do gráfico de dispersão ou em um nome de um dos seus eixos irá levar o usuário a uma representação diferente, na qual aquele elemento passa a determinar (ou seja, filtrar) os dados das demais variáveis. Isso pode ser estendido a todos os gráficos de dispersão da ferramenta Acessos, criando assim uma rede que interliga várias das estruturas visuais propostas. Na Figura 10, essa interligação é indicada por setas simples não pontilhadas. Setas pontilhadas representam o caminho inverso de navegação, que pode tornar-se disponível por meio de links específicos. Em algumas situações, o usuário pode escolher entre duas estruturas visuais distintas, o que pode ser feito via menu de contexto.

Outra interligação necessária é a que permite navegar de uma estrutura visual que mostre *quantidade de acessos* para outra equivalente que mostre *últimos acessos*, e vice-versa. Setas em negrito representam essa possibilidade de navegação na Figura 10.

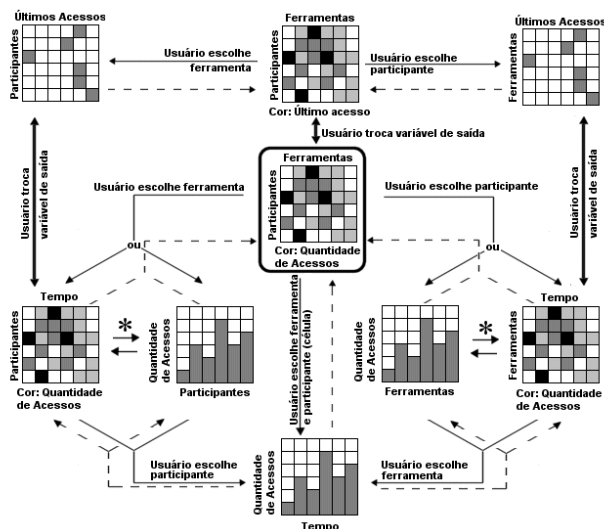


Figura 10 – Diagrama de navegação entre estruturas visuais. Em destaque, no centro, a estrutura visual inicial.

Existem ainda casos especiais, marcados por asteriscos na Figura 10, em que o usuário pode alternar entre estruturas que ofereçam ou não um detalhamento temporal dos dados quantitativos apresentados.

Questões diversas

Existem diversos pontos a serem ainda melhor estudados e que irão influenciar a aparência final da ferramenta e das representações por ela apresentadas. Alguns desses pontos são mostrados a seguir.

- *Agrupamento*: É possível representar os acessos de uma maneira mais genérica ou mais específica, considerando ou não grupos de participantes e de ferramentas. No caso de participantes, é possível representar apenas os grupos de que participam (dado obtido na ferramenta Grupos, no TelEduc). Ferramentas, por sua vez, podem ser agrupadas segundo a divisão estabelecida por [7] (comunicação, coordenação e administração). Esses dois tipos de agrupamento devem ser controlados pelo usuário diretamente na estrutura visual. Outra forma de agrupar acessos é reuni-los em minutos, horas, dias, semanas ou meses. Esse agrupamento temporal pode ser definido tanto automaticamente quanto pelo usuário.
- *Reordenação*: É preciso que o usuário possa estabelecer a ordem entre os elementos da representação, uma vez que diferentes rearranjos desses elementos podem levar a diferentes compreensões dos fatos mostrados na representação [13]. Essa reordenação pode ser provida de duas formas: semi-automática (o usuário solicita a ordenação alfabética dos participantes, por exemplo) ou manual (por manipulação direta, o usuário reordena as linhas e colunas da representação como melhor lhe aprouver).
- *Representação da quantidade de acessos*: Atualmente se representa a quantidade de acessos nas células do Relatório de Freqüência por meio de números. No entanto, isso aumenta a carga cognitiva do usuário, que não consegue efetuar comparações entre regiões com mais ou menos acessos na representação. Utilizando uma escala de cores que agrupe os valores em intervalos, o usuário pode analisá-los visualmente com maior facilidade. Mostrar ou não numericamente a quantidade de acessos pode ser uma opção disponível para o usuário.
- *Escala de cores*: Ao representar estruturas com duas variáveis de entrada e uma de saída, é comum representar o valor da variável de saída por propriedades visuais de fácil percepção pelo usuário. No caso da ferramenta Acessos, as duas variáveis de saída possíveis são *últimos acessos* (variável ordinal temporal) e *quantidade de acessos* (variável quantitativa). Considerando que essas variáveis tivessem seus valores agrupados em poucos intervalos de tempo e de quantidade de acessos, então ambas seriam ordinais. Analisando a tabela de efetividade de propriedades retiniais apresentada em [2], percebe-se que usar cores é uma das maneiras mais adequadas de se representar marcas em variáveis ordinais. Contudo, escolher quais escalas de cores utilizar depende muito dos dados que se deseja mostrar [13]. Para representar quantidade de acessos, o uso de uma “escala de calor” (branco, amarelo, laranja, vermelho) parece apropriada. Por

outro lado, não se encontra na literatura nenhuma escala de cor diretamente associada à representação de tempo; assim, é muito provável que a escala de cores associada a essa dimensão precise ser ensinada aos usuários. Além disso, existem ainda diversos problemas relacionados à influência dos agrupamentos na definição dos valores representados pela escala de cores.

- *Dicas de navegação*: Em estruturas nas quais uma das variáveis de entrada seja *tempo*, é interessante mostrar setas laterais indicando em que direções no tempo existem mais acessos. Esse tipo de informação de sensibilidade [13] é essencial para orientar o usuário em situações de *desert fog* como a mostrada na Figura 8.
- *Tratando um grande número de participantes*: Quando o número de participantes for muito grande em uma tabela contendo *participantes* como variável de entrada, a ponto de esses participantes não caberem na tela, deve-se pensar como proceder. As abordagens mais simples seriam utilizar barra de rolagem ou paginação da representação; no entanto, essas abordagens dificultariam para o usuário a análise de dados como um único conjunto, e ainda seriam um empecilho para a manipulação direta de linhas e colunas. Uma alternativa seria “espremer” verticalmente todos os participantes na mesma coluna, mostrando todos na tela, mesmo que com altura menor que a necessária para escrever seus nomes. Nessa solução, adotada pelo programa Table Lens [6], um elemento selecionado aparece em seu tamanho normal, enquanto os demais aparecem achatados. O principal problema dessa abordagem é a falta de legibilidade dos nomes não selecionados.
- *Seleção direta*: O usuário deve poder selecionar um retângulo na tela para filtrar elementos que devem permanecer na representação, descartando os demais.
- *Uso de range-slider*: Como citado anteriormente, *range-sliders* podem ser utilizados para selecionar intervalos de tempo a serem considerados nas representações. É desejado, no entanto, que o usuário possa analisar (e, logo, escolher) intervalos de tempo que podem envolver tanto meses quanto minutos. Essa possibilidade de análise macroscópica e microscópica esbarra em limitações do *slider* tal como hoje ele é concebido: se ele for feito para representar meses, pode ser que um intervalo de 1 pixel na tela represente várias horas de acessos, tornando impossível acessar intervalos de 1 hora ou menos. Variações de *range-slider* que superem esse tipo de limitação estão sendo estudadas.
- *Estruturas*: A princípio, considerando que a interface deve ser o mais simples possível, o uso de gráficos de dispersão similares a tabelas parece ser o mais adequado no contexto da ferramenta Acessos. Todavia, usar grafos para representar participantes acessando ferramentas é uma possibilidade que deve ser estudada.

CONCLUSÃO

Acessos e InterMap são duas ferramentas cujos respectivos papéis são informar sobre a presença e a interação dos participantes de cursos no TelEduc. Esses dados são necessários tanto para professores quanto alunos, e devem poder ser acessados facilmente dentro do sistema. Dada a quantidade de acessos e de interações que acontecem no ambiente, foi cogitada a possibilidade de utilizar conceitos da área de Visualização de Informação para auxiliar na exibição desses dados, e conseqüentemente no seu entendimento pelos usuários do TelEduc. Os primeiros usos desses conceitos com o InterMap indicaram a existência de grande potencial de obtenção de informação ainda não explorado no TelEduc, e que pode auxiliar formadores e alunos a utilizar o ambiente de maneira mais efetiva.

Este artigo apresentou algumas iniciativas rumo a essa tentativa de uso de Visualização de Informação junto a um ambiente de Educação a Distância. A estrutura do Grafo Polar, o acoplamento forte entre controles e visualização, e as diversas melhorias propostas para a ferramenta Acessos fazem o TelEduc caminhar cada vez mais nessa direção. Trabalhos futuros planejam tornar disponível toda a base de dados do TelEduc para consulta dos seus usuários utilizando técnicas de Visualização de Informação.

Finalmente, o desenrolar desta pesquisa tem demonstrado os desafios e o enorme potencial de aplicação que ambientes de EaD oferecem para o uso de técnicas e teorias advindas da área de Visualização de Informação, como também a necessária apropriação e adequação destas técnicas no sentido de ampliar tanto as funcionalidades como a usabilidade dos atuais ambientes para EaD.

REFERÊNCIAS

1. Ahlberg, C., Shneiderman, B. Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. *Proc. CHI'94, ACM Conference on Human Factors in Computing Systems*, 1994, 313-317. <http://citeseer.nj.nec.com/hlberg94visual.html> (23/Maio/2004).
2. Card, S.K., Mackinlay, J.D. e Shneiderman, B. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufman Publishers, 1999.
3. Carr, D. A., Jog, N., Kumar, H. P., Teittinen, M., e Ahlberg, C. *Using Interaction Object Graphs to Specify and Develop Graphical Widgets*. Technical Report CS-TR-3344, University of Maryland, 1994.
4. Chen, C. Editorial – Information Visualization. *Information Visualization* 1, 1-4, Palgrave Macmillan, 2002.
5. Jul, S. e Furnas, G.W. Critical Zones in Desert Fog: Aids to Multiscale Navigation. *Proc. UIST 1998*.
6. Rao, R. e Card, S.K., The Table Lens: Merging Graphical and Symbolic Representations in an Interactive Focus + Context Visualization for Tabular Information. In Card, S.K., Mackinlay, J.D. e Shneiderman, B. *Readings in Information Visualization: Using Vision to Think*. 343-349. Morgan Kaufman Publishers, 1999.

7. Rocha, H.V. da. O ambiente TelEduc para Educação a Distância Baseada na Web: Princípios, Funcionalidades e Perspectivas de Desenvolvimento. In: Moraes, M.C. (org.), *Educação a Distância: Fundamentos e Práticas* (11), 197-212, Nied/Unicamp, 2002. http://teleduc.nied.unicamp.br/pagina/publicacoes/8_Cap11_heloisa.pdf (23/Maio/2004).
8. Rocha, H.V. da, et al. *Projeto TelEduc: Pesquisa e Desenvolvimento de Tecnologia para Educação a Distância*. IX Congresso Internacional de Educação a Distância da ABED (Associação Brasileira de Educação a Distância), 2002. http://teleduc.nied.unicamp.br/pagina/publicacoes/premio_abed2002.pdf (23/Maio/2004).
9. Rocha, H.V. da, Oeiras, J.Y.Y., Freire, F.M.P., Romani, L.A.S. Design de ambientes para EaD: (re)significações do usuário. *Anais do IV Workshop sobre Fatores Humanos em Sistemas Computacionais*, 2001, 84-95, UFSC e SBC. http://teleduc.nied.unicamp.br/pagina/publicacoes/5_ihc2001.pdf (23/Maio/2004).
10. Romani, L.A.S. *InterMap: Ferramenta para Visualização da Interação em Ambientes de Educação a Distância na Web*, Dissertação de Mestrado, Instituto de Computação, Universidade Estadual de Campinas, 2000. http://teleduc.nied.unicamp.br/pagina/publicacoes/lromani_disser.pdf (23/Maio/2004).
11. Romani, L.A.S., Rocha, H.V. da, Silva, C.G. da. Ambientes para educação a distância baseados na Web: Onde estão as pessoas? *Anais do III Workshop sobre Fatores Humanos em Sistemas Computacionais*, 2000. http://teleduc.nied.unicamp.br/pagina/publicacoes/lromani_ihc.pdf (23/Maio/2004).
12. Shneiderman, B. The Eyes Have It: A Task by Data Type *Taxonomy* for Information Visualizations. *Proc. IEEE Conference on Visual Languages*, 1996, 336-343.
13. Spence, R. *Information Visualization*. Addison-Wesley, 2001.
14. Silva, C.G. da, Rocha, H.V. da. *Grafo Polar: uma abordagem radial para visualização e exploração de grafos*, Relatório Técnico IC-03-17, Instituto de Computação, Universidade Estadual de Campinas, 2003. <http://www.ic.unicamp.br/ic-tr-ftp/2003/03-17.ps.gz>. (23/Maio/2004).

Estudo do Impacto do Design e das Formas de Uso sobre a Recuperação de Informações em Fóruns de Discussão Online

Adéle Malta Pontes¹

Carla Faria Leitão^{1,2}

Clarisse Sieckenius de Souza¹

Simone Junqueira Barbosa¹

Violeta S.T.D.B Quental³

SERG- Semiotic Engineering Research Group¹

Depto de Informática, PUC-Rio

Depto. de Psicologia, PUC-Rio²

Depto. de Letras, PUC-Rio³

{[adele](mailto:adele@inf.puc-rio.br); [cfaria](mailto:cfaria@inf.puc-rio.br); [clarisse](mailto:clarisse@inf.puc-rio.br); [simone](mailto:simone@inf.puc-rio.br)}@inf.puc-rio.br

violetaq@rdc.puc-rio.br

RESUMO

Este trabalho apresenta resultados de um estudo de caso, através do qual verificamos que a recuperação consistente de informações em sistemas do tipo fórum de discussão é fortemente influenciada pelo design da ferramenta (arbitrado pelo designer) e, também, pelas diferentes maneiras como os usuários se apropriam desta ferramenta. Observamos, por exemplo, que decisões de design voltadas estrategicamente para a recuperação das informações podem ser sistematicamente ignoradas por alguns usuários, o que certamente pode impactar negativamente a consistência de recuperação almejada. Utilizamos como base para este estudo conceitos da Engenharia Semiótica, bem como conceitos e métodos provenientes dos estudos lingüísticos do texto e da interação, especialmente da Análise da Conversação.

ABSTRACT

This paper presents the results of a case of study in which we verified that consistent information retrieval on discussion forum systems is strongly influenced by certain design decisions (made by the designer), as well as by the various ways how users use the tools. We verified that design features included specifically in view of information retrieval can be declined by some users, which may certainly influence the desired retrieval consistency in a negative way. The foundation of our study relies on Semiotic Engineering concepts, as well as methods and concepts from Linguistic, like as Conversation Analysis.

Palavras Chaves

Engenharia Semiótica, sistemas de fóruns de discussão, recuperação de informação.

INTRODUÇÃO

A interface de usuário é responsável por fazer com que o usuário consiga realizar suas tarefas de maneira fácil, rápida, inteligente e satisfatória. A Engenharia Semiótica [4] vê esta interface como uma mensagem enviada pelo projetista aos usuários, na qual ele transmite suas intenções e decisões de projeto, motivando os usuários a adotarem formas de interação compatíveis com sua (*i.e.* do projetista) visão. Todo projeto, ou design, determina possibilidades e formas de uso, facilitando, obrigando, dificultando ou impedindo vários tipos de comportamento. O importante da “mensagem do designer”, enviada através dos desdobramentos interativos da interface, é comunicar ao usuário estas formas de determinação. Por isto, é essencial que durante a elaboração do seu trabalho o designer possa refletir intensa e extensamente sobre as implicações humanas e técnicas de suas escolhas.

Uma das aplicações em que este papel do designer é mais complexo é a de sistemas de *groupware*. Sob o aspecto humano, nestes sistemas o designer não só define como será a interação do usuário com o sistema, mas também como será a interação entre os usuários membros de um grupo. Além disto, visando permitir o acesso ao histórico das interações do grupo, alguns recursos tecnológicos necessários para a recuperação de informações apresentam certos requisitos sobre os quais o designer tem de ter a máxima clareza.

Focando especificamente o contexto de fóruns de discussão, presentes em quase todas as aplicações bem-sucedidas de *groupware*, um designer pode ver-se diante da escolha entre duas alternativas opostas: ou arbitra estruturas retóricas rígidas em que os participantes têm de *encaixar* as suas intervenções, ou então relaxa os mecanismos estruturantes, e exige dos participantes que apenas indiquem (se tanto) a que outra intervenção a sua está se referindo. Sob o aspecto da experiência dos usuários, estruturas rígidas podem sacrificar a agilidade da discussão e, em última análise, os propósitos do grupo, fazendo parecer que a opção de usar um mínimo de estruturação é mais promissora. Porém, ao pensarmos que

a *memória* das discussões é um rico acervo do qual é não apenas desejável, mas freqüentemente necessário, recuperar informações, com rapidez e precisão, nossa avaliação pode mudar. São justamente os mecanismos estruturantes, que marcam explicitamente relações e propriedades das intervenções dos participantes, os elementos que permitirão um processamento computacionalmente eficiente de informações e por conseguinte o atingimento dos objetivos dos usuários em inúmeras situações. Na ausência de tais mecanismos, o processamento computacional tem por objeto o texto livre registrado pelos participantes a cada intervenção, o que significa que a recuperação das informações ali contidas é dependente de um sistema capaz de interpretar linguagem natural. Esta multiplicidade de aspectos nas tomadas de decisão de design é dificilmente percebida ou entendida pelos usuários, e cria com facilidade atritos e descontentamentos seja durante o processo de desenvolvimento de software, seja já quando o produto está em pleno uso. A proposta da Engenharia Semiótica é facilitar a conscientização dos designers sobre a importância de comunicar seus porquês *em tempo de interação* [5] e oferecer-lhes ferramentas para elaborar eficientemente este tipo de comunicação.

Neste contexto, nosso trabalho tem o objetivo de apresentar resultados de um estudo sobre o uso de fóruns de discussão. O estudo envolve a análise da natureza das discussões de dois grupos de usuários – Grupo A, interagindo em um ambiente fracamente estruturado, e Grupo B, interagindo em um ambiente fortemente estruturado. Nosso objetivo é ter maior clareza do que pode realmente acontecer, quando um designer opta por maior ou menor estrutura, favorecendo uma recuperação de informação computacionalmente mais eficiente (o que é, indiscutivelmente, um benefício para os usuários) ou uma expressão mais espontânea e leve durante a discussão, respectivamente.

A base teórica deste estudo está ancorada nos conceitos da Engenharia Semiótica, principalmente no pressuposto de que diferenças estáticas e dinâmicas no discurso do designer (utilizadas neste caso para dar o teor da captura e recuperação da informação) influenciam no processo comunicativo dos usuários. Nesta perspectiva os casos aqui analisados nos mostram “**como**” estas diferenças estão influenciando na interação dos usuários, com o sistema e uns com os outros. Para a definição da estrutura da interação dos ambientes de discussão e para a análise dos resultados, utilizamos também conceitos e métodos provenientes dos estudos lingüísticos do texto e da interação, especialmente da Análise da Conversação [10;13]. Nossos resultados mostram que os usuários podem *burlar* estruturas propostas e, com isto, comprometer o próprio propósito do emprego de tais estruturas. Ou seja, na medida em que burlam estas estruturas e apresentam informação de um tipo no lugar ou na vez de outra, todo o argumento da recuperação de informação computacionalmente eficiente que justifica a carga estruturante do design cai por terra.

Nas próximas seções oferecemos um breve resumo dos conceitos de Engenharia Semiótica aplicados a

groupware. Em seguida, fazemos o relato do estudo realizado, descrevendo as ferramentas utilizadas e os métodos adotados para a coleta e análise dos dados, bem como apresentando os achados principais. Por último, discutimos os pontos deste estudo que nos parecem mais relevantes para a pesquisa em IHC, identificando questões importantes para serem investigadas em trabalhos subsequentes.

ENGENHARIA SEMIÓTICA E GROUPWARE

O hoje canônico “projeto centrado no usuário” [11] prega que o processo de desenvolvimento de software de qualidade deve seguir os seguintes princípios: envolver ativamente os usuários e promover a máxima compreensão, pelos desenvolvedores, das atividades e requisitos dos usuários; distribuir adequadamente o “trabalho” entre usuários e tecnologia; iterar várias vezes os ciclos de projeto até convergir para soluções ótimas; e trabalhar com equipes e perspectivas multidisciplinares. Está implícito nesta norma, e em todas as propostas de técnicas que facilitam que os designers a sigam, que as conversas e contatos com os usuários se encerram em fases pré-implantação ou se retomam pós-implantação em processos de avaliação normalmente destinados a informar o projeto de novas versões da tecnologia. Ou seja, a conversa entre designers e usuário *não se passa em tempo de interação* com a tecnologia. A Engenharia Semiótica propõe algo radicalmente diferente – que esta conversa *continua* durante a interação, através do sistema que é, em última análise, o preposto do designer, aquele que fala em nome do designer [6; 5].

O designer fala com o usuário através da interface. Esta fala tem peculiaridades muito interessantes, se comparada à fala face-a-face em contextos sociais. Primeiramente, ela se realiza por meio de códigos de interação que incluem ícones, menus, linhas de comando, estados e reações do sistema. Comandar e controlar um sistema interativo é, nesta perspectiva, indiretamente conversar com o designer através de atos de fala que criam objetos computacionais, estabelecem comunicações com outros usuários, e tudo o mais que as aplicações computacionais podem realizar. O estilo da comunicação também é especial porque o interlocutor, humano e representado por um sistema computacional, normalmente aparece *metaforizado* para o usuário como ferramenta, como agente inteligente, ou qualquer outra das metáforas comumente usadas em IHC. Uma segunda peculiaridade desta fala é que raramente os designers aproveitam conscientemente toda a gama de oportunidades que têm para *falar sobre* os atos de fala que se podem realizar com a tecnologia. Quando o fazem, em geral ficam presos a indicações de natureza operacional no uso da linguagem de interface – que botão apertar para isto, que menus e diálogos ativar para aquilo, e assim por diante. Pouco ou nada dizem sobre as razões e os valores embutidos na tecnologia, sobre as vantagens relativas de uma ou outra estratégia de uso, embora não só seja possível mas mesmo desejável fazê-lo no contexto da ajuda *online* [17]. Uma terceira e última peculiaridade de que vamos falar aqui é o fato de os usuários poderem declinar *affordances*, ou oportunidades de uso da tecnologia. Este

declinar consiste em deliberadamente usar a tecnologia de uma maneira diferente daquela julgada pelo designer como a melhor ou mais apropriada em determinado contexto de interação [7]. Por exemplo, os *browsers* atuais permitem aos usuários configurarem vários itens relativos a segurança e privacidade. Ao serem instalados, eles guiam o usuário através de uma seqüência de telas em que se pergunta ao usuário que valor dar a um conjunto de parâmetros. Se bem que na maioria das vezes esteja marcado um valor *default* para os parâmetros, também há texto explicativo (uma comunicação explícita do designer para o usuário) sobre como diferentes faixas de valores afetam segurança e privacidade. Isto é, o designer está convidando o usuário a tomar uma decisão consciente sobre o tipo e o nível de controle que deseja ter sobre seus dados pessoais em transações na Internet. Porém, o usuário pode declinar este convite à reflexão e simplesmente apertar “enter” a cada diálogo, aceitando inconseqüentemente os valores *default* que lhe são propostos. A pergunta importante é: será que ele sabe o custo da decisão que tomou? A resposta é simples: em geral não tem a menor idéia, e fica surpreso e indignado quando a tecnologia não se comporta como ele acha que devia.

Uma das metas da Engenharia Semiótica é, portanto, permitir aos designers encontrar uma solução de comunicação eficiente e aceitável, que harmonize duas situações extremamente assimétricas em processos de tomada de decisão como o que acabamos de narrar. Por um lado, está o designer tentando alertar o usuário sobre todas as imbricações e conseqüências da parametrização de um *browser* em situações reais de uso (algo sobre o que o designer tem plena consciência, pois foi ele que assim produziu a tecnologia). Por outro, está o usuário ávido por começar logo a usar a ferramenta, com pressa de terminar um diálogo que no momento pouco lhe interessa, e aproveitando o caminho mais curto – o de “teclar *enter* até acabar”. A janela de design é ampla e a profundidade da reflexão também. Já a janela de interação é curta, e a reflexão não ultrapassa as fronteiras estabelecidas pela situação real e concreta em que o usuário se encontra. O resultado desta assimetria – não surpreende – é, muitas e muitas vezes, bastante negativo.

Para encaminhar a formulação destes e de outros tipos de problemas associados, a escolha e o emprego de métodos adequados para a sua solução, bem como a avaliação das conseqüências de uma solução candidata (ou adotada), a Engenharia Semiótica oferece um conjunto de ferramentas epistêmicas para a reflexão do designer. No caso específico de aplicações de *groupware*, estas ferramentas estão calcadas em uma ontologia onde figuram não apenas um usuário, um sistema e um designer, mas também as estruturas de comunicação humana, o que se pode fazer com a fala, a etiqueta e os protocolos culturalmente aceitáveis ou esperados, e o nível de interferência da tecnologia sobre as relações sociais nos grupos [6;8]. No caso específico do estudo de caso relatado neste artigo, a Engenharia Semiótica ajuda a investigar as relações entre comportamentos de usuários e determinações tecnológicas, enunciando-as mais

claramente, e a formular questões de pesquisa relevantes para o campo de IHC.

O ESTUDO DE CASO

No estudo de caso realizado, investigamos o uso de duas ferramentas de discussão online específicas – os sistemas “Espaço Conversação” (EC) e “Espaço Decisão” (ED). – por dois grupos de participantes distintos. Observando como esses grupos usavam essas ferramentas, tínhamos um objetivo específico: a análise das influências das estruturas de captura de informações projetadas pelo designer sobre a organização do discurso dos participantes (organização textual interativa).

Neste artigo, os resultados obtidos serão apresentados e discutidos a partir de um dentre vários ângulos de análise possíveis. Este se refere à *discussão das implicações das estruturas de captura de informações e da organização textual interativa sobre as atividades de recuperação de informação*. Antes de passarmos à apresentação desses resultados, faremos, no entanto, uma breve descrição das ferramentas utilizadas, do perfil dos participantes e dos procedimentos metodológicos adotados.

Sobre as ferramentas utilizadas

Os sistemas ED e EC são dois fóruns de discussão desenvolvidos pelo Grupo de pesquisa de Engenharia Semiótica (SERG) da PUC-Rio.

Estas ferramentas apresentam algumas características em comum, tais como: o tipo de discurso resultante das interações dos participantes, os atributos de temporalidade, anonimato e apresentação da informação.

Tal como o discurso que resulta de outras modalidades de fóruns de discussão, o discurso produzido nas ferramentas deste estudo de caso mistura características da fala e escrita. Este discurso, apesar de escrito, é, comumente, informal, ágil e, principalmente, interativo, se aproximando em vários aspectos, da comunicação falada [19;15;2].

Além das características do discurso produzido nestas ferramentas (onde as características das modalidades escrita e falada representam um contínuo e não uma distinção), outras características podem ser definidas como semelhantes nas duas ferramentas, tais como os atributos de temporalidade, anonimato e apresentação da informação. As principais diferenças entre elas são: (a) o discurso utilizado pelo designer para apresentar cada ferramenta; e (b) a estruturação retórica de captura de informações, imposta por este discurso.

Em relação à temporalidade, em ambas as ferramentas a comunicação ocorre de forma assíncrona, o que ocasiona ter alguma demora entre o envio da mensagem, o seu recebimento e a resposta. Já em relação ao critério de anonimato, tanto o EC quanto o ED identificam seus participantes através de um apelido. As duas ferramentas também são similares em termos de modalidade de apresentação de informação. Ambas as tecnologias suportam apenas textos escritos que são organizados em falas (apresentem elas funções sintáticas ou não - tipos) que representam unidades do discurso do usuário. As

principais diferenças entre as duas ferramentas estão no discurso do designer e na estrutura retórica da discussão dada por este discurso. Enquanto o EC apresenta um discurso voltado para a conversação informal e uma estrutura flexível (como a maioria dos fóruns de discussão), o ED apresenta um discurso voltado para a discussão argumentativa, com uma estrutura rígida e sugestões de marcas linguísticas, projetadas para imprimir um significativo teor retórico à discussão.

Tabela 1. Características dos ambientes EC e ED

Atributos	Espaço Conversação (EC)	Espaço Decisão (ED)
Discurso dos participantes	Mais próximo do debate informal, falado.	Mais próximo da argumentação formal, escrita.
Temporalidade	assíncrono	assíncrono
Anonimato	Identificação de participantes através de apelido	Identificação de participantes através de apelido
Apresentação de informação	textos escritos	textos escritos
Estrutura	Flexível	Rígida
Discurso do designer	Sistema de conversação sobre diferentes temas	Sistema de reuniões para a tomada de decisões

“Espaço Conversação” (EC)

O EC é um sistema multi-usuário, onde pessoas registram ou declaram de forma escrita conceitos e opiniões sobre diversos temas.

Seu uso pode ser bastante diversificado, podendo ser utilizado para registro do trabalho cooperativo, auxiliando na consolidação de conhecimento sobre temas propostos em diversas vertentes de assuntos, e com diferentes características conversacionais.

A idéia chave do modelo do EC (Figura 1) é permitir que, abaixo dos temas criados, possam ser propostos diversos assuntos contendo falas não-tipadas que apresentam um relacionamento entre si.

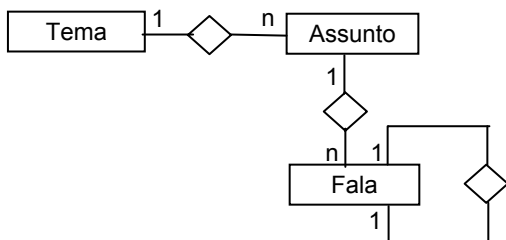


Figura 1: Modelo ER da estrutura de discussão no EC.

Este relacionamento pode ser caracterizado como um relacionamento de coordenação ou de subordinação, sugerindo ora o encadeamento entre as falas, ora a hierarquização entre elas.

De acordo com [10], o relacionamento de coordenação diz respeito ao paralelismo de funções ou valores sintáticos idênticos, tendo as falas a mesma natureza. No EC, este tipo de relacionamento é representado graficamente pela mesma posição na endentação entre as falas.

Já no relacionamento de subordinação não há paralelismo entre as falas, mas sim desigualdade de funções e de valores sintáticos (3). Há um relacionamento de hierarquização, onde a fala subordinada não subsiste sozinha sem o apoio da fala principal. Esta dependência entre as falas fica evidente na ferramenta por um tipo de linha de diálogo, representada graficamente. Esta representação mostra uma endentação diferenciada entre a fala principal e a(s) fala(s) subordinada(s).

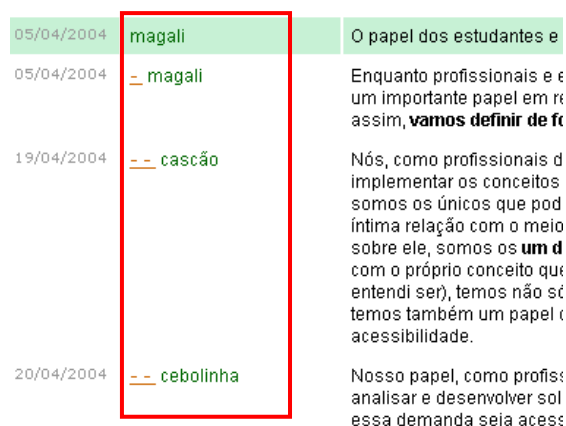


Figura 2: Representação das estruturas de coordenação e subordinação na interface do EC.

Apesar de existir uma representação gráfica para indicar os relacionamentos entre as falas dos participantes, acreditamos que esta representação não seja suficiente para imprimir teor retórico significativo à estrutura da discussão. Isto porque o discurso utilizado pelo designer no EC para representar esta estrutura é bastante genérico, apresentando signos de interpretação bastante aberta ..

Espaço Decisão (ED)

O sistema ED vem apoiado na idéia de dar suporte ao conflito de opiniões, por meio de um modelo de discussão argumentativa. O modelo de discussão está baseado no modelo IBIS [14], que divide as contribuições de um grupo em *Proposta*, *Opinião* e *Argumentação*.

Proposta é usada para levantar perguntas e tópicos para discussão; *opinião* é usada para expressar uma posição e responder a uma questão; e *argumentação* é usada para fornecer as razões onde se apoiam as opiniões.

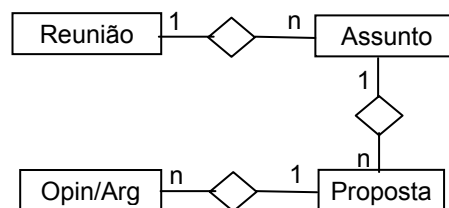


Figura 3. Modelo ER da estrutura de discussão no ED.

Outra característica deste sistema é a existência de marcas retóricas que se impõem a todo o momento na construção da discussão. Além das marcas de coordenação e subordinação — esta última um pouco diferente da forma tratada na ferramenta anterior, pois trata a fala do tipo *proposta* como a fala principal do período de qual todas as demais dependem — dadas pelos tipos diferenciados de marcadores de relacionamento entre as falas (Figura 4).

28/06/2004 magali inclui Assunto	Custos e Acessibilidade
▶ 07/04/2004 magali propoe	Sem dúvida todas as empres pesquisa e/ou desenvolver custos que este investimento
▶▶ 07/04/2004 magali opina a favor e argumenta	Afinal de contas a acessibilid.
▶▶ 07/04/2004 mônica opina a favor e argumenta	Alem disso Quando a navega satisfeito com a empresa e g
▶▶ 19/04/2004 anjinho opina a favor e argumenta	Quanto maior for a acessibili usuário alcançará os seus ob satisfeito em usar aquele sist grande retorno de investimer

Figura 4: Marcas de coordenação/subordinação e tipos de falas no ED.

O ambiente ED ainda apresenta sugestões de marcas lingüísticas no intuito de ajudar o usuário participante da discussão na escrita formal de suas propostas e argumentos. Estas marcas emergem de duas fontes principais, uma do contexto da argumentação formal e a outra do contexto da escrita do parágrafo [10].

Sobre o recrutamento dos participantes

Para que nosso estudo capturasse discussões o mais naturais e espontâneas possível, estabelecemos como critério-chave para o recrutamento dos participantes a sua familiaridade com tecnologias da informática. Com este critério em mente, visitamos as turmas da disciplina de Interação Humano-Computador (IHC), do Curso de Bacharelado em Informática da PUC-Rio, convidando seus alunos a participarem de uma discussão online. Nesta etapa, explicitamos claramente os objetivos da pesquisa (acima descritos), sua duração (3 semanas de discussão), o tema de discussão proposto (acessibilidade) e as tarefas exigidas (participação nas discussões do grupo). Foram, ainda, enfatizados os procedimentos éticos envolvidos no estudo (o caráter voluntário da participação, a garantia de anonimato, etc.). Como resultado, obtivemos a adesão de 38 estudantes, de ambos os sexos, que foram divididos em dois grupos: o grupo A, que participou da discussão do EC e o grupo B, das discussões do ED.

Sobre os procedimentos de coleta e de análise dos dados

Cada grupo de participantes foi previamente inscrito em uma das ferramentas, recebendo por email suas senhas de acesso. Isto feito, cada participante entrou (em hora e local de livre-escolha) no ambiente no qual foi inscrito, dando início às discussões sobre acessibilidade.

O tema acessibilidade foi escolhido por ser próximo à realidade dos alunos da disciplina de IHC (sem, no

entanto, exigir conhecimentos teóricos e técnicos específicos) e por ser uma questão aberta e polêmica (sem respostas certas ou erradas, e que comporta pontos de vista e sugestões heterogêneas). Para incentivar a discussão, em cada ambiente, havia: uma breve definição de acessibilidade, links para obtenção de informações sobre o tema, bem como subtemas específicos a serem desenvolvidos pelos participantes (garantia de acessibilidade, papel dos estudantes e profissionais de informática e de IHC e, finalmente, custos e acessibilidade).

Os grupos tiveram as 3 semanas previstas para encaminharem suas idéias e opiniões e, ao final deste período, a discussão foi encerrada, seguindo-se a análise dos discursos coletados.

Esta análise foi realizada com base na análise da conversação[13], abordagem oriunda da Lingüística e representante de uma importante tradição de pesquisa no campo da análise do discurso (ou seja, do estudo da língua em uso). Em outras palavras, a análise da conversação parte do pressuposto de que a realidade social é construída pelos atores sociais em suas interações. Em decorrência disto, estudar a realidade implica tomar por objeto de estudo os eventos cotidianos inseridos em seu contexto natural de ocorrência.

No contexto específico deste estudo, uma análise inspirada nos procedimentos da análise da conversação revelou-se interessante por uma razão específica: como dissemos, o discurso gerado em fóruns de discussão é similar a uma conversa cotidiana, aliando à escrita diversas das características da fala. Por este motivo, a análise do material coletado em nosso estudo guiou-se pela busca de regularidades lingüísticas nas “falas” dos participantes, regularidades estas capazes de trazer à tona informações sobre o uso das estruturas que caracterizam os dois ambientes (fracamente estruturada no caso do EC, e fortemente estruturada no ED).

Uma noção utilizada na análise da conversação mostrou-se especialmente útil na análise dos discursos gerados nos ambientes estudados, a saber, a noção de organização tópica. De forma muito simplista, organização tópica relaciona-se aos assuntos nos quais falantes/ouvintes mantêm sua atenção ao longo de uma determinada interação. Segundo Koch, no texto conversacional, esses assuntos podem ser delimitados e segmentados por meio dessa noção. A divisão do texto em unidades diferenciadas, acrescenta a autora, ajuda na constituição de uma organização tópica para a melhor compreensão desse texto. Nesta organização: “*fragmentos de um texto de nível mais baixo, podem ser denominados segmentos tópicos; um conjunto de segmentos tópicos formará um subtópico; diversos subtópicos constituirão um quadro tópico; havendo ainda um tópico superior que englobe vários tópicos, ter-se-á um supertópico*” (p.72) .

Em nosso estudo, o material discursivo gerado pelos dois grupos foi apreciado separadamente. A partir do supertópico previamente definido (acessibilidade), cada grupo gerou um texto conversacional específico. Passemos, então, à apresentação dos principais

resultados, travando contato com a análise da organização tópica desses textos conversacionais.

Apresentação dos principais resultados

Com o foco na estrutura geral da conversaç o no EC e no ED, a an lise do material revelou quatro pontos que merecem destaque: a) a forma da apresenta o dos subtemas da discuss o sobre acessibilidade, b) a rela o dos tipos de falas pr -definidos com a organiza o t pica do texto; c) o aceite ou o decl nio da estruturas de subordina o por parte dos participantes e, finalmente, d) a defini o ou indefini o do papel assumido pelos interlocutores ao longo das discuss es.

Categoria 1: A forma da apresenta o dos subtemas

Como dito anteriormente, foram inseridos nas ferramentas EC e ED, um tema (acessibilidade) e alguns subtemas id nticos que desencadearam o in cio das discuss es entre os usu rios de cada uma das ferramentas. No exato momento desta inser o, os pesquisadores perceberam que as diferen as de estrutura o (definida pelos designers) dessas ferramentas exigiam que a forma de apresenta o dos subtemas fosse tamb m diferenciada. Vejamos um exemplo desta diferen a, concentrando-nos na introdu o do subtema *Garantia de Acessibilidade*.

Na ferramenta EC, por exemplo, dada a aus ncia de tipagem das falas, foi poss vel inserir uma fala que introduziu diretamente o subtema em quest o.

- magali: *A acessibilidade da Internet possibilita maior flexibilidade no acesso   informa o e na intera o entre usu rios que possuam algum tipo de defici ncia ou necessidade especial, no que se refere aos mecanismos de navega o e apresenta o das p ginas,   opera o de softwares e hardwares e   adapta o de ambientes e situa es  s suas necessidades.*

Logo em seguida, o pesquisador introduziu uma afirma o complementar para estimular a participa o.

-- magali: *Certamente , o acesso aos benef cios da Internet deve ser assegurado, sem discrimina es ou exclus es, sendo necess rio considerar as caracter sticas e exig ncias pr prias dos cidad os com necessidades especiais (pessoas com defici ncias visuais, auditivas, motoras, cognitivas, pessoas analfabetas, entre outras). Voc  concorda?*

J  no ED, em fun o de suas caracter sticas estruturais, a inser o desse subtema teve que ser feita de forma invertida.

►07/04/2004 magali propoe: *Certamente o acesso aos benef cios da Internet deve ser assegurado, sem discrimina es ou exclus es, sendo necess rio considerar as caracter sticas e exig ncias pr prias dos cidad os com necessidades especiais (pessoas com defici ncias visuais, auditivas, motoras, cognitivas, pessoas analfabetas, entre outras). Voc  concorda ?*

►►07/04/2004 magali opina a favor e argumenta: *A acessibilidade na Internet possibilita maior flexibilidade no acesso   informa o e na intera o entre usu rios que possuam algum tipo de defici ncia ou necessidade especial, no que se*

refere aos mecanismos de navega o e apresenta o das p ginas,   opera o de softwares e hardwares e   adapta o de ambientes e situa es  s suas necessidades.

Neste caso, a afirma o que visava estimular a participa o do grupo foi a primeira ora o a ser inserida. Conseq entemente, o que no EC teve fun o de introdu o, no ED passou a ter fun o de argumento. Possivelmente, esta invers o ocorreu pelo fato de o EC ser fracamente estruturado e, portanto, permitir que a inclus o de falas acontea quase que simultaneamente ao processo cognitivo de constru o destas falas. Al m disso,   poss vel que o designer tenha se guiado pelos pressupostos da escrita de alguns g neros de textos, onde comeaamos sempre pela introdu o. J  no ED, a estrutura fortemente amarrada por tipos pr -definidos de falas leva aquele que se prop e a fazer uma proposta a organizar seu texto previamente, de forma a inserir suas falas dentro de uma estrutura composta apenas de proposta e argumento. Esta estrutura parece levar o proponente expor primeiramente o que pensa para, em seguida, justificar este pensamento.

Apesar das diferentes apresenta es da discuss o, ambas as ferramentas geraram f rteis textos conversacionais. Cada um desses textos, no entanto, apresentou uma determinada organiza o t pica e rela es distintas entre suas falas. Vejamos como isto ocorreu.

Categoria 2: Rela o dos tipos de falas pr -definidos com a organiza o t pica do texto

Conforme j  descrevemos, no ED, o designer definiu previamente uma rela o de tipos de fala. Esta pr -defini o facilitou a realiza o de um mapeamento sint tico da organiza o t pica [13] da discuss o (tabela 2).

Tabela 2: Mapeamento das falas do ED para organiza o t pica

Falas pr�-definidas	Organiza�o t�pica
Reuni�o sobre	Supert�pico
Assunto	Quadros t�picos
Proposta	Subt�picos
Argumento	Segmentos t�picos

Idealmente, a partir das estruturas pr -definidas, seria esperado que as discuss es ocorridas no ED refletissem o mapeamento acima (i.e., que as falas fossem postadas segundo a tipagem definida pelo designer). Em grande parte da discuss o do ED, o esperado ocorreu, dado que a maioria das falas foi postada da maneira apropriada.

A an lise mais cuidadosa do material discursivo do ED revelou, no entanto, que, de forma recorrente, v rias falas burlaram as estruturas pr -definidas pelo designer. Durante a intera o com a ferramenta, v rios dos participantes criaram outros t picos e subt picos al m daqueles originalmente propostos, seguindo caminhos n o previstos pelo designer. A ocorr ncia desses casos faz com que seja poss vel representar a estrutura t pica da discuss o do Espaço Decis o de duas maneiras. A

primeira (Figura 5) reflete o que ocorreu na prática. Reúne tópicos e subtópicos marcados sintaticamente no texto (por meio das estruturas propostas pelo designer) e outros tópicos e subtópicos implícitos, posto que sem marcações sintáticas claramente especificadas. Já a segunda maneira (Figura 6) reflete tão-somente a organização retórica idealizada pelo designer.

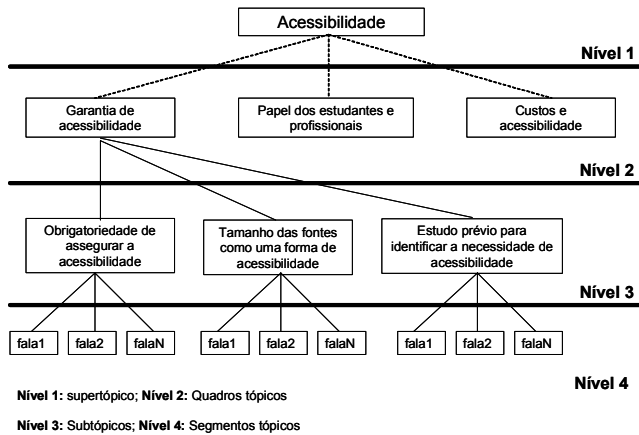


Figura 5: Organização tópica que ocorreu na prática

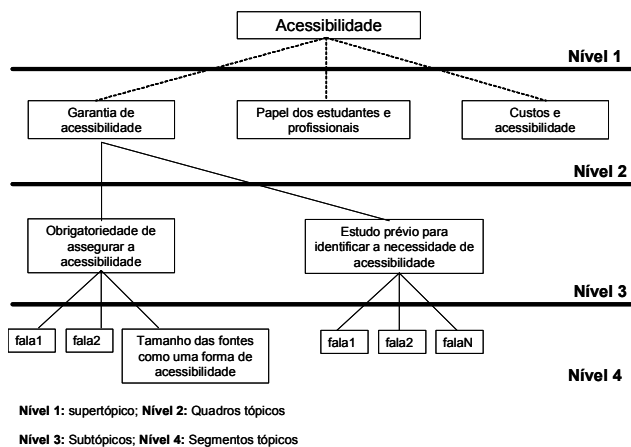


Figura 6: Organização retórica projetada pelo designer

Por meio das figuras acima, é possível verificar que o subtópico “Obrigatoriedade de assegurar a acessibilidade” foi introduzido pelo designer com o intuito de iniciar as discussões sobre a garantia de acessibilidade na Internet, sendo previamente disponibilizado aos participantes. Já os subtópicos “Tamanho das fontes como uma forma de acessibilidade” e “Estudo prévio para identificar a necessidade de acessibilidade” foram inseridos pelos participantes no decorrer da discussão.

O subtópico “Tamanho das fontes como uma forma de acessibilidade” é um exemplo de como vários dos participantes burlaram as estruturas de fala propostas pelo designer. Este subtópico é introduzido em meio a um segmento tópico já existente (Figura 7). O participante, no entanto, o introduz por meio de uma fala do tipo “opinião/argumento”. Contudo, além de dar sua opinião,

insere outro subtópico no assunto proposto. Neste caso, apesar de ter criado um outro subtópico de discussão, o usuário não criou uma fala do tipo proposta, logo não introduziu uma marca sintática deste tipo de fala.

07/04/2004 magali propoe
Certamente o acesso aos benefícios da Internet deve ser assegurado, sem discriminações ou exclusões, sendo necessário considerar as características e exigências próprias dos cidadãos com necessidades especiais (pessoas com deficiências visuais, auditivas, motoras, cognitivas, pessoas analfabetas, entre outras) Você concorda?

07/04/2004 magali opina a favor e argumenta
A acessibilidade na Internet possibilita maior flexibilidade no acesso à informação e na interação entre usuários que possuam algum tipo de deficiência ou necessidade especial, no que se refere aos mecanismos de navegação e apresentação das páginas, à operação de softwares e hardwares e à adaptação de ambientes e situações às suas necessidades.

08/04/2004 franjinha opina a favor e argumenta
Pessoas com dificuldades visuais, em determinados sites necessitam fazer um esforço para ler determinadas fontes de tamanhos pequenos. Até mesmo durante a noite, onde a vista já está mais “cansada”. Acho que seria interessante resolver este problema com tamanho de fonte.

15/04/2004 titi opina a favor e argumenta
Com o propósito de facilitar a leitura para pessoas com dificuldades em ler letras miúdas, os browsers dão ao usuário a opção de aumentar o tamanho da fonte. Isso é suficiente, ou a interface (xite) deve também oferecer essa opção aos usuários?

19/04/2004 mônica opina a favor e argumenta
Em minha opinião a opção oferecida pelo browser não é suficiente, pois acredito que os usuários em geral não saibam da existência deste recurso. Oferecer essa opção no site é uma maneira mais explícita de favorecer os usuários com problemas visuais. E ainda, já que o usuário tem esse problema, a localização da opção para aumentar a fonte deve ser cuidadosamente estudada, para que seja de fácil localização.

Figura 7: Subtópico introduzido em meio a um segmento tópico já existente

Como a figura 8 revela, apesar de não apresentar a marca sintática do tipo proposta, a fala sobre o tamanho da fonte parece ter sido – no contexto da discussão do grupo – reconhecida e aceita como uma proposta. Isto porque, as falas postadas em seqüência por outros participantes veiculam opiniões e argumentos em favor da possibilidade de aumento da fonte nos browsers. Em outras palavras, este subtópico tornou-se uma proposta e mereceu a atenção do grupo, apesar de não estar em destaque como era a intenção do designer da ferramenta.

Por sua vez, o subtópico denominado “Estudo prévio para identificar a necessidade de acessibilidade” (Figura 8) ilustra a inserção consciente do falante, cuja função interativa é introduzir uma proposta complementar à original. Para marcar esta função, o falante insere sua idéia em uma fala do tipo “proposta”. Esta, além de ter valor semântico na organização tópica do texto (de subtópico), apresenta valor semântico no escopo de projeto da ferramenta (de proposta).

29/04/2004 xaveco opina a favor e argumenta
Em minha opinião Todas as fontes didáticas relativas a internet deveriam conter um tópico como se fosse uma “letra”, sendo assim a melhor forma de agir seria pelo bom senso, basta conscientizar o programador a ter uma visão mais flexível para as possíveis dificuldades que alguns usuários poderiam ter.

29/04/2004 dudu opina contra e argumenta
Discordo por que se o objetivo de seu negócio tem um propósito específico, você acaba tendo que discriminar uma parcela dos usuários de internet para você conseguir obter sucesso no objetivo traçado.

19/04/2004 mônica propoe
Além disso acredito que todos os sites devem fazer um estudo prévio do seu público alvo e, caso neste se incluam pessoas com dificuldades especiais, os recursos adaptativos são fundamentais.

19/04/2004 mônica opina a favor e argumenta
Assim todos os usuários poderão fazer uso do site oferecido.

19/04/2004 anjinho opina a favor e argumenta
Além disso, com a garantia da acessibilidade na internet, haverá a redução da exclusão digital, promovendo o uso democrático da internet, como este deve ser.

20/04/2004 humberto opina contra e argumenta
Discordo parcialmente com o estudo prévio do público alvo de um site. Acho que pesquisas sobre público alvo distinguem usuários quanto aos gostos e interesses, por exemplo. Não dá para se distinguir um público alvo a partir de um critério que diga se esse público tem pessoas com algum tipo de deficiência física. Acho que todos os sites devem ter a preocupação com facilidades de visualização, pois uma pessoa, por mais que tenha uma deficiência física, tem o direito de se interessar por qualquer conteúdo que seja apresentado na internet.

Figura 8: Subtópico introduzido em uma fala do tipo “proposta”

Já a organização textual das discussões no “Espaço Conversação” (EC) apresentou características bem distintas das discutidas acima. Dado que fracamente estruturado, o design do EC torna explícito somente o supertópico (acessibilidade), nomeado na ferramenta sob o signo de *tema*, e seus quadros tópicos (garantia de acessibilidade, papel dos estudantes e profissionais de IHC e custo), referidos na interface como *assunto*. As noções de tópicos, subtópicos e segmentos tópicos não estão explicitadas na estrutura da ferramenta, embora possam ser sutilmente percebidas em algumas falas dos participantes. Isto só é possível, contudo, por meio da

identificação de recursos lingüísticos utilizados pelos falantes. Explicitamente, portanto, não há, no EC, nenhuma hierarquia tópica entre as falas. (Figura 9).

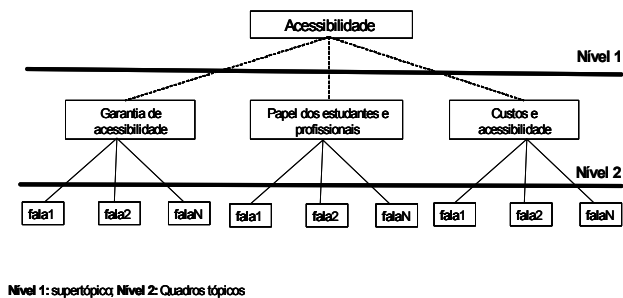


Figura 9: Organização tópica projetada pelo designer do EC.

A análise do material discursivo gerado no EC revelou, ainda, outro interessante resultado. Na discussão dos participantes, as falas estavam, em sua maioria, relacionadas aos quadros tópicos previamente introduzidos pelos pesquisadores. Grande parte dos participantes não postou falas que expressassem novas propostas de quadros tópicos (apenas dois deles inseriram falas cujo conteúdo veiculava um novo quadro tópico de discussão). Possivelmente, a maior linearidade na conversação e a falta de novas propostas de discussão no EC foram aspectos influenciados pelo design da interface. Como vimos, no ED, o próprio signo proposta, utilizado na interface, parece motivar a inclusão de novas falas deste tipo. Por sua vez, no EC, a ausência deste signo parece encaminhar a discussão para opiniões/argumentações sobre temas anteriormente disponibilizados, sem motivar o usuário a criar novas propostas para a discussão.

Categoria 3: Aceite ou declínio da estrutura de subordinação

No ED, a estrutura de subordinação define que argumentos são sempre subordinados a uma proposta. A forma como esta estrutura foi projetada não permite que exista a subordinação formal entre argumentos. Ainda assim, durante a análise do material discursivo coletado, foi possível identificar que os participantes criaram uma estrutura de subordinação *informal* para relacionar suas falas às de outros usuários, usando para este fim alguns recursos lingüísticos. Vejamos um exemplo:

“Em minha opinião a opção oferecida pelo browser não é suficiente, pois acredito que os usuários em geral não saibam da existência deste recurso. Oferecer esta opção no site é uma maneira mais explícita de favorecer os usuários com problemas visuais. E ainda, já que o usuário tem este problema, a localização da opção para aumentar a fonte deve ser cuidadosamente estudada, para que seja de fácil localização”.

Os recursos do *Browser* referidos nesta fala foram mencionados em um argumento anterior e não na proposta a qual a fala estava subordinada. Há, deste modo, um encadeamento entre fala e argumento anterior através da referência ao texto contido nesse mesmo argumento (referência à palavra chave). Através deste recurso, o usuário explicitamente define que o foco de sua atenção é o argumento referente aos recursos do browser e não a fala do tipo “proposta”. Outra fala ilustrativa da subordinação informal entre argumentos é a de quinzinho.

“De acordo com a opinião de rkg, também acho a elaboração de padrões um ponto de extrema importância no desenvolvimento de recursos de acessibilidade. As teclas de atalho, ações de contorno, feedback alternativos, etc...seriam facilmente reconhecidas independentemente do software utilizado.”

A fala de quinzinho ilustra um caso de uso da referência ao autor de uma mensagem como recurso para mostrar que um argumento está relacionado ao argumento de outro participante e não à proposta original. Por meio desta a referência ao autor, este usuário cria uma maneira (não prevista pelo designer) de imprimir uma relação de subordinação entre falas. No ED, os recursos de referência (de palavra chave e autor) foram recorrentemente usados para burlar a estrutura de subordinação pré-definida no projeto desse ambiente.

Já no EC, o relacionamento de subordinação entre falas é um recurso previsto pelo designer e disponibilizado na interface. Por essa razão, os participantes da discussão nesse ambiente não utilizaram nenhum recurso lingüístico para expressar este tipo de relacionamento. Cabe registrar, contudo, uma interessante ocorrência isolada observada na discussão deste grupo. Esta se refere à ação de um participante que postou uma única fala subordinada a duas outras falas distintas (Figura 10).

- magali Devido ao grande fator social todas as empresas desenvolvedoras de software **devem** investir em pesquisa e/ou desenvolvimento na área de acessibilidade, **independentemente** dos custos que este investimento venha a gerar.
- jeremias Dizer que *todas* as empresas desenvolvedoras de software **devem** investir em pesquisa e/ou desenvolvimento nesta área, independente dos custos envolvidos, é um exagero, uma vez que tais custos devem ser elevados e duradouros (ou contínuos).
Acredito que deveriam existir empresas especialistas na área de desenvolvimento de software com foco em acessibilidade, já que o perfil dos usuários necessitando de tratamentos especiais é bastante diferenciado e com grandes particularidades.
- pelezinho Acredito que as interfaces gráficas deveriam conter recursos que as tornassem independentes do monitor, como procuram ser em relação ao mouse. Com isso, a deficiência visual, a falta de óculos (que acontece muito) e, até mesmo, os monitores defeituosos não impediriam o acesso ao serviço. Por exemplo, os objetos poderiam não só esboçar 'hints' mas falá-los. Isso não parece ser tão complicado... Com isso, outros cuidados seriam tomados pelos desenvolvedores para que suas interfaces fossem 'monitor-free'.
- cascão Discordo. Uma avaliação deve ser feita para a obtenção do custo/benefício de se investir em uma pesquisa de acessibilidade. Deve-se avaliar se o retorno vai cobrir os custos e se, ao menos a longo prazo, um lucro maior será esperado. Outro fator é a avaliação do público alvo do qual a empresa pretende atingir. Investir em uma pesquisa que visa abranger um mercado diferente daquele pretendido é gasto desnecessário. Finalmente, concordo que devam existir empresas de desenvolvimento específicas na área de acessibilidade. Elas poderiam, então, prestar serviços à quem estivesse interessado em investir para obter uma solução útil com custos realistas e retornos desejáveis.

Figura 10: Fala que se refere a outras duas falas postadas anteriormente

Neste trecho da discussão, temos uma fala de magali e, subordinada a esta, uma fala de jeremias. Logo abaixo, cascão insere uma única fala em resposta a magali (“Discordo. Uma avaliação deve ser feita para a obtenção do custo/benefício...”) e a jeremias (“Finalmente, concordo que devam existir empresas de desenvolvimento específicas...”). Apesar de haver a possibilidade de fazer a relação de subordinação em separado (i.e. postar uma fala em resposta a magali e outra em resposta a jeremias), cascão, ao unificar sua resposta, acabou por negligenciar o recurso de subordinação oferecido pela ferramenta EC.

Em resumo, a maioria dos participantes das discussões no EC e no ED fizeram uso dos recursos de subordinação oferecidos pelas duas ferramentas. Alguns participantes

de ambos os grupos de discussão, no entanto, declinaram desses mesmos recursos, criando novos e informais relacionamentos de subordinação entre falas.

Categoria 4: Manifestações de papéis e a relação com tipos de falas e estruturas de subordinação

A inserção de uma nova fala também pode ser analisada sob o ponto de vista das manifestações de papéis entre os falantes. Neste contexto, a introdução de falas permite que sejam postas em prática diferentes tipos de relações sociais entre o mesmo conjunto de indivíduos participantes da interação.

No caso específico do ED, uma modalidade de relação social possível é aquela pautada na igualdade de posições dos participantes. Esta relação se dá, por exemplo, quando dois usuários – assumindo os papéis de opinantes – dão argumentos a uma proposta. Ocorre, porém, que esses mesmos opinantes podem, a qualquer momento da discussão, estabelecer outro tipo de relação. Isto ocorre na medida em que um deles insere uma nova proposta. Neste caso, temos uma relação de proponente x opinante.

A relação de proponente pode surgir de duas maneiras: A primeira diz respeito a uma situação na qual um usuário inclui seu texto em uma fala do tipo proposta e recebe opiniões sobre ela de outro participante. Já a outra maneira diz respeito à inclusão de uma fala que informalmente revela uma proposta, mesmo que não esteja tipada como tal. Caso esta fala seja qualificada por outro participante como uma proposta (i.e. caso esta fala obtenha respostas e argumentos de outro usuário), a relação entre estes participantes também será a de proponente x opinante. Segue-se um exemplo.

19/04/2004 aninho
opina a favor e argumenta

19/04/2004 quinzinho
opina a favor e argumenta

19/04/2004 aninho
opina a favor e argumenta

O nosso papel, enquanto profissionais de informática é de desenvolver sistemas que garantam a acessibilidade, além de pesquisar novos artifícios para melhorá-la. Também vale a divulgação desses princípios, em nosso ambiente de desenvolvimento, ou em nossas próprias homepages, que servirão como exemplos de interfaces com boa acessibilidade.

Atualmente, não temos nenhuma disciplina na grade do curso de bacharelado que seja direcionada para este assunto. Em IHC, conhecemos apenas os conceitos e premissas básicas sobre interfaces. O assunto IHC é bem vasto e poderia ser desdobrado numa segunda disciplina, onde seriam abordadas situações específicas, onde um trabalho mais elaborado de adaptação a interface é exigido. Nesta disciplina, poderiam ser adivulgados estudos de caso e os alunos seriam avaliados por projetos, desenvolvidos em grupo, onde seriam apresentadas soluções para problemas de acessibilidade. Cada grupo tratando um problema específico, como: deficiência visual, motora, etc...

Concordo. É, mais do que incluir uma nova disciplina de IHC, seria interessante constituir uma nova linha de especialização com disciplinas relacionadas a essa área, tendo em vista que se trata de um conceito amplo que ainda rende teses de doutorado.

Figura 11: Fala qualificada como proposta

O trecho acima é parte de uma discussão a respeito do papel do estudante e dos profissionais de IHC em relação à acessibilidade. Em um determinado momento, um participante cria uma fala do tipo opinião/argumento, propondo o desdobramento da disciplina de IHC visando contemplar tópicos sobre acessibilidade. Logo a seguir, outro falante qualifica esta fala como uma proposta (quando dá sua opinião sobre ela) e transforma a relação entre esses participantes em uma relação do proponente x opinante. Estas diferentes modalidades de relação são facilmente visualizadas em função do design da interface do ED (no qual as falas estão tipadas).

Já no EC, todos os participantes têm, a priori, o mesmo papel (i.e., todos são autores de falas). Ao longo das discussões, o reconhecimento de mudanças no papel de autor não é trivial, dado que o sistema não representa as falas através de tipos pré-definidos. A identificação de novos papéis só é possível, como mostra a figura abaixo,

por meio da análise do conjunto das falas de uma discussão.

magali
Enquanto profissionais e estudantes de informática desempenhamos um importante papel em relação a acessibilidade na Internet. Sendo assim, **vamos definir de forma mais específica que papel é este ?**

cascão
Nós, como profissionais de informática, podemos pesquisar, definir e implementar os conceitos de acessibilidade. Obviamente, nós não somos os únicos que podemos estar envolvidos, mas por possuímos a íntima relação com o meio em questão, por termos o conhecimento sobre ele, somos os **um dos mais** envolvidos. Dessa forma, de acordo com o próprio conceito que a acessibilidade envolve (ou o que eu entendi ser), temos não só o papel de agir como profissionais, mas temos também um papel cívico e moral de disponibilizar esta acessibilidade.

cebolinha
Nosso papel, como profissionais e estudantes de informática, é analisar e desenvolver soluções para algum tipo de demanda. Caso essa demanda seja acessibilidade, cabe a nós desenvolvermos soluções que tragam benefícios de acessibilidade ao público demandante (também conhecido como usuário).

Figura 12: Conjunto de falas

A título de ilustração, a análise desse trecho de discussão revela que a primeira fala gera duas novas falas a ela subordinadas. Tomando este dado como pista, uma leitura mais cuidadosa da discussão leva-nos à constatação de que a primeira fala é uma proposta. Em decorrência disto, é possível reconhecer que há uma relação entre proponente e opinantes. Em comparação com a fácil visualização dos papéis no ED, o reconhecimento das relações sociais presentes em uma discussão no EC é muito mais complexo e sofisticado.

A identificação dos papéis existentes em uma discussão online, da questão originária dessa discussão, da organização tópica de seu conjunto de falas e das relações de subordinação entre essas falas foi o eixo central da apresentação dos resultados deste estudo. Os quatro pontos de análise adotados estão intimamente ligados à estrutura de captação de informação dos sistemas ED e EC. Nosso estudo revelou algumas diferenças entre as estruturas dos dois sistemas, enfatizando que essas são fortemente determinadas pelas decisões de projeto dos designers. A análise das formas de uso desses sistemas revelou, por sua vez, que seus usuários freqüentemente não seguem os caminhos de utilização idealizados pelos projetistas. Cabe, neste momento, discutir as implicações destes achados sobre a atividade de recuperação da informação em fóruns de discussão.

Considerações Finais

No âmbito computacional, a tarefa de mapear uma pergunta em um conjunto de possíveis documentos relevantes é chamada de recuperação de informações[20]. Especificamente em um sistema do tipo Fórum de discussão, permitir o acesso ao histórico do grupo e à memória das discussões, são exemplos de recuperação de informações possíveis e necessárias. A maneira mais explícita através da qual esta tarefa é apresentada aos usuários se dá por meio de mecanismos de busca. Todavia é importante considerar que tarefas que são praticamente inerentes à interação dos usuários, como por exemplo diferentes formas de visualização das informações, também envolvem mecanismos de recuperação implícitos.

Dada a sua importância para o bom andamento das discussões em ambientes de discussão online, a tarefa de recuperação de informações deve ser consistente.

Neste trabalho, verificamos que a recuperação consistente de informações é fortemente influenciada pelo design da ferramenta (arbitrado pelo designer) e, também, pelas diferentes maneiras que seus usuários se apropriam desta ferramenta. Observamos, por exemplo, que decisões de design voltadas estrategicamente para a recuperação consistente das informações podem ser sistematicamente burladas por alguns usuários, o que certamente impacta negativamente a consistência de recuperação almejada. Passemos, então, à discussão dos principais achados de nosso estudo.

Em relação à estrutura dos fóruns de discussão, foi possível observar que o fato de o EC ser fracamente estruturado dificulta a atividade de recuperação das informações e faz com que esta atividade envolva técnicas sofisticadas de processamento de linguagem natural (o único dado acessível de maneira direta é a quantidade de temas discutidos. Já a opção de design do ED (que cria uma estrutura retórica de captura e organização de falas) facilita fortemente a atividade de recuperação. No ED, a recuperação do conjunto de propostas postadas, por exemplo, seria tarefa trivial.

No contexto de uso, entretanto, esta recuperação esbarraria em vários obstáculos. Isto porque, como foi possível constatar, vários participantes da discussão no ED, em diferentes momentos, declinaram das opções oferecidas pelo designer. Este foi o caso, por exemplo, da postagem de falas do tipo “opinião/argumento” que, na realidade, tinham a função sintática de proposta. Ao recuperarmos o conjunto de propostas do grupo, falas deste tipo ficariam excluídas da recuperação. Em resumo, a definição de uma estrutura como a encontrada no ED facilita a recuperação de informações, mas não é suficiente para garantir a integridade destas informações em certos tipos de consulta. As diferentes formas de utilização dos fóruns de discussão examinados também revelaram implicações do uso sobre a recuperação no que diz respeito aos relacionamentos entre falas no EC e no ED. Observamos, por exemplo, que, no ED, mesmo o relacionamento (de coordenação e subordinação) entre falas sendo pré-definido (no momento da captação da informação), os usuários encontram formas de burlar este relacionamento por meio de recursos lingüísticos. À título de ilustração, no ED, a recuperação do número de argumentos vinculados a uma proposta seria tarefa fácil. Mais uma vez, no entanto, o contexto de uso aponta para a distância entre o ideal do designer e a experiência do usuário. Em nosso estudo, dado que vários usuários postaram argumentos que (por meio de referências de autor, de objeto, etc) se relacionavam a outros argumentos, a contabilização de argumentos relativos a uma proposta não seria, de fato, consistente.

Em um espectro mais geral podemos verificar que a dificuldade na tarefa de recuperação de informações, se dá ou por falta de estruturas pensadas a priori para facilitar a recuperação ou pelo fato do usuário perder ou declinar as *affordances* disponibilizadas pelo designer. Os usuários encontram formas de burlar estas estruturas deixando assim, comprometida a eficiência do projeto para a recuperação. Alguns trabalhos [9,12] já haviam

apontado problemas do modelo IBIS em relação a recuperação de informações, justamente pela perda ou declínio de *affordances* das estruturas pré-definidas.

Acreditamos que a distância entre o ideal do designer e a real experiência do usuário ao usar o sistema possa estar relacionada :

- ao conhecimento sobre a importância destas estruturas para a recuperação de informação pelos usuários. Eles não dão mostras de dar importância ou de saber a importância destas estruturas. Mesmo os usuários da área da informática (como é o perfil dos participantes de nosso estudo) não dão mostras de atentar para a importância de estruturar o discurso no momento de sua *produção* para garantir a consistência no momento de sua *recuperação*
- à diferença entre a forma de conversação proposta e a conversa cotidiana e informal que os usuários têm como experiência. A estrutura impõe uma forma de conversação não familiar aos usuários, posto que muito diferente da conversação cotidiana e informal (suas falas não se atêm aos tipos proposta e argumento; eles podem se dirigir a vários interlocutores e a vários tópicos de discussão na mesma fala, etc.), numa tentativa *deliberada* feita pelo designer de transmitir a importância e conveniência de estruturar o discurso na hora da *produção* com vistas a dar mais qualidade e utilidade às funções de recuperação.
- à necessidade de categorização das falas., o que não é feito pelos usuários em situação de comunicação natural, onde esta categorização é *inferida* pelos seus interlocutores que compartilham com ele da mesma competência lingüística e discursiva

De uma perspectiva de projeto centrado no usuário, contudo, sabemos que é de responsabilidade do designer projetar artefatos que ao menos minimizem os problemas encontrados por estes usuários durante a interação com o sistema. Para isto, acreditamos que, em relação aos sistemas de fóruns de discussão que visam a recuperação de informações, os designers devem:

- tornar a interação com as estruturas de captação de informação mais simples e mais agradável, fazendo com que o usuário tenha que despende o menor esforço possível com a atividade de marcar suas falas;
- fazer isto respeitando a visão focada do usuário;
- dado que isto ainda requeira algum esforço do usuário, deixar clara, na interface, sua visão sobre a importância da recuperação em longo prazo (relação custo x benefício), motivando, assim os usuários a utilizarem as estruturas requeridas.

Confrontados a esta problemática somos levados a acreditar na relevante contribuição da área de IHC e principalmente da Engenharia Semiótica no que se refere à disponibilização de insumos para que o designer encontre soluções de comunicação eficientes e aceitáveis

para harmonizar a utilização pelos usuários com o seu projeto de captação de informação.

Para finalizar, identificamos de imediato duas questões importantes para serem investigadas em trabalhos subsequentes:

Primeiro, a Engenharia Semiótica preconiza que os designers comuniquem mais ampla e eficientemente a sua visão de design para os usuários, com vistas a motivarmos a adotar um tipo de interação com a tecnologia que maximize a sua eficácia e eficiência ao ser usada. Ambos os fatores estão fortemente associados a um dos critérios centrais de usabilidade – a satisfação do usuário. Porém, o espectro de valores com que um designer trabalha *em tempo de design* e com que um usuário trabalha *em tempo de interação* são muito diferentes. Este trabalho mostra um exemplo simples. Em tempo de design, os valores de produção, visualização e recuperação de informações contida nas discussões do fórum são ponderados **conjuntamente** e a **decisão de design** reflete a melhor relação de custo/benefício. Já em tempo de interação, os valores de produção, visualização e recuperação são ponderados **separadamente** e a **escolha da estratégia de interação** reflete a melhor relação custo/benefício dentro do contexto específico da atividade focal do usuário, sem necessariamente levar em consideração *o quanto a decisão de agora vai custar depois*. A questão importante de pesquisa aqui é saber se podemos permitir ao usuário **declinar conscientemente uma affordance de design** [7] e, o que é fundamental, saber o custo, para o usuário, de escolhas **conscientes** de estratégias interativas.

Outra importante questão, que pode ser tratada dependente ou independentemente da anterior, é saber o decaimento qualitativo de uma tecnologia quando os usuários *declinam affordances* que desempenham um papel crítico na visão tecnológica global que empresta valor ao produto como um todo. Ou seja, um fórum eletrônico dentre cujas qualidades e vantagens consta a “recuperação e visualização de informações sobre o teor das discussões” na realidade só funciona com a aceitação e adesão dos usuários ao custo (interativo) dos benefícios (utilitários) que a tecnologia oferece. É frequente observarmos o descontentamento dos usuários ou a inutilidade de serviços em ambientes tecnológicos onde a relação custo/benefício em tempo de interação leva os usuários a *burlarem* estruturas (tal como ocorreu no estudo que apresentamos). Parece-nos interessante trazer para o escopo da interação e da tomada de decisão em grupo alguns temas que certamente são discutidos com os usuários em tempo de design. Por exemplo, se e como fazer a recuperação de informações, diante de requisitos funcionais ou não-funcionais associados a cada alternativa considerada. Em outras palavras, a questão interessante para uma engenharia semiótica de IHC para *groupware* seria permitir ao grupo decidir sobre como e por que usar a tecnologia desta ou daquela maneira. Este tipo de discussão já foi anteriormente tratado em trabalhos do SERG [1;3;16], mas nenhum destes trabalhos focou explicitamente o processo de decisão em grupo em tempo de interação.

Nossos trabalhos futuros adiantarão ao menos parte das duas questões identificadas acima, além de aprofundar a análise sobre outras marcas linguísticas que surgiram durante a interação dos participantes nas discussões citadas. Pretendemos também estudar a teoria da estrutura retórica RST [18], como uma possibilidade de prover outros recursos linguísticos para captura e recuperação de informações.

REFERÊNCIAS

- [1] Barbosa, C.M.A. (2002) MetaCom-G*: Especificação da Comunicação entre Membros de um Grupo. Dissertação de Mestrado, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil. Março de 2002.
- [2] Churchill, E. and Erickson, T. "Introduction to this Special Issue on Talking about Things in Mediated Conversations" In Human Computer *Interaction*, Vol. 18 Nos. 1&2, pp 1-12. Lawrence Erlbaum Associates, 2003.
- [3] Cunha, C.K.V. (2001) Um Modelo Semiótico dos Processos de Comunicação Relacionados à Atividade de Extensão à Aplicação por Usuários Finais Tese de doutorado, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil. Agosto de 2001.
- [4] de Souza, C.S. (1993) The Semiotic Engineering of User Interface Languages International Journal of Man-Machine Studies 39,753-773.
- [5] de Souza, C.S. (2004) Semiotic Engineering: bringing designers and users together at interaction time. *Interact Human Computer*, vol 16 in press.
- [6] de Souza, C.S. The Semiotic Engineering of Human-Computer Interaction. The MIT press, 2004.
- [7] de Souza, C.S.; Prates, R.O.; Carey, T. (2000) Missing and Declining Affordances: Are these Appropriate Concepts? *Journal of the Brazilian Computer Society* vol.6 n.3. Campinas - SP, 2000
- [8] de Souza, C. S., Preece, J. A Framework for analyzing and understanding online Communities. *Interact Human Computer*, vol 16. In Press.
- [9] Garcia, A. C. B. Active design Documents: A new Approach for supporting documentation in preliminary routine design. Department of civil engineering and the committee on graduate studies of Stanford university. PhD dissertation. August 1992.
- [10] Garcia, Othon. M. Comunicação em prosa moderna. 3ª Edição. Rio de Janeiro: Fundação Getúlio Vargas, 1975.
- [11] ISO 13407: Human-centred design processes for interactive systems (1999).
- [12] J Conklin and ML Begeman. gIBIS: A hypertext tool for exploratory policy discussion. *ACM Transactions on Office Information Systems*, 6(4):303--331, October 1988.
- [13] Koch, I.G.V. A Inter-ação pela Linguagem. 2ª ed. São Paulo: Contexto, 1995.
- [14] Kunz, W. Rittel, H. (1970). Issues as elements of information systems. Working Paper No. 131, Institute of

Urban and Regional Development, University of California at Berkeley, Berkeley, California, 1970.

- [15] Marcoccia, M. (2004). On-line polylogues: Conversation structure and participation framework in internet newsgroups. *Journal of Pragmatics* Volume: 36, Issue: 1 January, 2004
- [16] Prates, R. O. (1998) Engenharia Semiótica de Linguagens de Interfaces Multi-Usuário Tese de doutorado, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil, outubro de 1998
- [17] Selbach, M. S. Metacomunicação designer-usuário na interação humano-computador : design e construção do sistema de ajuda. Tese de doutorado. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. Rio de Janeiro, Brasil, 2002.
- [18] Welcome to the RST Web Site. Disponível em: <http://www.sil.org/~mannb/rst>.
- [19] Yates, S. J. (1996). Oral and written linguistic aspects of computer conferencing: A corpus-based study. In Susan C. Herring (Ed.), *Computer-mediated communication: Linguistic, social and cross-cultural perspectives*. Philadelphia: John Benjamins Publishing Co.
- [20] Yang, M. C., Wood, W. H., Cutkosky, M. R. Data Mining for Thesaurus Generation in Informal Design Information Retrieval. *Proceedings of the 1998 International Congress on Computing in Civil Engineering*. Boston, MA, Oct. 18-21, 1998.

Ferramentas para Apoio à Construção de Sistemas de Ajuda Online

Alex Pacheco da Silva

André Luiz Farias

Ana Paula Lorenzoni

Micheline Parizotto

Milene Silveira

Faculdade de Informática/PUCRS
Avenida Ipiranga, 6681 – Prédio 30
Porto Alegre – RS – Brasil
milene@inf.pucrs.br
55 51 33203558

RESUMO

Este trabalho apresenta um conjunto de ferramentas para apoio aos *designers* na construção de sistemas de ajuda *online*, com base em modelos de *design* de Interação Humano-Computador (IHC) e na proposta de ajuda *online* da Engenharia Semiótica. Neste conjunto encontram-se o Editor de Help, ferramenta utilizada para o armazenamento das informações dos diferentes modelos de IHC e para geração automática de um rascunho do conteúdo da ajuda; o Editor do Conteúdo de Ajuda, ferramenta que auxilia no refinamento das informações geradas; e, a Biblioteca de Funções de Ajuda, conjunto de funções que facilitam a criação de um sistema de ajuda *online* de acordo com a arquitetura proposta.

ABSTRACT

This work presents a set of tools to help designers to build online help systems based in HCI design models and in the Semiotic Engineering view of help systems. The tools herein presented are a Help Editor, to help designers to maintain information about the design models and to generate a draft of the help content; a Help Content Editor, to aid into the content refinement; and a Library of Help Functions, a set of functions to facilitate the creation of online help systems under the architecture presented.

Palavras Chaves

Sistemas de Ajuda *Online*, Engenharia Semiótica, *Design* de IHC

INTRODUÇÃO

Cada vez mais as pessoas perdem o hábito de procurar no sistema de ajuda respostas para as dificuldades que encontram na utilização de uma aplicação computacional. Talvez uma das causas deste problema seja a forma como os sistemas de ajuda atuais apresentam este tipo de informação aos usuários.

Isto leva a sua elaboração ter um papel cada vez mais importante na área de desenvolvimento de aplicações,

VI Simpósio sobre Fatores Humanos em Sistemas Computacionais — Mediando e Transformando o Cotidiano. Hotel Bourbon, Curitiba, 17 a 20 de outubro de 2004. UFPR, CEIHC—SBC.

onde se necessitam criar sistemas de ajuda que apresentem não só uma interface mais amigável mas, principalmente, informações rápidas, objetivas e sucintas sobre o que é aplicação e como interagir com ela como um todo.

No mercado, já encontramos algumas ferramentas que auxiliam os *designers*¹ a elaborar o sistema de ajuda de suas aplicações. Dentre algumas destas podemos citar o Html Help Workshop®, o RoboHelp® e o HelpBreeze®. Tipicamente, essas ferramentas são utilizadas na criação de sistema de ajudas para os ambientes Windows e Web, e seguem um mesmo padrão de interface e navegação, o qual é reproduzido na maioria dos sistemas de ajuda existentes em aplicações disponíveis no mercado.

Mas será que estes padrões realmente satisfazem as necessidades dos usuários? Será que seu desenvolvimento tem buscado sanar as carências de quem utiliza os sistemas de ajuda ou apenas pretende acompanhar o nível tecnológico dos softwares?

Uma pesquisa realizada no ano de 2002, relativa à resposta de usuários a experiências frustrantes, revela dados preocupantes [2]. Numa situação de dificuldade, somente em último caso estes recorrem ao sistema de ajuda da aplicação (as três últimas tentativas que usuários fazem, nestes casos, são: consultar o manual, consultar o sistema de ajuda e reinicializar o programa). Percebe-se claramente, através desses resultados, que os sistemas de ajuda existentes não atendem as necessidades dos usuários, já que os mesmos, quando utilizam estes sistemas, o fazem praticamente como um último recurso.

A partir disto uma importante questão deve ser levada em consideração: Será que a contínua utilização desses padrões é a melhor opção para o usuário?

Pensando nisso, a proposta deste trabalho é auxiliar *designers* a criar o sistema de ajuda de suas aplicações, sob uma nova abordagem. Nesta abordagem [9], o *designer* poderá oferecer em sua aplicação, acesso ao sistema de ajuda por meio de diferentes expressões (por exemplo, “O que é isto?”, “E agora?” e “Para que serve isto?”), obtendo como resposta pequenas porções de conteúdo. Através destas expressões, possibilita-se, ao

¹ Neste trabalho é utilizado o termo *designer* tanto para *designers* de IHC quanto para desenvolvedores de aplicação.

usuário, um acesso a suas dúvidas de forma rápida e objetiva.

Além deste acesso diferenciado, o período em que o sistema de ajuda é construído também precisa ser discutido. O processo de criação do sistema de ajuda deve acompanhar o desenvolvimento da aplicação desde seu início até seu término, fazendo com que detalhes, que antes passavam despercebidos, passem a estar presentes, pois as informações necessárias para a construção da ajuda surgem em diferentes momentos da criação da aplicação.

Uma forma de capturar os dados necessários à construção da ajuda é buscá-los dos modelos de *design* de Interação Humano Computador (IHC) [10]. Estes modelos são usados durante o processo de *design* de IHC da aplicação para auxiliar a reflexão do *designer* sobre os materiais disponíveis para a sua construção e auxiliar no planejamento da mesma. Os modelos contemplam, tipicamente, informações sobre o domínio da aplicação, a aplicação em si, suas tarefas, seus usuários, seus distintos caminhos de interação e elementos que compõem a interface da mesma.

Neste contexto, o trabalho aqui proposto tem como objetivo fornecer ao *designer* ferramentas que o apoiem na construção de sistemas de ajuda *online* com base nos modelos de *design* de IHC. Este apoio será dado através de um editor de sistemas de ajuda (aqui tratado como editor de *help*), para auxílio aos *designers* na captura das informações relativas aos diferentes modelos de *design* de IHC utilizados e na geração prévia dos conteúdos de ajuda adequados à arquitetura proposta em [11].

Além de ajudar o *designer* a construir o conteúdo do sistema de ajuda durante o processo de desenvolvimento da aplicação, são oferecidas, também, ferramentas para que ele possa trabalhar este conteúdo e disponibilizá-lo em sua aplicação. Isto é concretizado através de uma biblioteca de funções de ajuda. Estas funções possuem o objetivo de permitir a ligação da aplicação (qualquer que seja ela) com o sistema de ajuda, exibir as expressões disponíveis para acesso ao mesmo (e suas respectivas respostas), bem como possibilitar a navegação entre o conteúdo (respostas) de ajuda, seguindo a proposta de ajuda em questão.

ABORDAGEM TEÓRICA UTILIZADA

Neste trabalho, é utilizada, como base teórica, a Engenharia Semiótica [3]. Na Engenharia Semiótica, a interface é vista como uma mensagem unidirecional enviada do *designer* ao usuário. Segundo esta abordagem, a mensagem é um artefato de metacomunicação, já que não apenas os *designers* se comunicam com os usuários, mas a própria interface troca mensagens com os usuários.

Neste contexto, a interface é um conjunto de signos, ou seja, representações que os *designers* usam para comunicar aos usuários como manipular o sistema que eles – *designers* – construíram para que os usuários possam atingir seus objetivos.

O processo de criação de uma interface é um processo de semiose ilimitada, isto é, uma geração indefinidamente

longa e variada, imprevisível, de significados associados entre si. Este processo significa que, por exemplo, a partir da definição do problema do usuário, o *designer* pode gerar diferentes soluções possíveis, e, sobre cada uma destas soluções, sempre podem ser imaginadas novas possibilidades de soluções e/ou operacionalizações das mesmas. Cada nova solução é capaz de gerar novas imaginações, novas possibilidades, novos *insights*, e assim sucessiva e indefinidamente. Este processo só é interrompido quando a pessoa que o está realizando não desejar ou não puder mais prosseguir com ele, seja por falta de necessidade, de tempo, de capacidade ou de interesse, por exemplo [9].

Assim como este é um processo de semiose ilimitada, a interpretação do produto desta semiose - a interface - também o é, do ponto de vista do usuário. E, de acordo com esta concepção, não é possível prever como cada usuário irá interpretar a interface criada pelo *designer*. E é aí que entra a importância do sistema de ajuda, no ponto de vista da Engenharia Semiótica.

Um sistema de ajuda que segue a abordagem da Engenharia Semiótica tem como principal objetivo explicitar a mensagem do *designer* ao usuário através da interface da aplicação [9]. E nenhuma ferramenta do sistema é mais adequada para passar esta mensagem que o sistema de ajuda, onde o *designer* pode, literalmente, “falar” ao usuário sobre a sua concepção, a aplicação. Para que aconteça esta comunicação *designer*-usuário, o sistema de ajuda proposto utiliza, para acesso, principalmente, expressões de comunicabilidade, que procuram representar a reação do usuário, no momento de dúvida, quanto ao uso de algum elemento da interface ou tarefa da aplicação [11]. [9] faz uma relação entre estas expressões e as falhas de comunicação *designer*-usuário associadas:

- **A quem isto afeta? De quem isto depende? Quem pode fazer isto?** O usuário quer saber se, executando determinada tarefa, a quem (determinados papéis de usuários) ele vai afetar ou de quem esta tarefa dependerá para sua realização. No outro caso (“Quem pode fazer isto?”), ele pode querer saber quais papéis estão habilitados à tarefa em questão.
- **Como faço isto?** O usuário não sabe como executar determinada tarefa.
- **E agora?** O usuário não sabe o que fazer como próximo passo na interação ou o usuário não sabe nem determinar a tarefa que necessita realizar.
- **Epa!** O usuário efetuou uma ação e/ou tarefa indesejada e quer desfazê-la.
- **Existe outra maneira de fazer isto?** O usuário deseja saber se existem outras possibilidades de caminhos que levem ao mesmo resultado.
- **O que aconteceu?** O usuário executa determinada ação (acreditando ser a correta para o que deseja realizar) e a resposta esperada não ocorre (ou ele não obtém a resposta desejada ou ele não obtém

resposta alguma). Ele não consegue entender o que aconteceu.

- **O que é isto?** O usuário não compreende determinado elemento encontrado na interface.
- **Onde está?** O usuário sabe o que quer fazer, mas não consegue encontrar o elemento correspondente na interface.
- **Onde eu estava?** O usuário quer saber “onde ele estava”, ou seja, qual a tarefa na qual ele estava trabalhando anteriormente. Ele quer saber seus passos anteriores para entender o estado em que está no momento.
- **Para que serve isto?** O usuário quer saber a utilidade da tarefa em questão.
- **Por que devo fazer isto?** O usuário quer saber por que deve fazer determinada tarefa.
- **Por que não funciona?** O usuário executa determinada ação que acredita ser a necessária no momento, e não obtém a resposta desejada. Ele tenta a mesma opção mais de uma vez, porque está convencido de estar fazendo a coisa certa.
- **Socorro!** O usuário quer um maior detalhamento dos dados de ajuda.

Ao usar estas expressões, o usuário pode explicitar mais diretamente, no momento da interação, a sua dúvida quanto ao uso da aplicação, obtendo uma resposta correspondente ao que foi solicitado. A Figura 1 exemplifica o uso de uma dessas expressões, num exemplo fictício com uso do Microsoft Word®, onde, ao ver uma série de marcas de revisão no texto, e não sabendo como trabalhar com as mesmas, o usuário acessa o sistema de ajuda através da expressão “E agora?”.

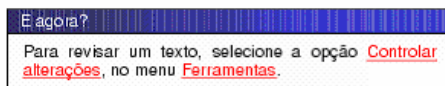


Figura 1: Resposta Minimalista.

Mas esta resposta pode não ser capaz de esclarecer por completo a sua dúvida. Por este motivo, o modelo de sistema de ajuda proposto permite ao usuário não apenas questionar o sistema quando necessita de algum auxílio, mas também oferecer a ele uma possibilidade de re-questionar este sistema a partir das respostas obtidas. A técnica de *layering* [4] é utilizada para viabilizar este re-questionamento conectando pequenas porções do conteúdo em uma abordagem minimalista [1]. A idéia é permitir ao usuário acessar estas porções de informação conforme as suas necessidades.

Com base nestes princípios, o sistema de ajuda proposto permite que o usuário acesse o conteúdo da ajuda através das expressões de comunicabilidade, obtendo uma pequena porção deste conteúdo como resposta. A partir desta resposta obtida, ele pode “re”questionar o sistema de ajuda, utilizando novas expressões para acessar informações sobre determinadas partes da resposta obtida

e assim sucessivamente, até que consiga obter as informações desejadas ou satisfazer sua curiosidade [11].

Seguindo o mesmo exemplo anterior (sobre revisão de texto), vemos o uso de recorrências de expressões (*layering*) a partir de uma resposta obtida (Figura 2).

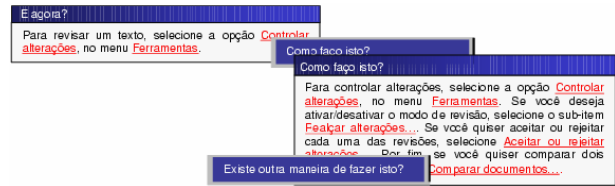


Figura 2: Respostas minimalistas e recorrência de expressões

PADRÕES E FERRAMENTAS PARA CONSTRUÇÃO DE SISTEMAS DE AJUDA ONLINE

O que existe atualmente para ajudar o *designer* a construir o sistema de ajuda de suas aplicações? Será que estas ferramentas permitem o uso da abordagem de ajuda aqui defendida?

A seguir serão apresentados alguns padrões de sistemas de ajuda bem como algumas ferramentas que tem o propósito de auxiliar os *designers* na criação destes sistemas. Após será feita uma análise das mesmas a fim de verificar se elas permitem a construção de um sistema de ajuda na forma descrita na seção anterior (minimalismo e *layering*).

Padrões Existentes

Atualmente, a maioria dos sistemas de ajuda encontrados no mercado se caracteriza por utilizar um ou outro de dois padrões bastante conhecidos no mundo todo: o WinHelp e HTML help. Devido ao fato de que esses sistemas vêm sendo utilizados em versões do sistema operacional Windows, sistema este bastante difundido atualmente, eles tornaram-se bastantes conhecidos e, por esse motivo, são utilizados pela maioria dos *designers*, levando ao “padrão” de sistemas de ajuda discutidos anteriormente.

WinHelp

WinHelp é o nome de um formato padrão de ajuda que teve origem com o Windows 3.0, sendo, também, utilizado em versões posteriores a esta.

Um sistema – utilizando este padrão - exibe o texto de ajuda, contido numa seção de tópico de ajuda, gerado a partir de um documento RTF. Arquivos com extensão .RTF armazenam o código fonte do sistema de ajuda e, quando compilados por um compilador de arquivos de ajuda, geram um arquivo no formato HLP, de propriedade da Microsoft. Um arquivo formato HLP tem por objetivo manter uma documentação interativa e *on-line* do software, para uma melhor operação do mesmo por parte do usuário.

Como estrutura de navegação, o WinHelp utiliza *links*, ou *jumps*, que são âncoras, geralmente exibidas de forma destacada no texto, que servem como desvio para outro tópico de ajuda. Quando selecionado, o *link* instrui o visualizador para que a exibição seja focada no tópico de ajuda referenciado por ele.

Outra característica peculiar ao WinHelp é sua barra de navegação, que permite ao usuário o percurso por todos os tópicos de ajuda da maneira que mais lhe convier, inclusive com a geração de um índice analítico montado a partir de referências de rodapé do arquivo RTF original (Figura 3).

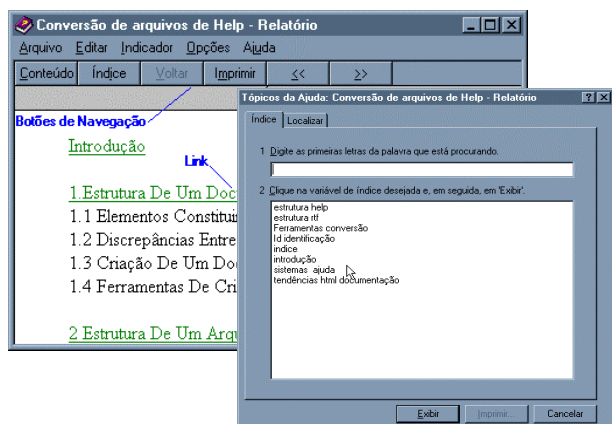


Figura 3: Tela principal do WinHelp 4.0 e janela do índice analítico

HTML Help

Atualmente, o padrão de ajuda utilizado na maioria das aplicações é o HTML Help. Esse formato se caracteriza por oferecer, através do sistema de ajuda, uma interface semelhante a páginas da Web.

O HTML Help é o formato para arquivos de *help* introduzido pela Microsoft com o Windows 98, onde arquivos de *help* são gerados a partir de formato html.

A interface – em sistemas que usam este padrão - apresenta duas áreas principais (Figura 4). A primeira é onde o usuário terá a possibilidade de efetuar pesquisas, listar os tópicos da ajuda, relacionar em favoritos os tópicos de maior interesse e visualizar os índices do conteúdo da ajuda. A outra área é onde a informação é mostrada ao usuário. Nessa área, o usuário dispõe da informação solicitada e de *links* que serão usados para acessar áreas que estão relacionadas com o tipo de informação exibida.

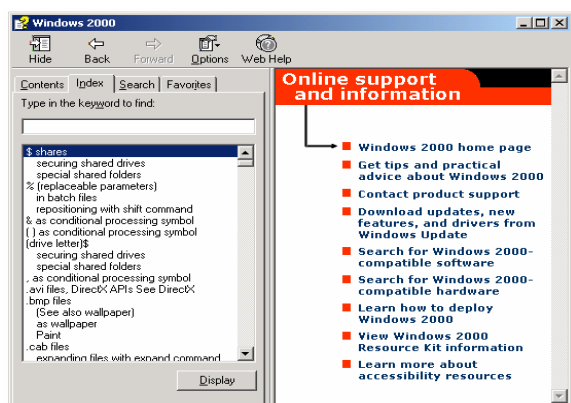


Figura 4: Interface do HTML Help.

Ferramentas Existentes

As ferramentas a serem apresentadas têm o propósito específico de auxiliar os *designers* na criação de sistemas

de ajuda a serem utilizados por suas aplicações. Estas ferramentas geralmente seguem um dos padrões mencionados na seção anterior, com algumas poucas inovações introduzidas em suas versões. A seguir serão apresentadas algumas das ferramentas mais utilizadas pelos *designers* hoje em dia.

HTML Help Workshop®

O HTML Help Workshop® [6] é uma ferramenta que permite elaborar sistemas de ajudas no formato padrão html, o qual é utilizado na maioria das aplicações atualmente.

Entre algumas opções desta ferramenta, está a possibilidade de configurar a apresentação da janela de ajuda, podendo-se incluir ou remover tópicos, adicionar categorias, definir os botões a serem utilizados e escolher o visual da janela e a navegação dos tópicos. Depois de definidas as configurações da apresentação das janelas, os índices, as categorias, os tópicos e o conteúdo dos itens, deve-se utilizar um compilador para gerar um único arquivo de ajuda e, conseqüentemente, gerar o sistema de ajuda da aplicação (Figura 5).

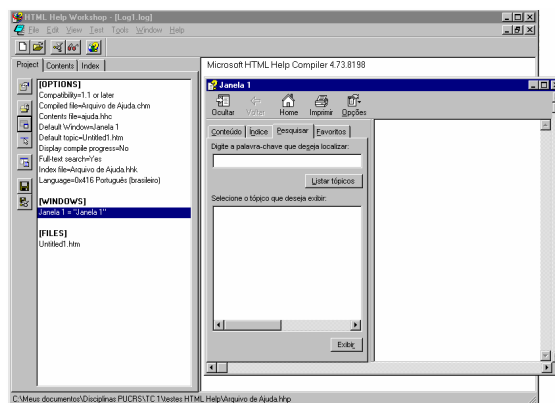


Figura 5: Configurações do projeto e visualização da janela de ajuda concluída

Podemos citar como características do HTML Help Workshop®:

- permite converter um projeto existente elaborado no padrão WinHelp em um projeto padrão HTML Help;
- oferece suporte para uma tabela interativa de sumários, índice de palavras chaves e pesquisa completa de texto;
- utiliza um novo formato de arquivo .CHM, que armazena todos os arquivos que compõem o projeto do sistema de ajuda.

HelpBreeze®

A tela principal da ferramenta para construção de ajuda HelpBreeze® [5] segue o mesmo padrão utilizado no sistema Microsoft Windows®, ou seja, é composta pelas abas tradicionais (conteúdo, índice e pesquisa) e pela janela onde é visualizada a informação de ajuda.

Um importante recurso disponível no padrão HTML Help é a inclusão de *links* de navegação para que o usuário

possa aprofundar-se num assunto de acordo com o seu grau de interesse. A Figura 6 procura demonstrar como seria este recurso de navegação, criado nesta ferramenta.

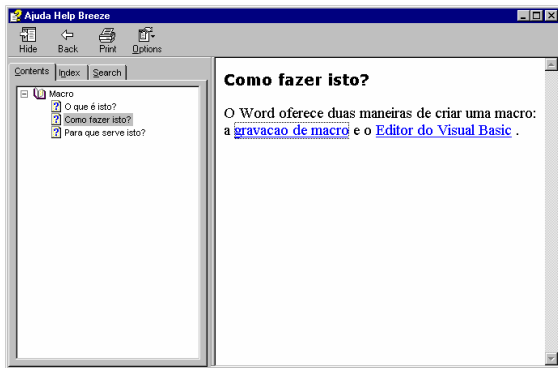


Figura 6: Inclusão de links no Helpbreeze

Quando o usuário demonstra interesse em obter a descrição de uma frase ou de uma expressão inserido no conteúdo de ajuda, apresenta-se, através de uma janela *popup*, a informação solicitada pelo usuário. A Figura 7 ilustra este recurso.

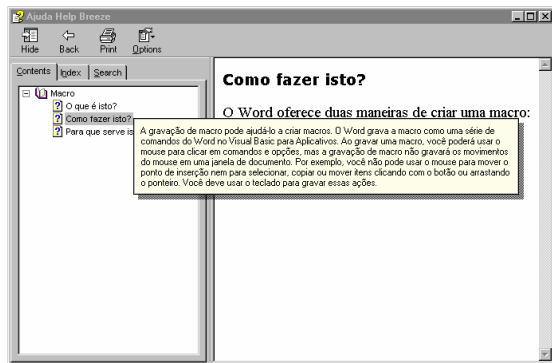


Figura 7: Janelas *popup* no Helpbreeze

O HelpBreeze® armazena as informações e definições de interface dos tópicos em vários arquivos html, de modo que, para cada tópico, exista um arquivo html correspondente. Portanto o usuário poderá clicar num determinado tópico e em seguida clicar na aba HTML Source para poder visualizar a implementação de determinado tópico na linguagem HTML.

Outra funcionalidade oferecida pelo HelpBreeze® é que o usuário pode ter uma visualização prévia de como vai ser apresentada a interface sem a necessidade de haver a compilação dos arquivos fontes do projeto.

Algumas características desta ferramenta são:

- possibilidade de testar características como navegação e janelas *popup* sem a necessidade de compilar o projeto;
- possibilidade de criar sistemas de ajuda no formato padrão WinHelp;
- possibilidade de criar sistemas de ajuda no formato padrão Microsoft Html Help;

- suporte ao Microsoft Word® 95, 97, 2000 e 2002 (XP).

RoboHelp®

O RoboHelp® [8] é uma ferramenta que tem, como público alvo, projetistas, *designers*, autores e escritores técnicos de sistemas de ajuda. Esta ferramenta permite que o usuário escolha um dos possíveis formatos de sistema de ajuda que são suportados por este software, tais como WebHelp², HtmlHelp, WinHelp, FlashHelp³.

Escolhendo como exemplo o formato WinHelp surgem duas janelas. A primeira (Figura 8) apresenta um gerenciador de diretórios e arquivos responsáveis por organizar o conteúdo do projeto (a figura exhibe alguns tópicos que foram criados para compor um sistema de ajuda de exemplo).

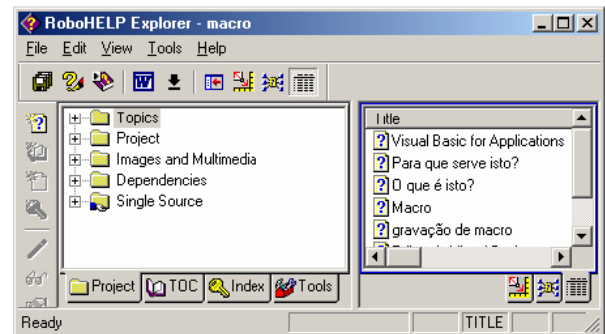


Figura 8: Gerenciador de arquivos.

Já a segunda (Figura 9) é uma página do programa Microsoft Word® que serve como uma janela de edição do projeto onde o usuário pode fazer uso das ferramentas disponíveis no programa.



Figura 9: Janela de edição.

Podem ser relacionadas algumas características desta ferramenta:

- permite a criação de sistemas de ajuda para aplicações *desktop* (Windows, Mac, Linux) e aplicações baseadas em Web;

² WebHelp: padrão de ajuda que se apresenta em forma de site da web, sendo utilizado para o acesso às informações de sistemas disponíveis via Internet e Intranets.

³ FlashHelp: formato de ajuda incorporado dentro do RoboHelp® para assistência ao usuário, suportado em múltiplas plataformas e *browsers*.

- possibilita adicionar a um projeto de ajuda, recursos padronizados e avançados como uma tabela de conteúdo, índice, glossário, gráficos, efeitos especiais e outros;
- possibilita desenvolver sistemas de ajuda com um editor HTML integrado, do tipo WYSIWYG, um Microsoft Word® ou qualquer outro editor HTML comercial (Dreamweaver, FrontPage, HomeSite, etc);
- possui uma interface intuitiva dando ao usuário a capacidade de criar tópicos de ajuda e *hiperlinks* com a simplicidade do recurso arrastar e soltar.

Aplicabilidade de Uso das Ferramentas Analisadas na Construção de Sistemas de Ajuda sob a Abordagem Utilizada

A partir do estudo realizado sobre as ferramentas de auxílio, a fim de verificar-se a possibilidade de implementar o modelo de ajuda proposto em (Silveira, 2002) nas ferramentas citadas no capítulo anterior e as dificuldades e/ou facilidades em fazê-lo, foram criados exemplos de ajuda – sob esta perspectiva – em diferentes ferramentas. A ferramenta Macro, do editor de texto Microsoft Word®, foi utilizada como fonte para o conteúdo da ajuda.

Sistema de Ajuda no HTML Help WorkShop®

Esta ferramenta permite apenas a criação de janelas seguindo um mesmo padrão, ou seja, apresentação da ajuda em janela a parte do sistema. O conteúdo minimalista pode ser criado, mas a navegação pelo mesmo somente pode ser feita pelo índice lateral, ou pelos *links* no texto, que recarregam a mesma janela, com novos itens no índice lateral (Figura 10).

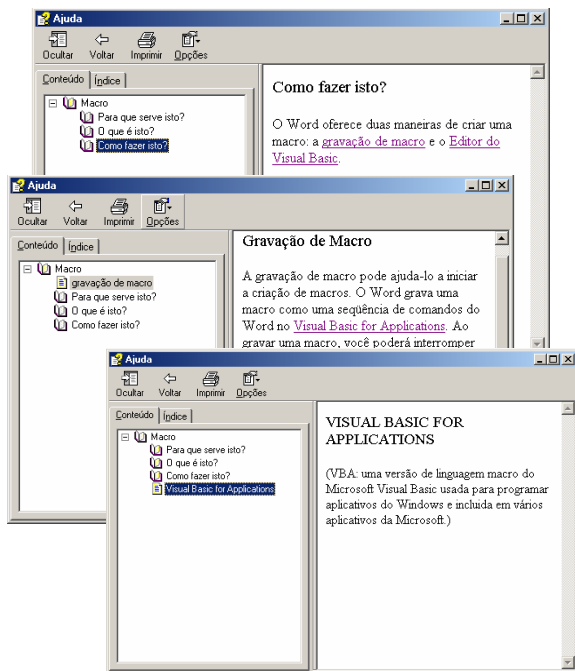


Figura 10: Acesso ao conteúdo de ajuda e navegação no HTML Help Workshop

Sistema de Ajuda no HelpBreeze®

Esta ferramenta também permite apenas a criação de janelas seguindo um mesmo padrão, ou seja, apresentação da ajuda em janela a parte do sistema. O conteúdo minimalista pode ser criado, mas a navegação pelo mesmo somente pode ser feita pelo índice lateral, ou pelos *links* no texto, que ou recarregam a mesma janela, com novos itens no índice lateral ou apresentam uma janela popup com a explicação do *link* (Figura 11).

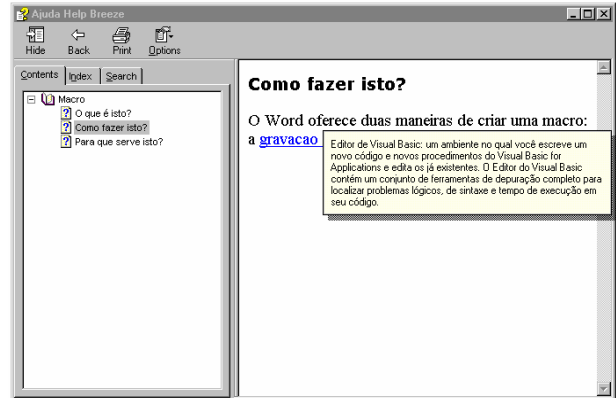


Figura 11: Acesso ao conteúdo de ajuda e navegação no HelpBreeze

Sistema de Ajuda no RoboHelp®

Esta ferramenta é a que chega mais próxima ao formato sugerido para sistemas de ajuda *online*. Ela também permite apenas a criação de janelas seguindo um mesmo padrão, ou seja, apresentação da ajuda em janela a parte do sistema. Permite, também, a criação do conteúdo minimalista pode ser criado, mas possibilita a navegação através de *links* (Figura 11).

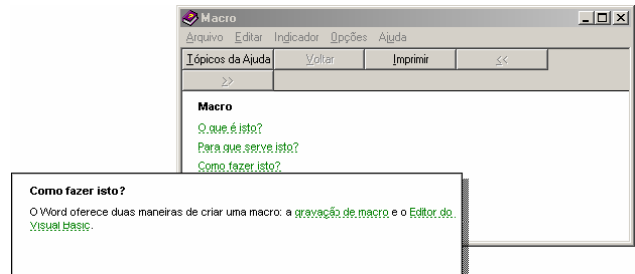


Figura 12: Tópico “Como fazer isto?” disponibilizando links de acesso

Ao acessar o *link* gravação de macro surge outra janela (Figura 13) exibindo o conteúdo requerido no qual observa-se um outro *link* de acesso, possibilitando mais um nível de navegação. É importante salientar que a janela *popup* Como fazer isto? desapareceu.

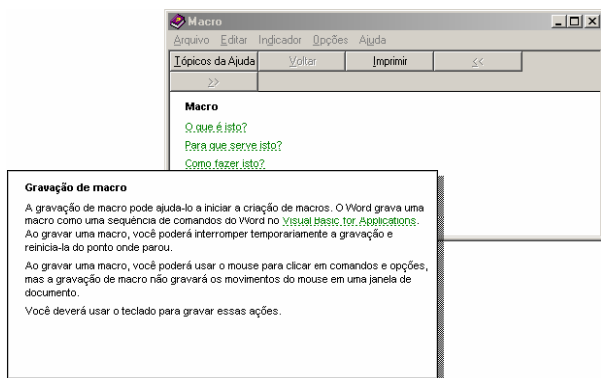


Figura 13: Acesso ao conteúdo gravação de macro.

Análise da Implementação do Sistema Proposto nas Ferramentas Analisadas

O uso das ferramentas para criar o sistema de ajuda proposto, possibilitou atribuir um grau de dificuldade para cada funcionalidade das ferramentas (Tabela 1).

Tabela 1: Funcionalidades das ferramentas.

Funcionalidades	HTML Help Workshop	HelpBreeze	Robo Help
Criação dos tópicos de ajuda	Fácil	Fácil	Fácil
Criação de janelas <i>popup</i>	Médio	Fácil	Fácil
Uso de <i>links</i> para navegação entre os diversos níveis de informação	Difícil	Fácil	Médio
Inserção o conteúdo da ajuda nas janelas de visualização	Difícil	Fácil	Fácil
Criação de índices	Médio	Médio	Médio
Pesquisa por palavras-chaves	Médio	Médio	Médio

Além disto, foi possível comparar estas ferramentas a partir das facilidades (ou não) de implementação da proposta em cada ferramenta (Tabela 2).

Tabela 2: Facilidades das ferramentas.

Facilidades	HTML Help Workshop	Help Breeze	Robo Help
Capacidade de criar <i>links</i> de navegação entre janelas <i>popup</i>	x		x
Possibilidade de implementar o sistema de ajuda dentro da aplicação			
Adaptação à sintaxe dos <i>scripts</i> que armazenam o conteúdo de ajuda			
Visualização de mais de um nível de janela <i>popup</i>			
Possibilidade de desenvolver o sistema de ajuda durante o processo de criação da aplicação	x	x	x
Capacidade de gerar uma janela simples, contendo apenas as expressões sem barras de menus e ferramentas			

FERRAMENTAS PROPOSTAS

Através do estudo apresentado na seção anterior, verificou-se que as ferramentas tradicionais utilizadas para construção de sistemas de ajuda *online* não possuem algumas das funcionalidades que seriam fundamentais para permitir e/ou facilitar o uso da arquitetura de ajuda aqui trabalhada [90] tanto pelos *designers* de aplicações como pelos usuários finais. Dentre estas funcionalidades estão, por exemplo:

- permitir o vínculo entre elementos da aplicação e janelas de ajuda *popup*, ou seja, oferecer ao *designer* a possibilidade de associar aos elementos de sua aplicação (qualquer que seja esta aplicação) um acesso direto ao sistema de ajuda criado por meio destas ferramentas. A Figura 14 mostra um *link* de acesso ao sistema de ajuda associado ao elemento Seção.

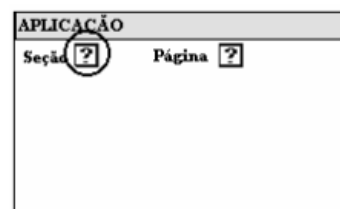


Figura 14: Acesso ao sistema de ajuda vinculado a um elemento da aplicação

- possibilitar que, a cada *link* acessado, novas expressões possam ser disponibilizadas. A Figura 15 ilustra esta característica, pois, ao elemento avisos, estão disponíveis as expressões “O que é isto?” e “Para que serve isto?”.

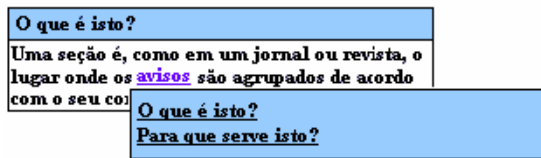


Figura 15: Exibição de novas expressões sobre um mesmo tema

- manter a visualização de diversos níveis de janelas de ajuda *popup*, isto é, a cada acesso – recorrente – ao sistema de ajuda, exibir uma nova janela sem que a anterior desapareça. Desta forma, o usuário estará ciente do caminho que ele percorreu até o momento permitindo ainda que ele possa voltar a uma janela previamente aberta para realizar novas consultas. Tem-se um exemplo disso através da Figura 16 que mostra a janela referente à expressão “O que é isto?” permanecendo aberta mesmo após o acesso ao link [Para que serve isto?](#) levando à exibição da janela correspondente a sua resposta.

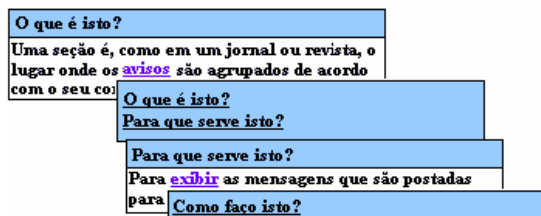


Figura 16: Visualização dos níveis de janela *popup*

Visando auxiliar o *designer* a construir um sistema de ajuda que possa oferecer estas características, visto que as ferramentas tradicionais não o fazem, este trabalho apresenta um conjunto de ferramentas para dar suporte à arquitetura proposta em [9], possibilitando seu uso em aplicações que são executadas tanto em ambiente Windows quanto em ambiente Web. Dentre estas ferramentas, encontram-se o **Editor de Help**, o **Editor do Conteúdo da Ajuda** e a **Biblioteca de Funções de Ajuda**, os quais serão descritos em maior detalhe a seguir.

Editor de Help

Segundo [7], o interesse pelo uso de modelos de *design* de IHC cresceu graças ao elevado potencial para produção de ambientes integrados de desenvolvimento e a possibilidade de suporte a todas as fases do projeto. Estes modelos são utilizados – principalmente - para apoiar a reflexão e a tomada de decisão sobre o *design* de uma aplicação. E, assim como existem diferentes formas de realizar as tomadas de decisão, existem, também, diferentes modelos para apoiá-las, dentre os quais encontram-se modelos de domínio, de aplicação, de apresentação, de interação, de interfaces e de usuário, dentre outros.

Além de utilizar estes modelos para apoio para a tomada de decisão, em [10] foi proposta a extensão de um conjunto de modelos tradicionais de *design* de IHC (modelo de domínio, de aplicação, de tarefas, de usuário, de interação e de interface), para serem utilizados, como

fonte de informação para a construção do conteúdo do sistema de ajuda de uma aplicação.

Em [10], defende-se que estas informações devem ser capturadas durante o processo de *design* da aplicação, mais precisamente no momento em que são geradas, evitando a perda de informações se deixadas para serem obtidas no final do processo de criação.

O **editor de help** tem como base os modelos de *design* de IHC, conforme proposto em [10]. A idéia é fornecer ao *designer* uma ferramenta de apoio à construção do sistema de ajuda, mas esta ferramenta poderá, também, auxiliá-lo durante a modelagem de IHC da aplicação, na medida em que ele poderá estruturar todo o processo com uso do editor.

O *designer* poderá utilizar o editor durante todo o processo de *design*, incluindo – a cada nova etapa – as informações por ele trabalhadas e, ao final da modelagem (e respectiva inclusão de seus dados no editor) poderá gerar automaticamente o conteúdo da ajuda, de acordo com a arquitetura descrita anteriormente.

A fim de não só capturar as informações pertinentes a cada modelo de *design*, mas, também, de deixá-las disponíveis ao usuário durante todo o processo de *design*, as mesmas estão estruturadas em um banco de dados.

Para isto foram analisadas as informações relativas a cada modelo e projetado um banco de dados que refletisse as mesmas bem como seus relacionamentos. Além da base de dados foi implementada uma ferramenta para manipulação dos dados da base (Figura 17).

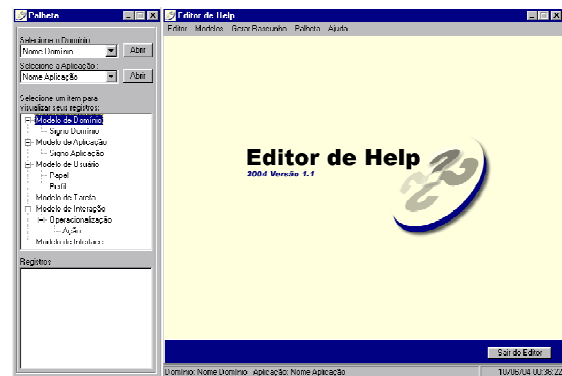
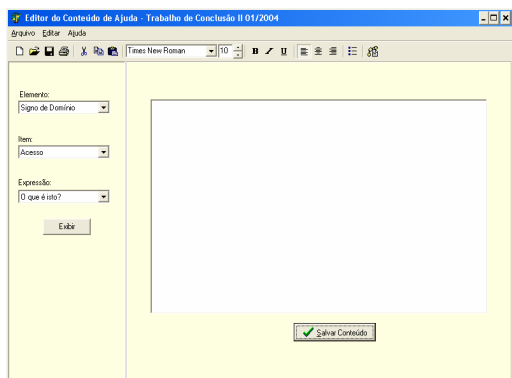


Figura 17: Editor de Help.

No momento em que o *designer* se sentir satisfeito quanto às informações fornecidas aos modelos, será possível fazer a geração de um rascunho de conteúdo de ajuda.

Editor do Conteúdo da Ajuda

O conteúdo de ajuda será exibido ao usuário sempre de forma minimalista, ou seja, janelas *popup* surgirão na interface da aplicação com pequenas porções de informações. Mas mesmo que estas informações sejam apresentadas compactamente, quanto melhor for sua aparência e a disposição de seu conteúdo, melhor o *designer* transmitirá sua mensagem e melhor será o entendimento do usuário.



Por este motivo, foi desenvolvido este segundo editor (Figura 18). Ele permite ao *designer* organizar e estruturar as informações de ajuda – rascunho gerado pelo **Editor de Help** - da maneira mais clara possível, facilitando assim, a compreensão do usuário.

Figura 18: Editor do Conteúdo da Ajuda.

Algumas funcionalidades deste editor:

- criação e remoção de links para a navegação entre o conteúdo de ajuda (*layering*);
- formatação de quebras de linhas e tabulação;
- modificação do estilo da fonte em negrito e/ou itálico;
- modificação do tamanho da fonte;
- inserção de marcadores.

Biblioteca de Funções de Ajuda

Tendo-se o conteúdo da ajuda gerado através do **Editor de Help**, e refinado através do **Editor de Conteúdo da Ajuda**, é possível fazer a chamada a estes conteúdos a partir da aplicação desejada. Para isto foram criadas uma série de funções de ajuda, as quais compõem a biblioteca aqui descrita.

As funções que irão compor a biblioteca podem ser descritas, em termos gerais, da seguinte forma:

- **função ativa-ajuda:** Função responsável por fazer a ligação entre a aplicação criada pelo *designer* (qualquer que seja ela) e o sistema de ajuda. Das funções componentes desta biblioteca proposta, esta é a única que o *designer* da aplicação necessita conhecer;
- **função busca-expressões:** Esta função tem o propósito de buscar as expressões que estão disponíveis para um determinado elemento da aplicação;
- **função apresenta-expressões:** Esta função tem a finalidade de mostrar, em uma janela *popup*, as expressões que foram buscadas pela função busca-expressões;
- **função busca-resposta:** Esta função tem o objetivo de buscar a resposta correspondente à expressão escolhida;

- **função apresenta-resposta:** Esta função tem o propósito de exibir a resposta buscada pela **função busca-resposta** em uma janela *popup* juntamente com os possíveis *links* de recorrência (novas chamadas à função **busca-expressões**).

Método para Construção de Sistemas de Ajuda Online

Em [12], foi apresentado um método para construção de sistemas de ajuda *online*, baseado na Engenharia Semiótica. Na figura a seguir (Figura 19), estão representados as etapas deste método e onde – no mesmo – as ferramentas acima descrita se encaixam.

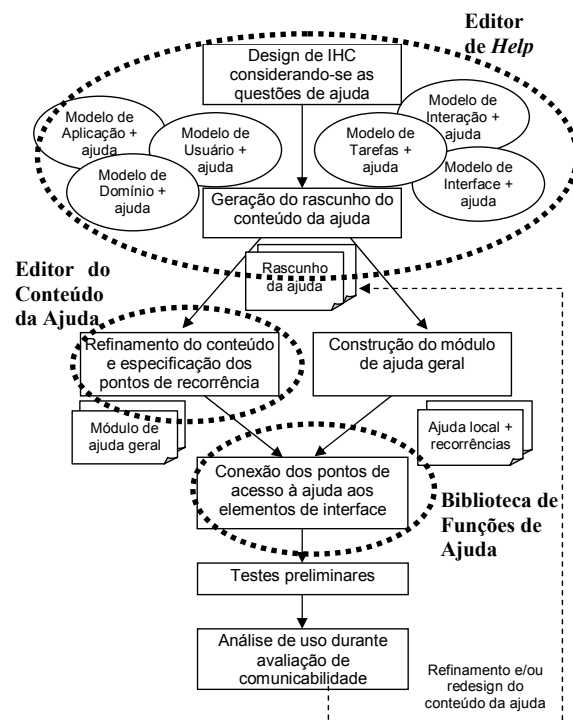


Figura 19: O método [12] e sua relação às ferramentas propostas.

CONSIDERAÇÕES FINAIS

Cada vez mais, os sistemas de ajuda existentes estão se tornando o último recurso de ajuda ao qual o usuário recorre ao se defrontar com um problema ou dúvida durante a interação com uma aplicação.

Um dos motivos dessa resistência deve-se a forma como estes sistemas são elaborados, o que remete a maneira como as ferramentas tradicionais para sua construção estão modeladas e a forma como estas exibem as informações nelas inseridas.

Inicialmente, foi feito um estudo sobre os padrões e ferramentas para construção de sistemas de ajuda existentes, onde foi possível constatar a dificuldade de elaborarem-se sistemas de ajuda mais flexíveis, alternativos aos padrões existentes atualmente.

Constatou-se que as ferramentas mais tradicionalmente utilizadas seguem todas um processo de interação semelhante entre o usuário e a aplicação, ou seja, o acesso à informação de ajuda se dá através da utilização de pesquisa por palavra-chave, índices e conteúdo. Também

não proporcionam ao *designer* da aplicação uma técnica para exibir o conteúdo de forma mais direcionada a dúvida do usuário, sobrecarregando-o com uma grande quantidade de informações. Além disso, dificultam elaborar o sistema de ajuda paralelamente com o desenvolvimento da aplicação.

Para dar aos *designers* (e aos usuários de suas aplicações) uma alternativa a este padrão pré-estabelecido de ajuda, este trabalho propôs o uso de um conjunto de ferramentas que facilitam a construção de ajuda segundo a abordagem proposta em [90].

A idéia de uso destas ferramentas é facilitar o trabalho do *designer*, na construção do sistema de ajuda de suas aplicações, tentando suprir, desta forma, as necessidades do usuário em relação às ferramentas tradicionais, estimulando o uso do sistema de ajuda e auxiliando os usuários a esclarecerem suas dúvidas, navegarem pelas informações obtidas e obterem um maior conhecimento de toda a funcionalidade da aplicação.

REFERENCES

1. Carroll, J.C. (ed.) (1998) *Minimalism Beyond the Nurnberg Funnel*. Cambridge, The MIT Press, Cambridge.
2. Ceaparu, I., Lazar, J., Bessiere, K., Robinson, J., Shneiderman, B. (2002) Determining Causes and Severity of End-User Frustration. *Technical Report, HCIL-2002-11, CS-TR-4371, UMIACS-TR-2002-51* [<ftp://ftp.cs.umd.edu/pub/hcil/Reports-Abstracts-Bibliography/2002-11html/2002-11.pdf>, visited September 2002]
3. de Souza, C.S. (1993) The Semiotic Engineering of User Interface Languages. *International Journal of Man-Machine Studies*, 39, 753-773.
4. Farkas, D.K. (1998) Layering as a Safety Net for Minimalist Documentation. Carroll, J.C. (ed.) *Minimalism Beyond the Nurnberg Funnel*. Cambridge, The MIT Press, Cambridge.
5. HelpBreeze®, disponível em http://www.lemon.com.br/collection/browser_detail.cf mvisitado de Agosto a Outubro de 2003.
6. Html Help Workshop®, disponível em <http://msdn.microsoft.com/library/default.asp?url=/library/en-us/htmlhelp/html/hwMicrosoftHTMLHelpDownloads.asp>, visitado de Agosto a Outubro de 2003.
7. Puerta, A. (1996) The Mecano Project: Comprehensive and Integrated Support for Model-Based Interface Development. Vanderdonck, J. (ed) *Computer-Aided Design of User Interfaces*. Namur, Presses Universitaires de Namur.
8. RoboHelp®, disponível em <http://www.ehelp.com/> visitado de Agosto a Outubro de 2003.
9. Silveira, M.S. Metacomunicação Designer-Usuário na Interação Humano-Computador: design e construção do sistema de ajuda, 2002. *Tese de Doutorado*. Disponível em http://www.inf.pucrs.br/~milene/pessoal/download_tese.htm
10. Silveira, M.S., Barbosa, S. D. J., de Souza, C.S. Design de Sistemas de Ajuda *Online* baseados em Modelos. *Anais do Simpósio Brasileiro de Fatores Humanos em Sistemas Computacionais – IHC2002*. 2002.
11. Silveira, M.S., Barbosa, S.D.J.; de Souza, C.S. (2001) Augmenting the Affordance of *Online* Help Content. *Proceedings of IHM-HCI 2001*, Lille, Springer-Verlag.
12. Silveira, M.S., de Souza, C.S., Barbosa, S.D.J. Um Método da Engenharia Semiótica para a Construção de Sistemas de Ajuda Online. *Anais do Congresso Latinoamericano de Interação Humano-Computador – CLIHC 2003*. 2003.

Inserindo IHC em Empresas Brasileiras de Informática: Relato, Discussão e Lições Aprendidas

Antônio Marcos Endler

PROCERGS (Companhia de Processamento de
Dados do Estado do Rio Grande do Sul)
Praça Açorianos S/N – Porto Alegre - RS
antonio-endler@procergs.rs.gov.br

Marcelo Soares Pimenta

Instituto de Informática – UFRGS
Caixa Postal 15.064 – 91.501-970
Porto Alegre – RS – Brazil
mpimenta@inf.ufrgs.br

RESUMO

A integração de IHC (Interação Humano-Computador) com a prática de desenvolvimento de software em uma organização é um problema ao mesmo tempo técnico e gerencial e ainda há pouco material disponível na literatura nacional sobre este tema. Neste artigo são descritas e discutidas algumas das atividades de sensibilização e capacitação de equipes, modificação de processo de desenvolvimento e inserção de princípios, métodos e padrões em IHC, reunidas a partir de experiências e práticas das equipes de desenvolvimento de uma empresa pública de informática. Essas experiências foram convertidas em dicas e recomendações que podem servir de subsídio para outras instituições com objetivos similares.

ABSTRACT

The integration of HCI (Human Computer Interaction) aspects in actual business software development practices is a very interesting problem, including both technical and management topics. Particularly, as interactive systems became more and more sophisticated, we identified the need for an approach to effectively integrate concepts, techniques, methods and practices from software engineering and HCI in order to develop systems considering at same time quality criteria of these two domains. However, surprisingly there is not enough published work reporting experiences in companies and discussing results, lessons and problems. This article relates some integration activities accomplished day-by-day in a public software development company. In addition, it presents some lessons, difficulties and recommendations we consider useful as a discussion basis or aid for institutions with similar goals.

Palavras chaves

Integração entre IHC e Engenharia de Software, desenvolvimento de sistemas interativos, relatos de experiências.

VI Simpósio sobre Fatores Humanos em Sistemas Computacionais — Mediando e Transformando o Cotidiano. Hotel Bourbon, Curitiba, 17 a 20 de outubro de 2004. UFPR, CEIHC—SBC.

INTRODUÇÃO

Como é possível introduzir conceitos, métodos e técnicas de IHC (Interação Humano-Computador) no processo de desenvolvimento de software em uma empresa de informática de grande porte? Quais mudanças são necessárias no processo de desenvolvimento de software para introduzir esses conceitos, métodos e técnicas? Que atividades precisam ser realizadas para sensibilização e capacitação de equipes? Buscando respostas para estas perguntas, desde 1996 a PROCERGS – Companhia de Processamento de Dados do Estado do Rio Grande do Sul realiza esforços no sentido de consolidar uma cultura de desenvolvimento de sistemas que privilegie os aspectos de IHC. Com este objetivo, atuou-se em três áreas estratégicas:

- capacitação dos técnicos da empresa;
- mudança nos processos de desenvolvimento das interfaces gráficas dos sistemas; e
- introdução de (novos) métodos, técnicas e ferramentas para dar suporte às mudanças nos processos.

Neste período, a experiência da empresa acumulou-se através do emprego de ações vinculadas às 3 estratégias mencionadas. Isso produziu conhecimentos práticos a respeito da aplicação dos conceitos, métodos e técnicas em IHC que serão apresentados neste trabalho na forma de recomendações. Elas podem auxiliar aqueles desenvolvedores de sistemas de informação que pensam em introduzir um novo método de IHC, alterar os procedimentos de projeto, introduzir novos papéis no processo ou mesmo qualificar os técnicos participantes dos projetos. Por isso, este artigo detalha na segunda seção algumas das ações realizadas ao longo dos últimos anos, destacando entre elas:

- composição de equipes multidisciplinares em projetos de interface;
- desenvolvimento de padrões de interface com ênfase na usabilidade;
- desenvolvimento de métodos e ferramentas de prototipagem;
- desenvolvimento ergonômico de um sistema com grande número de usuários;

- uso de métodos de avaliação de interfaces.

Na terceira seção é proporcionada uma discussão sobre as principais ações realizadas, onde é apresentada uma lista de recomendações para os desenvolvedores de sistemas. A lista pode ser bastante útil pois apresenta aprendizados práticos para aqueles que desejam implantar ou estejam em processo de implantação de IHC em suas empresas, instituições ou mesmo laboratórios acadêmicos que prestem serviços à comunidade. Na quarta seção são expostas as conclusões tiradas das experiências ao longo do período em que a empresa vem tentando consolidar a IHC no desenvolvimento dos sistemas.

IHC NO DIA-A-DIA DE UMA EMPRESA DE INFORMÁTICA

Até o final dos anos 80, a informática esteve restrita aos grandes Centros de Processamento de Dados (CPDs), num ambiente dominado por poucos e grandes fornecedores de *software* e *hardware*, onde predominavam as interfaces orientadas a caractere. Com o crescimento do ambiente Windows, a partir dos anos 90, observou-se o aumento da demanda por projetos de sistemas com interfaces gráficas. Linguagens de programação como o *Visual Basic* e o *Delphi* deram suporte à criação de sistemas com interfaces gráficas mais elaboradas. Em agosto de 1993 foi apresentado o primeiro aplicativo multimídia (com recursos de áudio, animação e vídeo) desenvolvido inteiramente no Estado do Rio Grande do Sul para a Expoiner (Exposição Internacional Agropecuária). Em outubro de 1995, iniciou a operação da Internet Comercial do RS, através do provedor de acesso Via RS. A Internet, a partir deste momento, passou a ser um importante agente impulsionador da demanda por projetos de interface gráfica. Os sites passaram a oferecer serviços através de sistemas que começaram a

ser utilizados mais frequentemente pelos cidadãos, ocasionando maior preocupação em relação a facilidade de uso desses serviços.

Em buscas realizadas nos anais de eventos como o IHC e em revistas setoriais pode-se perceber que no Brasil ainda são poucas as publicações existentes que relatam casos práticos sobre a introdução dos conceitos de IHC no processo de desenvolvimento das empresas de informática. Academicamente, a IHC ganhou impulso nos anos 90 com o surgimento de eventos específicos sobre o assunto que incentivaram a publicação de pesquisas e reflexões. Foi neste período que surgiram centros de pesquisa como LabiUtil/UFSC, SERG, COMUNIHC e que as primeiras disciplinas de IHC foram ministradas em cursos de graduação e pós-graduação.

Novos profissionais oriundos de outras áreas, como os designers gráficos, os publicitários, os jornalistas e outros, passaram a participar dos projetos junto com analistas de sistemas, demandando métodos que integrassem diferentes perfis, papéis e pontos-de-vista ao processo de desenvolvimento de sistemas.

O cenário descrito acima retrata o ambiente vivenciado pelas empresas e instituições nos últimos anos e também pela PROCERGS, sendo que hoje a empresa vem se destacando na integração de equipes multidisciplinares, no uso de muitos dos métodos e ferramentas surgidos no decorrer deste período e nos investimentos em capacitação de seu pessoal através da realização de cursos especialmente adaptados as suas necessidades. Na Tabela 1 é possível observar as ações concretizadas que ajudaram no desenvolvimento de uma cultura de IHC na PROCERGS.

Tabela 1. Atividades desenvolvidas em IHC na PROCERGS

Tipo de Ação	Descrição da Ação	Instituição Responsável	Período
Formação/Capacitação	Formação de um grupo interno de interesse sobre a área de IHC de auto-aprendizagem.	PROCERGS	1995
Formação/Capacitação	Curso sobre Ergonomia de software ministrado por Walter de Abreu Cybis (UFSC) em Porto Alegre para os técnicos da PROCERGS.	LabiUtil/UFSC U	1996
Melhoria no processo de desenvolvimento	Desenvolvimento do Padrão PROCERGS de interfaces gráficas Windows.	PROCERGS	1996
Formação/Capacitação	Realização do Seminário de Interfaces Gráficas com palestras, debates e apresentações de vídeos com depoimentos de técnicos da PROCERGS sobre a importância das interfaces gráficas para os clientes.	PROCERGS	1997
Formação/Capacitação	Distribuição de publicação interna com 4 artigos escritos por funcionários da própria empresa.	PROCERGS	1997
Aplicação de métodos e técnicas em IHC	Desenvolvimento do sistema PROCERGS Escola de acordo com as diretrizes contidas no padrão PROCERGS de interface gráfica Windows.	PROCERGS	1998
Aplicação de métodos e técnicas em IHC	Avaliação ergonômica do software PROCERGS Escola.	LabiUtil/UFSC	1998
Melhoria no processo de desenvolvimento	Elaboração do manual de desenvolvimento de ícones para sistemas que utilizam o padrão Windows.	Curso de Desenho Industrial –UFSM Universidade Federal de Santa Maria	1998
Formação/Capacitação	Curso de desenvolvimento de ícones para os designers da PROCERGS.	Curso de Desenho Industrial –UFSM	1998
Formação/Capacitação	Curso de Ergonomia de interfaces com exercícios práticos, ministrado pela professora Ana Maria de Moraes (PUC-RJ) para os funcionários da PROCERGS e para os mestrandos da Engenharia de Produção/UFRGS(Universidade Federal do Rio Grande do Sul).	PUCRJ Pontifícia Universidade Católica do Rio de Janeiro.	1998
Formação/Capacitação	Incentivo aos funcionários para realização de mestrado e de participação como alunos especiais ou ouvintes nas disciplinas na área de IHC.	PROCERGS/UFRGS	1999/2000

Aplicação de métodos e técnicas em IHC	Desenvolvimento da interface gráfica do software Direto de acordo com o método do Design Macroergonômico (Fogliatto & Guimarães, 1999) envolvendo a participação dos usuários e especialistas durante o processo de concepção e testes.	PROCERGS	1999/ 2000
Melhoria no processo de desenvolvimento	Desenvolvimento do Padrão Web de interfaces gráficas para sistemas transacionais.	PROCERGS	2001
Melhoria no processo de desenvolvimento	Definição de processos de prototipação de interfaces de sistemas transacionais na web.	PROCERGS	2001
Melhoria no processo de desenvolvimento	Criação e descrição de novos papéis no desenvolvimento de software com responsabilidades específicas sobre questões de IHC.	PROCERGS	2001/ 2002
Melhoria no processo de desenvolvimento	Desenvolvimento da Aplicação Modelo para sistemas transacionais na web, integrando as atividades dos designers e dos programadores.	PROCERGS	2002
Formação/Capacitação	Realização do Curso de projeto de Interfaces Gráficas ministrado pelo Professor Marcelo Soares Pimenta para os técnicos da PROCERGS.	UFRGS	2003

As atividades acima permitiram a elaboração de um conjunto de recomendações que julgamos relevantes para qualquer instituição que esteja em processo de criação de uma cultura de IHC no desenvolvimento de sistemas interativos. Algumas das atividades mostradas acima, selecionadas a partir do grau de importância e contribuição para os resultados gerados no dia-a-dia da empresa, são discutidas a seguir:

Melhorias no processo de desenvolvimento – Equipes Multidisciplinares

As equipes de desenvolvimento da empresa contavam até o início dos anos 90 apenas com programadores e analistas de sistemas. Com o aumento da demanda por sistemas com interface gráfica outros papéis tornaram-se necessários.

A PROCERGS é uma empresa de economia mista sujeita a uma legislação que obriga a realização de concurso público para o ingresso de novos profissionais. Em função disso, foi uma das primeiras empresas públicas de informática a realizar concurso público para a função “designer” sendo que até hoje foram realizados três concursos públicos para preencher vagas nas equipes de trabalho. No total a empresa possui 11 designers atuando em projetos de interface gráfica, todos com formação superior em Desenho Industrial e áreas afins.

O papel de projetista de interface também foi introduzido na empresa e atualmente é exercido oficialmente por 2 analistas com formação na área de informática, sendo que as equipes de projeto são idealmente formadas por gerentes de projeto, analistas de sistemas, designers, projetistas de interface, e programadores, embora esta seja uma mudança implementada ainda gradualmente na área de desenvolvimento. É claro que outros perfis como, por exemplo, documentadores e testadores também participam dos projetos, mas este trabalho procura focar o relato nas experiências relacionadas ao projeto de interfaces.

As inter-relações entre esses diferentes papéis podem ser observadas de forma simplificada na Figura 1. Também foi criada, para dar suporte às demandas dessas equipes, a função de consultor para projetos de interfaces e usabilidade exercida atualmente por 2 técnicos, um com formação superior em Desenho Industrial e o outro com formação em Informática.

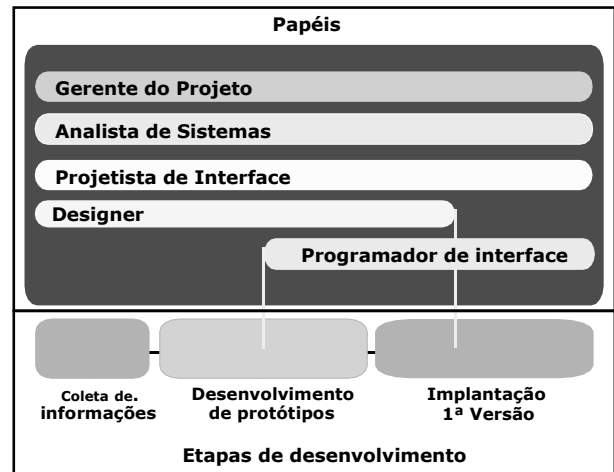


Figura 1 – Papéis no desenvolvimento PROCERGS

A função dos designers é a de prospectar, projetar, especificar e desenvolver soluções de design gráfico (diagramação, cores, estilos, desenhos, etc...) para os sistemas de informação em desenvolvimento ou manutenção. Os projetistas de interface são responsáveis pela elaboração da interface (diálogo e apresentação), os analistas pela arquitetura do sistema e pelas definições tecnológicas e o gerente de projeto pela coordenação e acompanhamento do projeto. Esses papéis não implicam que a equipe deva ser composta necessariamente por vários profissionais pois, dependendo do porte do projeto, um profissional poderá exercer mais de um papel ao mesmo tempo.

Melhorias no processo de desenvolvimento – Processo de Prototipação e Definição de Padrões

A definição de padrões de interface para os sistemas transacionais na empresa vem proporcionando maior velocidade no desenvolvimento de interfaces gráficas, além de garantir consistência de estrutura e procedimentos entre diferentes sistemas, reduzindo a curva de aprendizado dos usuários e dos desenvolvedores no médio prazo. Os padrões de interface definem diretrizes aplicáveis às principais transações encontradas nos sistemas como inclusão, exclusão, pesquisa e edição.

As diretrizes recomendadas pelos padrões de interface da empresa seguiram orientações existentes de autores como, Righi, Parizotto, Nielsen [13, 12, 8], do LabiUtil [6] e das normas ISO 9241. Essas diretrizes preocuparam-se também em adequar os conceitos e

normas às tecnologias disponíveis na instituição, resultando em um padrão para sistemas transacionais em plataforma desktop (doravante denominado padrão desktop) e em um padrão em plataforma web (doravante denominado padrão Web).

A estrutura básica do padrão web pode ser observada nas Figuras 2 e 3 . A Figura 2 mostra o sequenciamento de diálogo mais comum existente em sistemas transacionais da empresa.

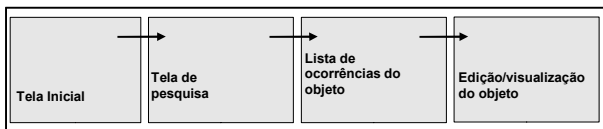


Figura 2 - Estrutura Geral de Navegação

Embora sugerida sua adoção generalizada, esta estrutura de navegação do padrão pode ser alterada de acordo com as características de cada sistema. Se, por exemplo, um sistema apresentar poucas ocorrências para um objeto, o padrão permite que seja exibida a tela de lista de ocorrências do objeto sem que o usuário necessite realizar uma pesquisa prévia. Essa flexibilidade foi um dos principais requisitos do projeto que definiu o padrão web.

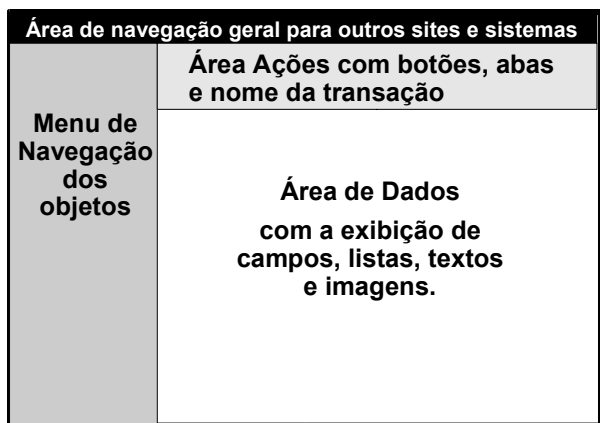


Figura 3 – Áreas do Padrão Web

A partir da definição de uma estrutura de navegação e de áreas específicas foi possível detalhar e aplicar as diretrizes propostas em cada diálogo e posteriormente, com o objetivo de aumentar a produtividade no desenvolvimento, foram construídos modelos de interface para os projetistas utilizarem nas etapas de prototipação (Figura 4) também sugeridas pelo padrão web. Segundo Kelley [5], a prototipagem soluciona problemas e pode-se fazer protótipos de qualquer coisa – um novo produto, serviço ou promoção especial. O que conta é seguir em frente, atingindo parte de sua meta.

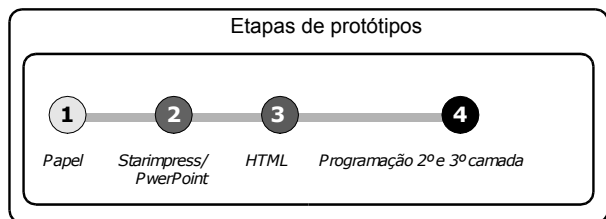


Figura 4 – Etapas de prototipação

Atualmente, quase todos os projetos de novos sistemas transacionais da empresa, que usam as tecnologias da web, utilizam-se dos conceitos, métodos e ferramentas definidos pelo Padrão Web PROCERGS que sugere uma prototipação composta por 4 níveis:

Prototipação em papel - Esboços da navegação geral

A primeira consiste na elaboração de esboços no papel (ver Figura 5) para a definição do fluxo de navegação. Nesta etapa os analistas de sistemas, designers e projetistas de interface definem questões básicas sobre como o usuário terá acesso aos sistemas e como poderá sair dele. É possível discutir nesta fase qual será a tarefa crítica a ser escolhida para detalhamento sendo que o padrão web recomenda que a escolha recaia para a tarefa mais utilizada ou a mais complexa do sistema.

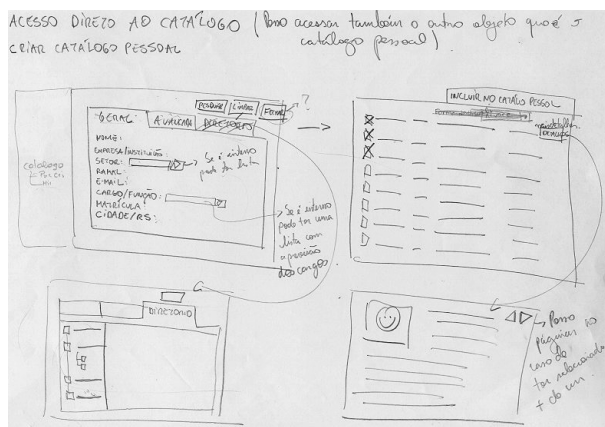


Figura 5 – Esboços no papel para definição da navegação geral

Prototipação em papel – principais diálogos

Neste momento o analista de sistemas e o projetista de interface utilizam-se de um modelo (ver Figura 6) que permite que informações e lógicas de funcionamento e uso dos diálogos necessários para a conclusão da tarefa crítica sejam discutidas e organizadas no papel sem a necessidade de conhecimento prévio de nenhuma ferramenta de programação.

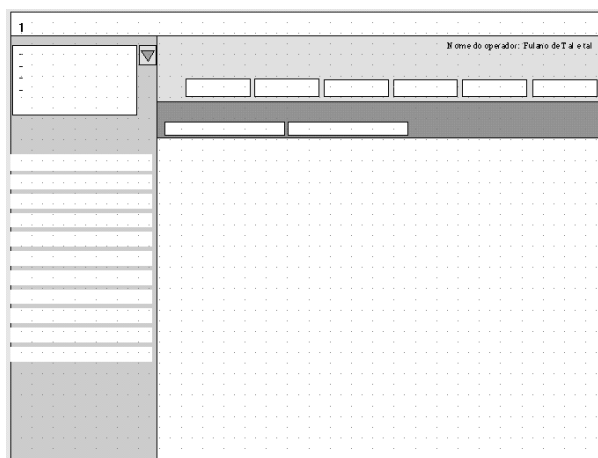


Figura 6 – Modelo para prototipação em papel

O analista ou o projetista de interface imprime várias cópias do modelo e depois a lápis inclui o nome dos itens de menu, os nomes de botões específicos do sistema, as abas de relacionamentos e os campos necessários para o entendimento dos diálogos pertencentes às tarefas críticas do sistema. Esse recurso permite que a equipe de projeto possa ter uma visualização rápida das possíveis alternativas de solução.

Prototipação em Starimpress ou Powerpoint

Esta etapa do trabalho é recomendada no momento em que a equipe de trabalho definiu algumas questões cruciais referentes aos diálogos e serve como importante instrumento de documentação visual do projeto.

Os analistas e projetistas de interface podem realizar essa etapa sem necessitar de conhecimentos aprofundados em ferramentas de apresentação como o powerpoint ou o starimpress.

Os desenvolvedores têm a sua disposição modelos prontos (ver Figura 7) nessas ferramentas para os principais diálogos (pesquisa, inclusão, exclusão e edição) e necessitam apenas inserir, de acordo com as definições do protótipo em papel, as informações específicas do projeto.

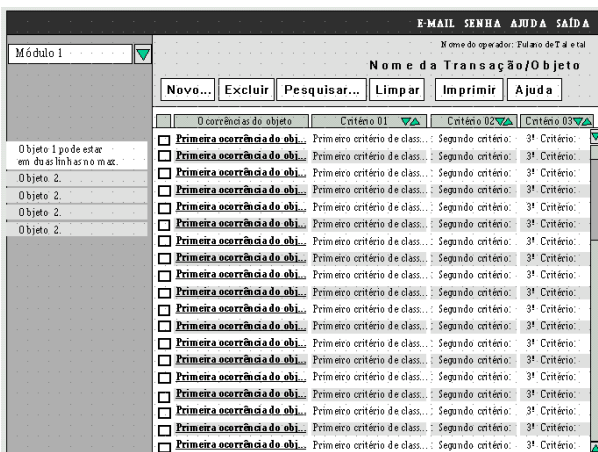


Figura 7 – Diálogo modelo para o retorno de uma pesquisa

Prototipação em html através da aplicação modelo

Para esta etapa foi construída uma aplicação modelo (ver Figura 8) com os principais diálogos propostos no padrão web. Seu desenvolvimento teve como princípio a integração das atividades realizadas por designers e programadores na camada de interface. Os designers são responsáveis pela personalização do projeto de interface (estilo gráfico, cores, textos, ícones, etc...) e os programadores pelo comportamento da interface (programação de primeira camada em html e javascript) e pela inclusão dos campos dos diálogos. Nesta etapa é possível apresentar os principais diálogos da interface para o cliente e para os usuários e submetê-los a um primeiro processo de avaliação.

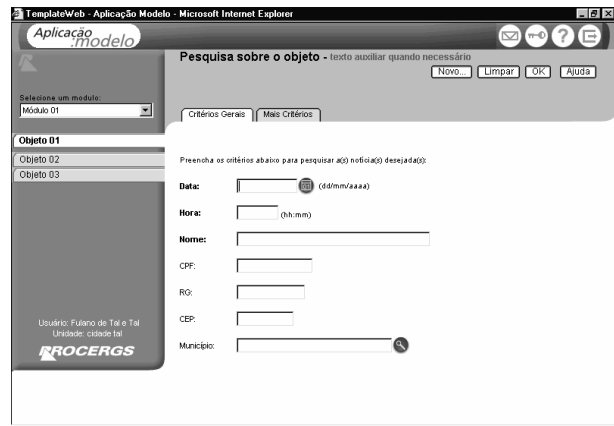


Figura 08 – Aplicação que integra atividades de design e programação

Este protótipo pode ser utilizado para dar seqüência ao trabalho de programação da 2ª e da 3ª camadas permitindo o uso de diferentes linguagens de programação. A partir dele, todos os demais diálogos do sistema podem ser construídos (programação das regras de negócio e do acesso ao banco de dados), mas não é propósito deste artigo o detalhamento das fases de programação.

A capacitação dos desenvolvedores

De acordo com Cybis [1], os problemas da não ergonomia são apontados por longos tempos de aprendizado, altas taxas de erro, aborrecimentos e frustrações. Por isso, nos últimos anos, têm sido feitos esforços para melhorar o nível de conhecimento dos profissionais da PROCERGS quanto aos conceitos ergonômicos, visando a melhoria da qualidade dos sistemas por ela desenvolvidos.

Foram realizadas diversas iniciativas para a capacitação do corpo funcional como: a realização de eventos e de cursos, o incentivo a participação em eventos (p.ex. IHC2000 e IHC2001) , a contratação de consultorias, a publicação de artigos e a promoção de palestras customizadas para cada grupo participante de algum projeto novo da empresa. Somente nos 5 cursos ministrados por professores de instituições como PUCRJ, UFSC, UFRGS e UFSM foi atingido um número de cerca de 50 técnicos da empresa que representam aproximadamente 12,5% dos 400 desenvolvedores. Sempre que é possível procura-se realizar novas palestras sobre o assunto IHC.

Desenvolvendo sistemas com IHC em mente - O software Direto

O *Direto*, um sistema de correio e agenda corporativos, baseado em protocolos abertos e totalmente direcionado ao ambiente *Web*, possui atualmente cerca de 15 mil usuários somente no estado do Rio Grande do Sul sendo que o produto também está sendo implantado em outros estados brasileiros.

Por tratar-se de um produto de uso geral e amplo, a empresa decidiu dispensar esforços especiais à concepção ergonômica da interface gráfica. Neste projeto foram

utilizados diversos métodos ergonômicos. Entre eles está o Design Macroergonômico [4] que ajudou na definição e na priorização dos itens demandados pelos usuários através da mensuração dos índices de importância e de satisfação. O DM ofereceu análises estatísticas que permitiram quantificar e hierarquizar os itens de design que deveriam ser melhorados, apoiando as decisões projetuais da equipe de desenvolvimento. A metodologia sugerida pelo DM e seu processo de utilização na concepção e nos testes com usuários do Direto já foram amplamente discutidos em outras publicações [2, 3].

Além do DM, pode-se destacar a utilização de Diretrizes de Design [12], de Diretrizes Ergonômicas [6], da Avaliação Heurística [10] e das Listas de Verificação [6] que apoiaram a construção e a revisão da usabilidade e das funcionalidades da interface do Direto. Estes métodos complementaram-se no processo de desenvolvimento da interface e apontaram problemas ergonômicos, sugerindo soluções para as futuras versões do sistema, sem deixar de avaliar os esforços necessários para implementá-las.

Avaliação de usabilidade das interfaces projetadas

Sabe-se que a Avaliação de Usabilidade é um nome genérico para um grupo de métodos baseados na avaliação e inspeção ou no exame relacionado com aspectos de usabilidade da interface com o usuário [7]. Devido ao custo envolvido no processo de avaliação e ao tempo e estrutura necessários para pôr os métodos de avaliação em prática apenas alguns produtos-chave da empresa participaram de um processo sistemático de avaliação de usabilidade.

Em 1998 o PROCERGS Escola (software de gestão escolar) foi submetido a avaliação com especialistas do LabiUtil. Este sistema tornou-se um ponto de referência para outros sistemas da organização pois a partir da avaliação de sua usabilidade e funcionalidade foi possível construir o padrão desktop que define diretrizes da empresa para projetos de interface .

O primeiro produto que foi submetido à avaliação de usabilidade com usuários foi o Direto devido a sua importância estratégica e aos riscos envolvidos em um projeto deste porte. Atualmente a aplicação modelo, que implementa as diretrizes propostas no padrão web, está sendo submetida a um processo de avaliação heurística e ensaios de interação [10], realizada por instituição externa e independente, com o objetivo de que problemas ainda não detectados possam ser corrigidos.

DISCUSSÃO: DIFICULDADES ENCONTRADAS , LIÇÕES APRENDIDAS E ALGUMAS RECOMENDAÇÕES

Nesta seção são discutidas algumas dificuldades encontradas durante os esforços empreendidos nesse período que resultaram em experiências fundamentais para o processo de aprendizado corporativo.

Dificuldades encontradas

Um processo de implantação de novos conceitos e métodos acaba por modificar a forma como as pessoas trabalham. Por isto, muitas delas apresentam um grau elevado de resistência e ceticismo num primeiro momento. Talvez este seja um dos aspectos mais difíceis de ser gerenciado na implantação de IHC no dia-a-dia de trabalho, principalmente em empresas (como a PROCERGS) onde os técnicos já possuem crenças consolidadas ao longo de vários anos de experiência.

A falta de disciplinas de IHC nos cursos de graduação de informática e de design também retardou o processo de adoção, já que foi preciso dispensar esforços adicionais na sensibilização dos técnicos sobre o assunto.

Outra dificuldade refere-se ao fluxo de definição e aprovação dos projetos, que na maioria das vezes é realizada pelo cliente sem o envolvimento direto dos usuários finais. Como consequência, quem acaba avaliando a implementação das tarefas pelo sistema não são os usuários reais desse sistema. Assim, muitos dos problemas de interface acabam sendo identificados no processo de implantação, resultando em custos mais elevados na sua correção do que se tivessem sido identificados na fase inicial de prototipação.

No início foi difícil aplicar os padrões de interface apenas com documentação e treinamento, pois muitos desenvolvedores foram modificando a interface, ocasionando diferenças entre o sistema e as recomendações existentes no padrão web da empresa. Somente após a implementação dos diálogos em HTML e Javascript é que o volume de alterações em relação ao padrão web foi reduzido. Mesmo assim, muitos desenvolvedores pressionaram as equipes de infraestrutura da empresa para que construíssem geradores automáticos de interface com código visando agilizar ainda mais o seu trabalho.

Os ganhos de qualidade e produtividade nos projetos de interface aconteceram efetivamente após a implementação de ferramentas que auxiliaram na etapa de prototipação.

A pressão dos clientes por prazos menores muitas vezes obrigava os desenvolvedores a buscarem as soluções mais rápidas que, necessariamente, não eram as mais adequadas, acarretando em perdas na qualidade da usabilidade dos sistemas. Multiplicavam-se, desta maneira, muitas formas diferentes de fazer as interfaces e que formavam uma barreira natural ao processo de aprendizagem e disseminação do conhecimento em IHC.

A inexistência na empresa de um processo de auditoria das interfaces fez com que muitos projetos fossem alterados ao longo do tempo e perdessem as características conceituais iniciais.

Em alguns projetos o gerenciamento das equipes multidisciplinares foi complexo, pois havia a necessidade

de conciliar e mediar diversos pontos de vista em relação às soluções para os problemas de IHC.

Lições aprendidas

Alguns dos aprendizados ocorridos durante o período em que a PROCERGS investiu na consolidação de IHC estão listados abaixo.

- **Importância da documentação:** alguns dos participantes do projeto Direto saíram da empresa e com eles foi uma parte significativa dos conhecimentos necessários para a resolução dos problemas de interface. Muitas das soluções discutidas não foram devidamente documentadas o que resultou em perdas significativas de tempo no aprendizado dos desenvolvedores que substituíram aqueles que deixaram a empresa;
- **Importância do comprometimento da equipe:** foi necessário dispor de tempo para convencer os novos participantes das equipes dos projetos a respeito da importância dada pela empresa para a usabilidade dos sistemas. Sem o convencimento das equipes é muito difícil implementar na prática qualquer conceito ou método novo;
- **Importância da integração entre métodos:** os métodos de concepção e avaliação em IHC devem estar integrados aos métodos existentes de desenvolvimento de sistemas como a orientação a objetos. O método de desenvolvimento deve incorporar em suas etapas os métodos de concepção e avaliação em IHC;
- **Importância das equipes multidisciplinares:** a formação de equipes multidisciplinares com a presença de designers, consultores em IHC, analistas de sistemas e programadores permitiu que as discussões fossem enriquecidas com a inserção de diversos pontos de vista a respeito de como implementar a usabilidade e isso contribuiu para o resultado final;
- **Importância das ferramentas:** sem elas é muito difícil modificar procedimentos já existentes por cada equipe de projeto. As equipes tendem a ser menos resistentes no momento em que percebem reais vantagens na redução de tempo das suas atividades e na melhoria da qualidade final dos produtos.

Dicas úteis e recomendações

A síntese das questões expostas neste trabalho está listada a seguir sob a forma de dicas e recomendações para aqueles que desejam consolidar IHC em suas instituições.

- Os esboços realizados no projeto de interface devem ser guardados em uma pasta específica e somente os esboços principais são digitalizados, pois constituem-se em uma importante documentação sobre muitas decisões tomadas durante os projetos.
- Tenha paciência no processo de mudança cultural no desenvolvimento de interfaces de sistemas, pois leva tempo até que as pessoas assimilem os conhecimentos

e consigam colocar em prática novos métodos e técnicas.

- Em projetos-chave utilize os serviços de avaliação de usabilidade realizados por instituições externas independentes. Elas aumentam a credibilidade interna do que foi proposto e reduzem disputas internas, evitando o efeito “santo de casa não faz milagre”.
- Primeiro experimente e depois melhore. Tome muito cuidado com o excesso de preciosismo. Faça diversos protótipos e, na dúvida, prototipe ainda mais. Muitos problemas acabam se perpetuando sem que ninguém tenha feito protótipos para validá-los.
- Inicie discutindo primeiro o fluxo geral e depois discuta questões específicas utilizando o papel e o lápis para fazer isso. O protótipo de papel é o que tem a melhor relação custo-benefício.
- Não se preocupe em desenvolver 100% das tarefas. Resolva sempre a tarefa crítica (a mais usada ou a mais complexa) que o restante das tarefas acaba se encaixando nas suas definições, salvo, é claro, exceções que podem ser tratadas caso-a-caso.
- Apresente protótipos ao cliente e aos usuários o mais rápido possível e evolua o projeto junto com eles através de um processo participativo [5]. Discuta aspectos críticos e dúvidas diretamente com clientes e usuários através do esboço.
- Faça apresentações para os desenvolvedores dos novos métodos em IHC de forma individualizada e de acordo com a necessidade das equipes, pois é mais eficiente do que palestras para grandes públicos. No pequeno grupo muitos dos técnicos ficam mais a vontade para solucionar suas dúvidas.
- Capacite, sempre que possível, dois profissionais vindo de áreas de formação distintas para trabalhar em projetos de maior complexidade ou para a prestação de serviços internos de assessoria aos projetos. Isso agregará pontos de vista distintos sobre as soluções tornando-as mais abrangentes.
- Construa e ofereça ferramentas, pois o desenvolvimento de padrões deve ser seguido pelo uso de ferramentas que os implementem. Sem ganhos de produtividade reais é muito difícil mudar uma cultura de desenvolvimento.
- Mantenha perfis diferentes de profissionais nas equipes de trabalho. Contrate designers, especialistas em IHC, e tenha especialistas no negócio do cliente participando do desenvolvimento da interface gráfica.
- Priorize a utilização de métodos flexíveis, baratos e fáceis de usar. Antes de adotar um método de IHC é importante observar: a experiência anterior bem sucedida na aplicação do método, a disponibilidade de

recursos físicos e equipamentos, o custo para a aplicação dos métodos, o tempo disponível dentro do cronograma de desenvolvimento, a flexibilidade do método na sua adaptação ao projeto, a facilidade de aplicação do método por outros profissionais na empresa, a possibilidade de estender para outros produtos de forma simples, a possibilidade de envolvimento e participação dos usuários e o *Know-how* necessário para uma boa condução do método. Se o método não atender a esses requisitos deve-se optar por outro que satisfaça as necessidades da empresa com a melhor relação custo-benefício.

CONCLUSÕES E PERSPECTIVAS

Diversos desafios ainda permeiam o desenvolvimento das interfaces gráficas com usabilidade, sendo que a consolidação dos conceitos de IHC passa pelo domínio de métodos e técnicas que muitas vezes está restrito a um grupo de especialistas, enquanto os outros profissionais ainda não dominam a maioria dos procedimentos. É uma grande meta atingir um número maior de projetos e de profissionais, incluindo sempre os usuários do processo de desenvolvimento de software. Como fazer isso é um dos grandes desafios da empresa daqui para frente.

Sabendo da necessidade de construção de ferramentas que agilizem o processo de desenvolvimento das interfaces a PROCERGS está investindo em 2004 no desenvolvimento de um gerador dinâmico de interfaces que implementará o código a partir das diretrizes existentes no padrão web e das características de cada sistema. Além disso, a empresa deve investir ainda mais em avaliações sistemáticas de usabilidade em seus produtos-chave.

Norman [11] afirmou que cada vez mais os problemas existentes nos computadores devem migrar para outros dispositivos. É urgente a realização de pesquisas com enfoque em IHC ou IHT (Interação Humano-Tecnologia) sobre o uso de sistemas transacionais em outros dispositivos (celular, palms, etc...). As empresas necessitam realizar projetos piloto com usuários para validar e adaptar as diretrizes existentes hoje a esses dispositivos, para tornar viável a construção de ferramentas que garantam produtividade aos próximos projetos, num futuro não muito distante.

A montagem de um laboratório de usabilidade deve ser averiguada, pois ele pode ser usado em avaliações heurísticas e na realização de testes com grupos de usuários. Segundo Nielsen [9] esses laboratórios foram implantados nas companhias desenvolvedoras de *software* mais fortemente a partir de 1989. O autor sugere que não é preciso começar com uma grande estrutura para um laboratório e que o importante é começar para crescer depois. Enquanto o laboratório não é uma realidade deve-se procurar alternativas que minimizem a falta de um ambiente especialmente preparado, como a observação informal dos usuários durante a realização das atividades no computador e o compartilhamento de recursos através de convênios com

instituições que disponham de uma infra-estrutura de laboratório de usabilidade.

Este trabalho relata as ações práticas no dia a dia de uma empresa de informática que vem tentando consolidar uma cultura de IHC. A maioria dos problemas (e conseqüentemente das soluções) e argumentações valem para uma ampla classe de organizações que enfrentam o desafio da inserção de IHC nas suas práticas de desenvolvimento de software interativo. Espera-se que o compartilhamento dessas experiências seja útil para outras instituições que estejam planejando o mesmo e que isso favoreça, por conseqüência, uma melhor qualidade das interfaces de sistemas e dos conhecimentos conceituais e metodológicos produzidos hoje no país.

REFERÊNCIAS

1. Cybis, W.A.; Pimenta, Marcelo Soares; Silveira, M.C.; Gamez, L. Uma Abordagem Ergonomica para o Desenvolvimento de Sistemas Interativos. In: Anais do I Workshop Brasileiro Sobre Fatores Humanos Em Sistemas Computacionais, IHC 98, 1998.
2. Endler, Antônio M. Método do Design Macroergonômico aplicado ao Desenvolvimento de um *Software* Corporativo de Correio e Agenda na *Web*. Dissertação de Mestrado, PPGEP/UFRGS - Programa de Pós-graduação em engenharia de produção/Universidade Federal do Rio Grande do Sul, Porto Alegre, 2000.
3. Endler, A. M., Fogliatto, F. S., Guimaraes, L. B. M. Design Ergonômico de Software de Correio e Agenda Corporativos na Web In: IHC 2001, 2001, Florianópolis. IHC 2001 4 Workshop sobre Fatores Humanos em Sistemas Computacionais. , 2001. v.1. p.96 – 107
4. Fogliatto, Flávio.; Guimarães, Lia B.M. Design Macroergonômico: uma proposta metodológica para projeto de produto. *Produto & Produção* Porto Alegre, v.3, n.3, 1999.
5. Kelley, Tom; Litman, Jonathan A Arte da Inovação - Lições de criatividade da IDEO, a maior empresa norte-americana de design Ed Futura. São Paulo, 2001.
6. Labiutil/Ufsc - Laboratório de Utilizabilidade/ Universidade Federal de Santa Catarina. Web ErgoList. <http://www.labiutil.inf.ufsc.br/ergolist>, (documento recuperado em 29/04/1999).
7. Mack, Robert L, e Nielsen, Jakob. Usability inspection methods: rapport on a workshop held at chi'92. Monterey, CA, maio,1992. SIGCHI Bulletin, pag. 28-33MARCUS, Aeron. Graphical User interfaces. In HELANDER, M. Hand book of Human Computer-Interaction. 2ed . Elsevier, 1997. pag 423-440.
8. Nielsen, Jakob. Ten good deeds in Web Design. <http://www.useit.com/alertbox/991003.html>,

- outubro, 1999 (documento recuperado em 28/08/2000).
9. Nielsen, Jakob. Usability laboratories. *Behaviour & Information Technology* 13, 1&2, 3-8, 1994.
 10. Nielsen, Jakob. Technology Transfer of Heuristic Evaluation and Usability Inspection. *In IFIP Interact 95 Lillehammer, Norway, junho, 1995.* http://www.useit.com/papers/heristic/learning_inspection.html (documento recuperado em 17/04/2000).
 11. Norman, Don. Palestra convidada do IHC2001 Gramado-RS, 2001.
 12. Parizotto, Rosamelia. Elaboração de um Guia de Estilos para Serviços de Informação em Ciência e Tecnologia via Web. Dissertação de Mestrado, PPGEP/UFSC - Programa de Pós-graduação em engenharia de produção/Universidade Federal de Santa Catarina, Florianópolis, 1997.
 13. Righi, Carlos A. R. Aplicação de Recomendações Ergonômicas ao Componente de Apresentação da Interface de Softwares Interativos. Dissertação de Mestrado, PPGEP/UFSC - Programa de Pós-graduação em engenharia de produção/Universidade Federal de Santa Catarina, Florianópolis, 1993.

Integração IHC e ES: Processo de Planejamento da Reengenharia de Software Guiado pela Avaliação de Usabilidade – PPR-U

Sérgio Luisir Díscola Junior
André Constantino da Silva

Júnia Coutinho Anacleto Silva
Marcela Xavier Ribeiro

UFSCar
Rod. Washington Luís, Km 235
[sergiold; junia; acons; marcela}@dc.ufscar.br](mailto:{sergiold; junia; acons; marcela}@dc.ufscar.br)
+ 55 – 16 – 2608232

RESUMO

Neste artigo, a integração das áreas de Engenharia de Software e Interação Humano-Computador é abordada sob a forma da introdução de um novo Processo de Planejamento de Reengenharia de Software, chamado PPR-U, onde as fases, atividades e passos são guiados pela Avaliação de Usabilidade. Para tanto, todas as fases desse novo Processo são apresentadas bem como um estudo de caso relatando o experimento realizado a fim de testar o Processo desenvolvido.

ABSTRACT

In this paper, the integration of Software Engineering and Human Computer Interaction is approached, presenting a new Software Reengineering Planning Process, called PPR-U, where its phases, activities and steps are guided by Usability Evaluation. For this purpose, all phases of this new Process are presented as well as the experiment done in order to test the developed Process.

Palavras-chave

Avaliação de Usabilidade, Reengenharia, Planejamento.

INTRODUÇÃO

A integração das áreas de Interação Humano-Computador (IHC) e Engenharia de Software (ES) ainda está em um nível muito incipiente e ainda muito esforço é necessário para alcançar resultados satisfatórios. Dentre o oceano de pesquisas existente para a integração dessas duas áreas, encontra-se a integração entre a Reengenharia de Software e a IHC. A Reengenharia de software consiste nas fases de: *Engenharia Reversa* + Δ + *Engenharia Avante*, onde a documentação do Software Legado é revitalizada, o projeto modificado e, por fim, implementado.

Um motivo marcante para a necessidade de realizar Reengenharia é a falha na Usabilidade de um software Legado, e um fator crítico para o sucesso da Reengenharia é o seu correto Planejamento. Os Processos de Planejamento existentes não estão especificados em

um nível tal que os planejadores possam encontrar um conjunto de passos concretos que devem ser realizados para planejar a Reengenharia. Além disso, esses Processos não contemplam de maneira formal a Avaliação de Usabilidade. Portanto, o objetivo deste artigo é especificar um Processo de Planejamento de Reengenharia de Software com fases, atividades e passos guiados pelos resultados obtidos na Avaliação de Usabilidade. Para tanto, este artigo está estruturado da seguinte maneira:

Inicialmente, três Processos de Planejamento de Reengenharia de Software são apresentados. Em seguida, os métodos de Avaliação de Usabilidade são mostrados e, na seqüência, o PPR-U é explanado. Um estudo de caso é mostrado, e por fim algumas conclusões da pesquisa face aos resultados obtidos pelo experimento são apresentadas.

PROCESSOS DE PLANEJAMENTO DE REENGENHARIA DE SOFTWARE

O Planejamento da Reengenharia de Software deve ser realizado com os objetivos de: (a) estimar custos e prazos. (b) verificar a viabilidade da aplicação da Reengenharia e obter os benefícios com a modificação do software. (c) determinar as atividades necessárias para a Reengenharia. Para tanto, alguns Processos têm sido desenvolvidos e empregados. A próxima seção explora alguns Processos de Planejamento de Reengenharia de Software a fim de se entender como os métodos de Avaliação de Usabilidade podem apoiar o Planejamento.

PROCESSO DE PLANEJAMENTO DE HARRY SNEED

Harry Sneed apresenta em [6] um Processo de Planejamento de Reengenharia de Software composto por cinco fases:

1. Justificativa do projeto de Reengenharia: análise da qualidade técnica do software, do Processo de manutenção e do valor de negócio.
2. Análise de portfólio: elaboração de um gráfico informando a prioridade das aplicações do software.
3. Estimativa de custo: cálculo do custo da Reengenharia.
4. Análise do custo x benefício: o custo da Reengenharia é comparado com os ganhos esperados na manutenção.
5. Contratação: elaborar os contratos firmados com os clientes.

Uma limitação constatada nesse Processo é que durante o Planejamento da Reengenharia, não se deve realizar mudanças no software. Pode-se observar que as informações relativas aos usuários não são exploradas. No entanto, o usuário tem um papel importante na identificação da necessidade de mudança do software existente, então, faz-se necessário os métodos de Avaliação de Usabilidade.

PROCESSO DE PLANEJAMENTO SEGUNDO RANSOM, SOMMERVILLE E WARREN

Esse método de Avaliação do sistema Legado foi gerado pelo projeto Esprit - RENAISSANCE da Universidade de Lancaster que tinha o objetivo de desenvolver um método de Reengenharia com foco em aplicações comerciais [5]. O método de Avaliação de sistemas Legados em questão foi projetado para oferecer estimativas confiáveis de custo e riscos relativos a um possível projeto de Reengenharia. Na visão apresentada em [5], algumas questões relativas à Avaliação do sistema Legado devem ser respondidas:

- Quais são os objetivos da organização?
- Em quais requisitos o software existente deve evoluir?
- Qual é o estado técnico do sistema?
- A organização que o sistema opera aceita mudanças?
- A organização responsável pela mudança detém recursos suficientes para aplicá-las?

A fim de responder essas questões, esse Processo propõe um método de Avaliação do sistema Legado composto por cinco etapas:

1. Instanciação: consiste em adaptar o método de Avaliação do sistema Legado às necessidades da organização. As técnicas de Avaliação são definidas através de opiniões de especialistas que se baseiam na maturidade da organização em realizar tais avaliações.
2. Avaliação do valor de negócio: nessa etapa é analisada a importância do sistema para a organização.
3. Avaliação do ambiente externo: deve-se avaliar o hardware, o software e a infra-estrutura da organização, com o intuito de analisar o Processo de manutenção.
4. Avaliação da aplicação: a aplicação é avaliada segundo características, tais como documentação existente, manutenibilidade, segurança e casos de testes.
5. Interpretação dos resultados: os resultados obtidos das avaliações efetuadas anteriormente são utilizados para determinar a qualidade técnica, bem como o seu valor de negócio. A comparação desses dois elementos fornece bases para se decidir qual o tipo de Processo de evolução de sistemas que será empregado na situação em que o sistema Legado se encontra.

Esse método fornece meios para se obter um maior entendimento do sistema Legado avaliado, a fim de analisar qual é o melhor modo de evolução que pode ser empregado. Além disso, esse Processo analisa as propriedades técnicas, organizacionais e de negócios que

são informações empregadas na determinação de qual método de evolução deverá ser aplicado: Reengenharia, manutenção ou substituição do sistema existente.

O autor menciona a necessidade de se conhecer os objetivos da Reengenharia, bem como os requisitos do sistema existente que devem ser modificados. Nesse ponto, a Avaliação de Usabilidade é fundamental, pois a falha na Usabilidade é um forte motivo para a Reengenharia de Software.

PROCESSO DE PLANEJAMENTO SEGUNDO TILLEY

A fim de implementar um Planejamento da Reengenharia de Software, Tilley em [7] mostra um conjunto de questões que devem ser respondidas, dentre elas:

1. Quais são os motivos para a Reengenharia?
2. Quais são os fatores críticos de sucesso para o projeto de Reengenharia?
3. Quais objetivos a atualização do sistema deve atender?
4. Como os resultados do projeto de Reengenharia serão medidos?

Para tanto, o Planejamento da Reengenharia deve descrever as atividades que esse Processo contém e os recursos que necessitam. As informações usadas na definição das atividades são: a utilidade do sistema, as mudanças funcionais requeridas quais tarefas sofrerão modificações e quais critérios arquiteturais serão utilizados para avaliar o sistema. Uma vez realizada a descrição dessas informações a fim de definir as características das atividades, o plano deve descrever quais métodos, ferramentas, metodologias e tecnologias serão empregadas na Reengenharia do sistema. Ao final dessas definições, é elaborado um cronograma e são determinados os locais e as datas de aplicação da Reengenharia no sistema em questão. Então, o custo do projeto de Reengenharia é estimado através da utilização de métricas definidas. Por fim, deve-se fazer um levantamento dos riscos concernentes à aplicação da Reengenharia como Processo de evolução do software. Dessa forma, é possível realizar o Planejamento da Reengenharia identificando sua viabilidade através da análise dos riscos e dos benefícios da aplicação desse Processo de evolução de software. Uma vez comprovada a viabilidade da Reengenharia como método de evolução de software, os métodos de Reengenharia devem ser empregados para implantá-la. Segundo Tilley, deve-se analisar os motivos e os objetivos da Reengenharia e determinar quais mudanças devem ser empregadas. Para obter essas informações, também se podem utilizar os métodos de Avaliação de Usabilidade.

ANÁLISE DOS PROCESSOS DE PLANEJAMENTO FACE À QUESTÃO DA INTERAÇÃO HUMANO-COMPUTADOR

A respeito do Processo de Planejamento segundo Sneed, observou-se a lacuna de se avaliar questões de Usabilidade nos seguintes pontos:

- A primeira etapa (Justificativa do projeto de Reengenharia) desse Processo menciona a necessidade de comparar o Software Legado com padrões industriais ou através das exigências dos usuários, porém não explicita a forma de se realizar o levantamento dos parâmetros exigidos pelos usuários.

Uma Avaliação de Usabilidade poderia ser utilizada para realizar tal levantamento e quantificar as exigências dos usuários.

- A estimativa do custo-benefício da Reengenharia poderia ser realizada com a comparação do modelo GOMS [2] das tarefas do Legado com o gerado pela Reengenharia.
- A análise da qualidade técnica levantada por Sneed pode ser realizada através da Avaliação de Usabilidade para verificar a facilidade de uso e aprendizagem do Software.

O Processo de Planejamento segundo Ransom, Sommerville e Warren tem a necessidade da aplicação de métodos de Avaliação de Usabilidade nos seguintes pontos:

- Os objetivos da organização e os requisitos do sistema Legado que devem ser melhorados podem ser obtidos através da Avaliação de Usabilidade.
- Os especialistas podem analisar os aspectos de Usabilidade presentes no Software e estabelecer prioridades sobre a necessidade de mudança para cada tarefa do software.
- Na fase de Avaliação da aplicação pode-se avaliar a Usabilidade, pois a satisfação do usuário é importante para a determinação da necessidade de mudança.

Por fim, o Processo de Planejamento segundo Tilley necessita da aplicação de métodos de Avaliação de Usabilidade nas seguintes questões:

- O autor afirma que os motivos para Reengenharia devem ser analisados.
- Os objetivos da modificação do Software e os resultados da Reengenharia devem ser respondidas antes de escolher a Reengenharia como forma de modificação do Software.
- O autor sugere a necessidade de se conhecer quais mudanças funcionais são necessárias.

AVALIAÇÃO DE USABILIDADE

Usabilidade, segundo a norma ISO 9241-11, consiste na "amplitude a qual um produto pode ser utilizado por certos usuários para alcançar certos objetivos com eficácia, eficiência e satisfação em um certo contexto de uso" e consiste em um atributo de qualidade de Software. No contexto do Planejamento da Reengenharia de Software ele é um atributo chave, pois lida com as questões de eficácia, eficiência e satisfação dos usuários na sua interação com o Software. Portanto, o conhecimento do estado da Usabilidade do Software Legado, bem como do estado da Usabilidade que um Software que sofreu Reengenharia apresenta faz-se necessário para o Planejamento da sua Reengenharia. Dessa forma, a Avaliação de Usabilidade é um importante passo para o Planejamento da Reengenharia de Software. Essa Avaliação pode requerer a participação dos usuários ou não. Nesta pesquisa, a Avaliação Heurística [4], o Percurso Cognitivo [3] e o Teste de Usabilidade [1] foram os métodos utilizados para avaliar a Usabilidade no software Legado. Outra forma de se avaliar a Usabilidade de um Software é através da análise das suas tarefas. Um modelo que produz informações

interessantes para o PPR-U é o modelo GOMS o qual também é utilizado.

Avaliação Heurística

A Avaliação Heurística é realizada sem a participação dos usuários, portanto é um Método de Inspeção [4]. Essa análise leva em consideração as dez heurísticas de Nielsen e não apresenta técnicas para correção dos problemas, porém sugere uma reunião entre os avaliadores para a discussão de possíveis soluções dos problemas encontrados.

Em vista desse método de Avaliação de Usabilidade não exigir a presença de usuários, o custo de aplicação do mesmo torna-se baixo em comparação com os métodos de Avaliação de usabilidade que necessitam da presença dos usuários. Além disso, esse método é fácil e rápido de aprender e pode ser utilizado nas fases iniciais do desenvolvimento de um software, portanto, essas características fazem que a Avaliação Heurística seja um método de Avaliação de Usabilidade recomendado para apoiar o Planejamento da Reengenharia. Porém, uma vez que a Avaliação de Usabilidade é limitada nas dez heurísticas de Nielsen e pode gerar resultados falso-positivos, faz-se necessário utilizar outros métodos de Avaliação combinados com a Avaliação Heurística a fim de que a análise da Usabilidade seja a mais completa possível.

Percurso Cognitivo

Esse método não utiliza os usuários para realizar a Avaliação da Usabilidade da interface, e analisa os aspectos de Usabilidade concernentes à facilidade de aprendizagem da interface a qual é realizada através do Processo de exploração [3], que consiste em um Processo que os usuários aprendem o software através de tentativas e erros, sem treinamento prévio.

Como a Avaliação Heurística, o *Cognitive Walkthrough* não necessita da presença dos usuários, o que diminui os custos de sua aplicação. Além disso, esse método é fácil de aprender e usar e pode ser utilizado nas fases iniciais do desenvolvimento de um software. Já que se deseja obter um Processo de Planejamento de Reengenharia rápido e barato, o *Cognitive Walkthrough* pode ser utilizado para tal finalidade. Porém, devido a sua Avaliação ser limitada, também é necessário utilizar outros métodos de Avaliação de Usabilidade para prover uma análise completa e confiável da Usabilidade do sistema.

Teste de Usabilidade

O Teste de Usabilidade é realizado através da observação dos usuários representativos durante a execução da tarefa, o que é uma importante característica, pois a proximidade entre o avaliador e o usuário durante as sessões de teste estabelece um importante canal de comunicação entre essas pessoas a fim de detectar e corrigir problemas de Usabilidade existentes em uma interface através da interação entre o usuário e o avaliador [1]. Caso os participantes do teste sejam uma amostra de usuários representativos, logo os resultados obtidos reproduzirão eficazmente a realidade. Portanto, a aplicação do Teste de Usabilidade em um Processo de Planejamento é interessante para observar confiavelmente as métricas de

Usabilidade escolhidas para a observação do Software em questão.

Com o Teste de Usabilidade podem-se utilizar diversas técnicas de registro de informações como: gravação em vídeo, áudio e anotações dos observadores. Essa característica faz com que esse método seja rico em informações e ainda, tais informações são bastante próximas da realidade, ou seja, não contém um grau de subjetividade devido à opinião dos avaliadores uma vez que o que é analisado corresponde em exemplos práticos e não simulações ou suposições (como nos casos da Avaliação Heurística e Passeio Cognitivo).

Além de medidas qualitativas (como comportamento dos usuários face uma determinada situação, por exemplo), medidas quantitativas confiáveis, como tempo médio de execução de tarefas, número de erros cometidos, entre outras, são obtidas para análise. Essas medidas podem ser utilizadas para estimar a qualidade técnica de um Software através do cálculo do coeficiente de qualidade técnica do Software dado pela fórmula proposta em [6].

Modelo GOMS

Esse modelo é, geralmente, aplicado em Softwares cujos objetivos dos usuários são bem definidos (gerar relatórios, efetuar cálculos ou montar planilhas) [2]. Caso os objetivos dos usuários para utilizar o Software não sejam explícitos como, por exemplo, em uma revista eletrônica na qual o usuário navega pelas informações apenas para relaxamento sem um objetivo definido, o modelo GOMS não é recomendado, pois sua descrição é dirigida ao objetivo do usuário. Além disso, esse modelo é aplicado em usuários experientes que não precisam aprender o Software através da técnica da exploração do Software como visto no método Percurso Cognitivo.

O KLM (*Keystroke-Level Model*) apresenta seis operadores para descrever as ações que os usuários podem realizar através da interface. Cada um desses operadores possui estimativas de tempos de execução que podem ser encontrados em [2]. Como observado pelos experimentos realizados, a técnica KLM é simples e fácil de aprender e utilizar não requerendo muita experiência com as teorias cognitivas. Além disso, não necessita da observação de usuários, reduzindo o custo de aplicação.

Ainda de acordo com o observado nos experimentos realizados, o modelo GOMS produz estimativas de tempos de execução próximas do real. Portanto, esse modelo pode ser utilizado no Planejamento da Reengenharia para estimar um provável benefício que a interface produzirá após a sua Reengenharia. Por exemplo, caso seja estimado que uma certa tarefa de um Software Legado apresente um tempo de execução de 100 segundos, caso após a Reengenharia esse tempo cair para 90 segundos teremos um benefício de 10 segundos para cada execução desta tarefa e, conseqüentemente, diminuirá o custo operacional desse Software.

Além disso, através desse modelo é possível avaliar a consistência das interfaces de um Software, ou seja, através desse modelo pode-se estudar se o conjunto de operadores a serem executados pelos usuários realmente é o menor possível ou se esse conjunto ainda pode ser otimizado e se faz necessário para Planejar a Reengenharia da interface do Software.

PROCESSO DE PLANEJAMENTO DE REENGENHARIA DE SOFTWARE GUIADO PELA AVALIAÇÃO DE USABILIDADE: PPR-U

O Planejamento de Reengenharia tradicional leva em consideração apenas as questões de qualidade relacionadas com a funcionalidade do Software como, por exemplo, a Reengenharia do código Legado, das estruturas de dados, ou da sua interface. Porém, o Planejamento da Reengenharia guiado pela Avaliação de Usabilidade, além de levar em consideração esses aspectos da qualidade do Software, preocupa-se em analisar a Usabilidade do Software Legado de modo que a questão da interação humano-computador seja considerada desde o Planejamento da Reengenharia. Esse Planejamento guiado pela Avaliação de Usabilidade possibilita obter pela Reengenharia um Software com maior grau de Usabilidade.

Esta seção apresenta o Processo de Planejamento de Reengenharia guiado por Avaliação de Usabilidade e a importância da Avaliação da Usabilidade durante o Planejamento da Reengenharia.

Descrição do PPR-U

O PPR-U é baseado na proposta de Sneed [6] e define uma estrutura de Fases, Atividades e Passos a fim de especificar completamente todas as ações do planejador durante a aplicação desse Processo.

A Fase é a divisão mais abstrata das tarefas que o planejador deve realizar e é composta por um conjunto de Atividades que também é uma divisão abstrata de tarefas, porém com menor nível de abstração. E, por fim, os Passos são ações práticas que o planejador deve realizar com o intuito de aplicar o PPR-U. Portanto, cada Fase é composta por n Atividades que, por sua vez, é composta por m Passos. Os Passos são as ações mais básicas que os planejadores devem realizar para completar uma Atividade e, quando o planejador completar todas as Atividades de uma tarefa, uma Fase do PPR-U é finalizada. O PPR-U contém as cinco fases do Processo formulado por Sneed, porém essas cinco fases estão divididas em onze atividades que por sua vez estão divididas em quarenta e três passos baseados nos resultados obtidos pela Avaliação de Usabilidade. A seguir cada uma das fases do PPR-U é explicada através da apresentação das suas atividades e respectivos passos.

Fase-1: Justificativa do Projeto

Nessa fase justifica-se a Reengenharia através da análise da qualidade do Software Legado, do seu Processo de manutenção e do valor de negócio presente no Software Legado. Para tanto, as seguintes atividades devem ser realizadas:

Atividade-1: Análise da qualidade técnica do Software

Na análise da qualidade técnica de um Software, as métricas relevantes para justificar a sua mudança são quantificadas. Essas métricas podem ser de natureza funcional ou da sua Usabilidade. As métricas de natureza funcional são extraídas do código do Software ou dos seus elementos físicos (através dos arquivos do Software utilizados para processamento, acesso à banco de dados, entre outros elementos) utilizando Softwares chamados “auditores de código” [6]. Como exemplo de métricas de natureza funcional, tem-se: número de chaves de

estruturas de dados ou tabelas, número de relacionamentos entre arquivos ou tabelas, taxa de dependência dos dados, linhas de código, acessos à banco de dados, pontos de função, complexidade ciclométrica e de dados, entre outras. Já as métricas de natureza da Usabilidade são extraídas a partir da Avaliação dos componentes do Software responsáveis por tornar o Software fácil de usar, fácil de aprender e fácil de memorizar. Para definir e quantificar as métricas de Usabilidade os seguintes passos devem ser realizados:

- Passo-1. Elaboração do modelo GOMS das tarefas do Software Legado.
- Passo-2. Avaliação Heurística do Software Legado.
- Passo-3. Aplicação do "Percurso Cognitivo".
- Passo-4. Análise dos resultados obtidos nos passos um, dois e três dessa atividade.
- Passo-5. Teste de Usabilidade reportando os resultados através da norma *Common Industry Format* [1].

Atividade-2: Estimativa do custo de manutenção

Essa atividade destina-se a estimar o custo de manutenção do Software Legado. Essa informação é obtida através de um programa de mensuração do esforço de manutenção gasto pela organização que a mantém. Essa atividade é composta por um passo, que é explicado a seguir:

- Passo-1. Obtenção do esforço de manutenção.

Atividade-3: Avaliação do valor de negócio das tarefas disponíveis no Software

A Avaliação do valor de negócio das tarefas do Software deve ser realizada por analistas de valor de negócios de modo que esses valores devem ser registrados em forma de tabela, como mostrado na Tabela-1.

Tabela - 1. Exemplo de valores de negócio.

Tarefa	Valor de Mercado	Importância das informações	Pontuação
Cadastro de clientes	15	30	45
Cadastro de carros	15	20	35
Cadastro de Aluguel	35	35	70
Cadastro de Reserva	35	15	50
Total	100	100	200

Para quantificar o valor de negócio de cada tarefa, devem-se definir os objetivos do negócio que se deseja analisar. Então, para cada objetivo definido divide-se o valor total 100 entre as tarefas do Software cujos valores de negócio estão sendo analisados. Por exemplo, a Tabela-1 mostra que, para o objetivo do negócio "Valor de Mercado", a tarefa "Cadastro de Clientes" possui o valor de negócio 15, a tarefa "Cadastro de carros" possui o valor 15, a tarefa "Cadastro de Aluguel" possui o valor 35 e a tarefa "Cadastro de Reserva" possui o valor 35. Se somarmos o valor de negócio de todas essas tarefas, o resultado obtido é 100, que corresponde ao valor total do valor de negócio atribuído para o objetivo do negócio "Valor de Mercado".

A soma dos valores encontrados para todos os objetivos de negócio de uma dada tarefa corresponde ao valor total

do valor de negócio da tarefa. Para a tarefa "Cadastro de Clientes", os valores dos objetivos: Valor de Mercado, e Importância das informações é, respectivamente 15 e 30, e a soma desses é 45. Portanto, o valor de negócio da tarefa "Cadastro de Cliente" é 45.

Uma vez calculado o valor de negócio de cada tarefa, esses valores podem ser comparados para verificar quais tarefas possuem valores de negócio mais altos ou mais baixos e estimar a importância de cada tarefa para a organização. Esse resultado é utilizado no PPR-U para estabelecer a prioridade de execução da Reengenharia de cada tarefa do Software Legado e, resumidamente, essa atividade pode ser dividida nos seguintes passos:

- Passo-1. Definir os objetivos de negócio.
- Passo-2. Quantificação dos objetivos de negócio das tarefas e do Software selecionados no passo anterior.

Fase-2: Análise de Portfolio

Esta fase determina a ordem que as tarefas do Software Legado serão modificados através da Reengenharia, de acordo com os seus respectivos valores de negócios e qualidade técnica. Nessa fase, as seguintes atividades devem ser realizadas:

Atividade-1: Elaboração do gráfico "Qualidade técnica x Valor de negócio"

Para elaborar esse gráfico o coeficiente da qualidade técnica e o coeficiente do valor de negócio devem ser calculados. O coeficiente da qualidade técnica é calculado estimando-se o limite inferior e superior de cada métrica, bem como se medindo o valor da métrica do Software Legado. Dessa maneira, calcula-se o coeficiente da qualidade técnica do Software. O coeficiente do valor de negócio, por sua vez, é calculado através das informações proveniente da *Fase-1* e o gráfico "Qualidade técnica x valor de negócio" é construído. Essa atividade pode ser realizada aplicando os seguintes passos:

- Passo-1: Analisar as métricas de Usabilidade obtidas na Fase 1 (Justificativa do projeto de Reengenharia) e determinar os limites inferior e superior de cada métrica de Usabilidade.
- Passo-2: Para cada métrica de Usabilidade, calcular o coeficiente da qualidade técnica através da seguinte fórmula:

$$CQT = 1 - \frac{(\text{Medido} - \text{Inferior})}{(\text{Superior} - \text{Inferior})}$$

Onde:

- **CQT:** Coeficiente da Qualidade Técnica.
- **Medido:** Valor medido da métrica em questão.
- **Inferior:** Limite Inferior da métrica em questão.
- **Superior:** Limite Superior da métrica em questão.

O coeficiente da qualidade técnica representa o quão boa está a medida da métrica do software Legado em relação aos limites inferior e superior definidos.

- Passo-3: Para cada valor de negócio calcular todos os coeficientes do valor de negócio através da seguinte fórmula:

$$CVN = \frac{VNT}{Total}$$

Onde:

- **CVN:** Coeficiente do Valor de Negócio.
- **VNT:** Valor de Negócio da Tarefa.
- **Total:** Soma Total dos Valores de Negócio do software.

- Passo-4: Para cada métrica, elaborar o gráfico "Qualidade técnica x Valor de negócio" dos valores obtidos nas tarefas do software, como mostrado na Figura-1. Considerando que o valor do CQT e do CVN está entre 0 e 1, as tarefas que se situam no quadrante inferior esquerdo (CQT = [0 ; 0,5] e CVN = [0 ; 0,5]) possuem uma baixa qualidade técnica e um baixo valor de negócio, essas tarefas são candidatas a um Re-desenvolvimento, já as tarefas que se situam no quadrante inferior direito (CQT = [0 ; 0,5] e CVN = [0,5 ; 1]) possuem uma baixa qualidade técnica com um alto valor de negócio, portanto apresentam uma alta prioridade para Reengenharia.

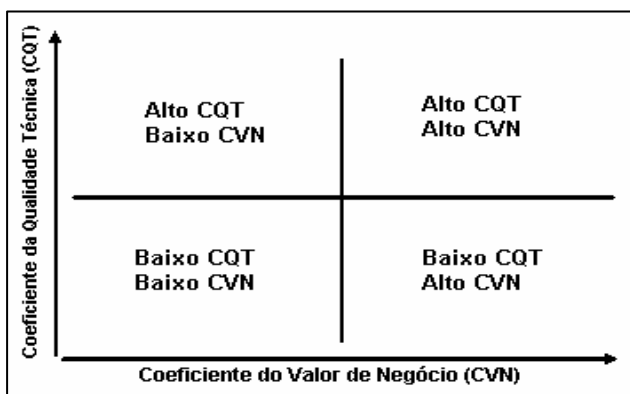


Figura-1. Exemplo do Gráfico “Qualidade Técnica x Valor de Negócio”

Além dos gráficos de cada métrica, também se deve calcular a média de todos os coeficientes de qualidade técnica obtidos e elaborar o gráfico global utilizando a média dos coeficientes de qualidade técnica e os valores de negócios totais de cada tarefa.

Atividade-2: Priorização da Reengenharia dos módulos do Software Legado

Essa atividade é realizada através da análise das informações contidas nos gráficos de Qualidade Técnica x Valor de negócio. Assim, para as tarefas com qualidade técnica mais baixa e maior valor de negócio, maior a prioridade atribuída. Devem-se classificar as tarefas em uma lista ordenada de tarefas mostrando a prioridade de cada uma sofrer Reengenharia. Por exemplo, considere a tabela a seguir:

Tabela - 2. Exemplo de tabela CQT x VN

Qualidade Técnica	Valor de negócio
0,5	0,9
0,4	0,5
0,8	0,7
0,6	0,6
0,8	0,7

Por fim, ordenando a Tabela-2 pelo menor valor da Qualidade Técnica e pelo maior valor do Valor de Negócio para as linhas de Qualidade Técnica iguais, tem-se a Tabela-3 com as respectivas prioridades para cada uma das tarefas:

Tabela-3: Exemplo de tabela CQT x VN ordenada pelo maior valor de VN

Prioridade	Qualidade Técnica	Valor de negócio
1	0,4	0,5
2	0,5	0,9
3	0,6	0,6
4	0,8	0,9
5	0,8	0,7

Essa atividade é realizada aplicando os seguintes passos:

- Passo-1: Para cada tarefa, preencher a tabela “Coeficiente da Qualidade Técnica (CQT) x Valor de Negócio (VN) - "CQT x VN"”.
- Passo-2: Ordenar a tabela "CQT x VN" pelo menor valor da qualidade técnica do Software.
- Passo-3: Ordenar as linhas da tabela (CQT x VN) que contiverem valores iguais de CQT pelo maior valor de negócio presente na mesma.

Fase-3: Estimativa do custo

Essa fase determina o custo monetário, o esforço necessário e o cronograma da Reengenharia. Para tanto as seguintes atividades devem ser executadas:

Atividade-1: Estimativa do esforço

Deve-se calcular o esforço para fazer a Reengenharia de cada item do Software que será modificado. Para tanto, deve-se encontrar a relação "linhas de código do Software" pelo esforço (dado em pessoas-mês) realizado na Reengenharia de outro Software com características similares ao Software que sofrerá Reengenharia. Outro item importante que entra na contabilidade dos custos de um projeto de Reengenharia é o custo referente aos testes realizados nesse projeto. Esse custo é calculado através da multiplicação do número de casos de testes encontrados para a Reengenharia do Software pelo custo médio de um caso de teste. A Reengenharia dos dados do Software também deve ter seus custos contabilizados. Essa Reengenharia envolve bases de dados, arquivos e campos de tabelas das bases de dados. Para calcular o custo desse item do Software deve-se ponderar o número de estruturas de dados (número de bases de dados, de arquivos e de campos) e dividir esse valor pela produtividade mensal (dado em pessoas-mês). Por fim, deve-se gerar uma tabela de produtividade que converta o número de cada item do Software em esforço dado em pessoas-dia. Com o cálculo dos custos de todos os itens do Software em questão, contabiliza-se o custo total da Reengenharia somando-se o valor dos custos desses itens. Através dos valores de esforço encontrados no cálculo do custo do projeto de Reengenharia, elabora-se o cronograma do projeto de Reengenharia contendo o tempo estimado para realizar a Reengenharia dos itens do Software analisados anteriormente.

Essa atividade é realizada através dos seguintes passos:

- Passo-1: Listar os itens que deverão sofrer Reengenharia e terão que ser contabilizados no custo total da Reengenharia (funcionalidades, banco de

dados, arquivos, testes para a Reengenharia, telas que deverão ser modificadas, entre outros).

- Passo-2: Estimar o esforço realizado através de pontos-função ou LOC do software Legado para cada item levantado no passo anterior. O esforço médio de Reengenharia deve ser obtido através da base de dados históricos da organização onde está sendo realizada a Reengenharia.

Atividade-2: Elaborar o cronograma da Reengenharia

Inicialmente deve-se estimar o tempo mínimo da Reengenharia usando uma variante do modelo COCOMO proposto por Sneed. Essa variante considera o fato da Reengenharia utilizar os documentos do Software Legado. No modelo original, o Tempo Mínimo é estimado pela seguinte fórmula:

$$\text{Tempo Mínimo} = 2.5 \times (\text{Esforço})^{0.38}$$

A variante proposta por Sneed contém o expoente com valor 0.19 devido à utilização dos documentos Legados na Reengenharia do Software. Após o cálculo do tempo mínimo para a Reengenharia, deve-se elaborar o seu cronograma utilizando o esforço calculado na *Atividade-1*.

Logo, essa atividade é realizada com os seguintes passos:

- Passo-1: Estimar o tempo mínimo da Reengenharia usando COCOMO.
- Passo-2: Dividir as tarefas entre os desenvolvedores e elaborar o cronograma da Reengenharia.

Fase-4 Análise custo-benefício

Essa Fase destina-se estudar a melhor maneira de se modificar o Software: através da própria manutenção habitual do Software Legado; através do Re-desenvolvimento do Software Legado ou através da Reengenharia do Software.

Para tanto, ela estuda:

1. O fator custo-benefício do Software Legado sem realizar nenhuma modificação;
2. O fator custo-benefício do Software após a sua Reengenharia;
3. O fator custo-benefício do Software após o Re-desenvolvimento do mesmo.

Esses fatores são explicados a seguir:

Atividade-1: Cálculo do custo-benefício do Software Legado

Nessa atividade deve-se calcular o custo-benefício do Software Legado. Para tanto, deve-se obter o custo de manutenção, o valor de negócio, o custo operacional, através de dados históricos provenientes das organizações envolvidas, e, finalmente, calcular o custo-benefício do Software Legado. Os seguintes passos são realizados nessa atividade:

- Passo-1: Obtenção do custo de manutenção do Software Legado.
- Passo-2: Obtenção do valor de negócio do Software Legado.
- Passo-3: Obtenção do custo operacional do Software Legado.
- Passo-4: Cálculo do benefício do Software Legado através da seguinte fórmula:

$$BAS = [P1 - (P2 + P3)] * P4$$

Onde:

- **BAS**: Benefício Atual do Software.
- **P1**: Valor de negócio do Software Legado (Passo-2).
- **P2**: Custo de manutenção do Software Legado (Passo-1).
- **P3**: Custo operacional do Software Legado (Passo-3).
- **P4**: Tempo de vida esperado do sistema atual (estimado pelo Planejador).

Atividade-2: Cálculo do custo-benefício do Software produzido pela Reengenharia

Com o intuito de estimar o benefício alcançado pela aplicação da Reengenharia, deve-se construir o novo modelo GOMS de cada tarefa que está sendo analisada após a mesma ter sofrido a Reengenharia. Nessa Atividade, deve-se construir o modelo GOMS com base no modelo construído a partir do Legado. Dessa forma, possivelmente obtém-se um modelo GOMS diferente do modelo GOMS do Software Legado. Através do novo modelo GOMS, estima-se o tempo de execução que o usuário realizará. Conseqüentemente, os tempos de execução presentes no modelo GOMS obtido pela análise da tarefa como ela ocorre na interface atual e no modelo GOMS obtido pelo modelo mental do usuário são comparados e a diferença entre esses corresponde ao benefício que a Reengenharia proporcionará ao Software. Para se calcular o benefício produzido pela Reengenharia do software Legado, os seguintes passos são executados:

- Passo-1: Obtenção do esforço de manutenção após a Reengenharia.
- Passo-2: Obtenção do custo operacional após a Reengenharia.
- Passo-3: Obtenção do valor de negócio após a Reengenharia.
- Passo-4: Obtenção do esforço da Reengenharia.
- Passo-5: Obtenção do custo da Reengenharia.
- Passo-6: Obtenção do fator de risco da Reengenharia.
- Passo-7: Cálculo do benefício obtido após a Reengenharia através da seguinte fórmula:

$$BRS = [(P5 - (P6 + P7)) * (P4 - P10) - (P8 * P9)] - BAS$$

Onde:

- **BRS**: Benefício obtido pela Reengenharia do Software.
- **P5**: Valor de negócio anual estimado do sistema depois da Reengenharia.
- **P6**: Custo anual estimado de manutenção do sistema depois da Reengenharia.
- **P7**: Custo anual de operação do sistema depois da Reengenharia.
- **P8**: Custo estimado da Reengenharia do software.
- **P9**: Fator de risco da Reengenharia.
- **P10**: Tempo estimado para a Reengenharia do software.

Atividade-3: Cálculo do custo-benefício do Software produzido pelo Re-desenvolvimento

Com o intuito de estimar o benefício alcançado pela aplicação do Re-desenvolvimento, deve-se construir o novo modelo GOMS de cada tarefa que está sendo analisada após a mesma ter sofrido o Re-desenvolvimento. Nessa Atividade, deve-se construir o modelo GOMS diferenciando-se do elaborado na Atividade-2 por não se basear no modelo gerado a partir do Legado, e sim a partir do modelo mental dos usuários. Dessa forma, possivelmente obtém-se um modelo GOMS diferente do modelo GOMS do Software Legado. Através do novo modelo GOMS, estima-se o tempo de execução que o usuário realizará. Conseqüentemente, os tempos de execução presentes no modelo GOMS obtido pela análise da tarefa como ela ocorre na interface atual e no modelo GOMS obtido pelo modelo mental do usuário são comparados e a diferença entre esses corresponde ao benefício que o Re-desenvolvimento proporcionará ao Software.

Para se calcular o benefício produzido pelo Re-desenvolvimento do software Legado, os seguintes passos são executados:

- Passo-1: Obtenção do esforço de manutenção após o Re-desenvolvimento.
- Passo-2: Obtenção do custo operacional após o Re-desenvolvimento.
- Passo-3: Obtenção do valor de negócio após o Re-desenvolvimento.
- Passo-4: Obtenção do esforço do Re-desenvolvimento.
- Passo-5: Obtenção do custo do Re-desenvolvimento.
- Passo-6: Obtenção do fator de risco do Re-desenvolvimento.
- Passo-7: Cálculo do benefício obtido após o Re-desenvolvimento através da fórmula utilizada na atividade anterior utilizando os dados instanciados para o Re-desenvolvimento:

Fase-5 Elaboração do contrato

Esta fase produz um documento especificando a qualidade técnica esperada do Software obtido na Reengenharia, o cronograma da Reengenharia, o seu custo e o esforço necessário. Nesta fase, a seguinte atividade deve ser realizada:

Atividade-1: Elaboração do contrato

Essa atividade consiste de um único passo que é relatar as informações obtidas nas fases anteriores.

- Passo-1: Relatar as informações obtidas nas Fases 1 à 4. As seguintes tarefas devem ser realizadas: listagem dos problemas e respectivos locais, sugestão de mudanças, priorização das tarefas de Reengenharia, análise custo-benefício, tarefas da Reengenharia.

ESTUDO DE CASO DO PPR-U

Essa seção apresenta o experimento realizado para verificar a viabilidade do Processo de Planejamento de Reengenharia de Software guiado pela Avaliação de Usabilidade.

Para a realização do experimento escolheu-se um Software desenvolvido por alunos de mestrado da Universidade Federal de São Carlos com o intuito de

planejar a sua Reengenharia. Porém, para a viabilização do experimento e diante da impossibilidade de se obter dados confidenciais de empresas, criou-se um cenário de testes para simular a situação vivida por uma empresa de locação de veículos. Assim, foi descrito um cenário que abrangeu duas visões:

- Visão da locadora – essa visão destinou-se a:
 1. Os custos operacionais do Software, ou seja, quanto à locadora gasta com as despesas de salários e infra-estrutura para operar o Software;
 2. O volume de alugueis realizados. Para isso, estimou-se a quantidade de alugueis efetuados diariamente, a quantidade média de dias de um aluguel e o preço médio dos carros por categoria;
 3. As categorias e os carros existentes e a quantidade média de alugueis por categoria.
- Visão do desenvolvedor do Software:
 1. Esforço de manutenção do Software;
 2. Esforço para Reengenharia do Software;
 3. Custo para manter o Software Legado;
 4. Custo para realizar a Reengenharia do Software;
 5. Custo por desenvolvedor.

Diante desses cenários, o experimento foi realizado com o PPR-U aplicado às tarefas: cadastrar categoria de carro, cadastrar carro, cadastrar cliente, alugar carro e devolver carro. O experimento dividiu-se em três etapas: a primeira foi o Planejamento da Reengenharia, a segunda foi a Reengenharia do Software Legado e a terceira a Avaliação de Usabilidade do Software obtido pela Reengenharia.

Planejamento da Reengenharia usando o PPR-U

A seguir os resultados obtidos por cada Fase são mostrados a seguir:

Fase-1: Justificativa da Reengenharia

Na Tabela-4, os resultados obtidos pela aplicação da Avaliação Heurística, do Percorso Cognitivo e da elaboração do Modelo GOMS são apresentados:

Tabela-4: Resultados obtidos pelos métodos de inspeção e pelo Modelo GOMS no PPR-U

Tarefas	Tempo de execução estimado pelo Modelo GOMS (segundos)	Quantidade de problemas detectados pela Avaliação Heurística	Quantidade de problemas detectados pelo Percorso Cognitivo
Cadastrar categoria de carros	Máximo: 31.1 Mínimo: 27.26	10	2
Cadastrar carros	Máximo: 31.84 Mínimo: 28.8	7	2
Cadastrar clientes	Máximo: 105.64 Mínimo: 87.96	6	2
Alugar carro	Máximo: 78.36 Mínimo: 70.76	17	6
Devolver carro	Máximo: 67.96 Mínimo: 60.12	19	5

Após a aplicação dos métodos de inspeção e da elaboração do Modelo GOMS, foi realizada uma análise

para determinar quais métricas de Usabilidade seriam relevantes para o Planejamento da Reengenharia do software em questão. Na Tabela-5 são apresentadas as métricas escolhidas e a justificativa para tais escolhas.

Tabela-5: Justificativa das métricas de Usabilidade escolhidas.

Métrica de Usabilidade	Justificativa
Tempo para aprender uma tarefa.	Os resultados do Percurso Cognitivo mostram uma dificuldade no aprendizado do software.
Tempo gasto para executar a tarefa.	O Modelo GOMS mostra que o fluxo de ações pode ser melhorado.
Número de erros na aprendizagem.	A heurística prevenção ao erro consiste na terceira heurística mais violada. Portanto, essa métrica quantifica essa constatação.
Número de erros após a aprendizagem.	A heurística prevenção ao erro consiste na terceira heurística mais violada. Portanto, essa métrica quantifica essa constatação.
Porcentagem de usuários que não completaram a tarefa.	O Percurso Cognitivo mostra que existem pontos da interação onde o software dificulta a sua aprendizagem.

As métricas definidas na Tabela-5 foram, então, quantificadas para o software Legado através da aplicação do Teste de Usabilidade. Esse Teste de Usabilidade teve a participação de cinco participantes e foram utilizadas técnicas de observação como *Think Aloud*, registro de áudio através de software de gravação e ferramenta para registro dos movimentos realizados pelos usuários na interação com o Software. O Teste foi planejado usando as métricas definidas anteriormente e a norma CIF [1] foi utilizada para relatar tanto o Planejamento quanto o resultado do teste.

Com isso, a análise da qualidade técnica do software foi realizada e foi estimado o esforço de manutenção do software Legado através dos dados provenientes dos cenários criados. Os objetivos do negócio foram definidos e estimados, conforme apresentados na Tabela-6.

Tabela-6: Valor de negócio das funções do software.

Tarefa	Valor de Mercado	Contribuição para o lucro	Importância das informações	Pontuação
Cadastro de clientes	15	15	30	60
Cadastro de carros	15	15	20	50
Cadastro de Aluguel	35	50	35	120
Cadastro de Reserva	35	20	15	70
Total	100	100	100	300

Fase-2: Análise de Portfolio

De posse dos valores obtidos das métricas de Usabilidade pelo Teste de Usabilidade, pela determinação dos limites inferior e superior de cada métrica e dos valores de negócio, foi possível elaborar o gráfico “Qualidade Técnica x Valor de Negócio” que tem o intuito de auxiliar no Processo de priorização da Reengenharia das tarefas analisadas. Como exemplo de cálculo de coeficiente de qualidade técnica, a Tabela-7 é apresentada:

Tabela-7: Coeficiente da qualidade técnica da funcionalidade Cadastrar Cliente.

Métrica	Média	Limite Inferior	Limite Superior	Coeficiente de qualidade técnica
Tempo gasto durante a aprendizagem da tarefa	138.4”	105.64”	316.92”	0.84
Porcentagem de usuários que completaram a tarefa	0%	0%	40%	1

O coeficiente do valor de negócio, por sua vez, foi calculado através da divisão do valor de negócio de uma dada tarefa pela pontuação total. Por exemplo, para a tarefa Cadastro de Clientes (visualizando os dados da Tabela-6) deve-se dividir o valor da Pontuação (60) pelo valor total do valor de negócio (300), obtendo-se o coeficiente de valor de negócio 0,2. Então, analisando os coeficientes, prioriza-se as funcionalidades que deverão sofrer Reengenharia. Essa análise resultou na seguinte ordenação de prioridades para a Reengenharia das funcionalidades do software:

1. Aluguel de carros;
2. Devolução de carros;
3. Cadastro de clientes;
4. Cadastro de categoria de carros;
5. Cadastro de carros.

Fase-3: Estimativa do custo

A estimativa de custo foi dividida em duas etapas: a estimativa do esforço para Reengenharia das interfaces do Software e a estimativa do esforço para Reengenharia dos itens do Software relativos à sua funcionalidade. A Figura-2 apresenta o cronograma da Reengenharia obtido pela execução dessa fase:

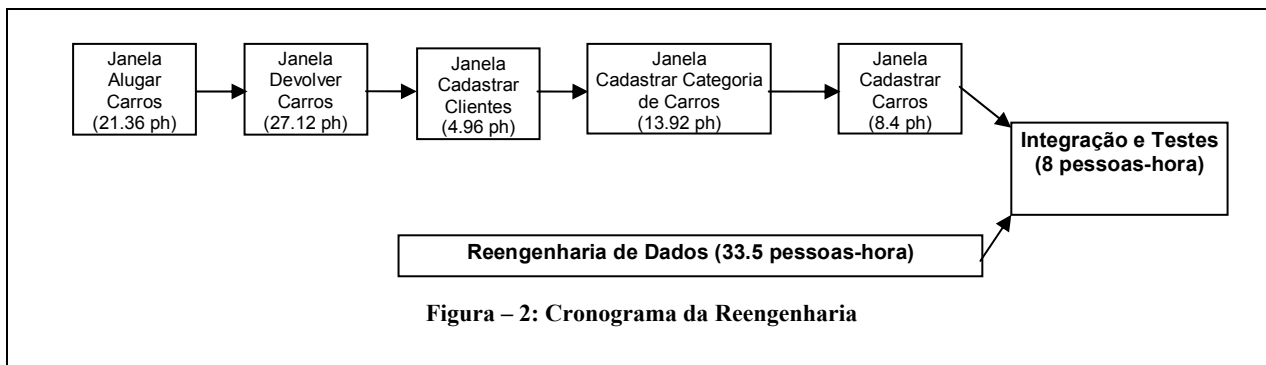


Figura – 2: Cronograma da Reengenharia

Fase-4: Análise custo-benefício

Estimativa do custo-benefício do Software Legado: nessa atividade, obteve-se o custo de manutenção do Software Legado através da multiplicação do esforço de manutenção obtido no cenário da visão do desenvolvedor (112 pessoas-hora) descrito anteriormente pelo valor da hora pago para o desenvolvedor (R\$ 15,00), resultando em R\$ 1680,00. Logo a seguir, o valor de negócio do Software Legado foi calculado da seguinte maneira. Pelo Modelo GOMS das tarefas do Software têm-se os tempos de execução para cada tarefa na Tabela-8:

Tabela-8: Estimativas de tempo de execução antes da Reengenharia obtidos pelo Modelo GOMS

Tarefa	Tempo de execução antes da Reengenharia (seg)
Cadastro de Categoria de Carros	31.1
Cadastro de Carros	31.84
Cadastro de Clientes	105.64
Aluguel de Carros	78.36
Devolução de Carros	67.96

Considerando os seguintes dados sobre o trabalho do funcionário que utiliza o Software:

- A jornada de trabalho do funcionário é 160 horas mensais.
- 70% do tempo o funcionário realiza Aluguel que, em média, consiste nas tarefas de cadastrar novo cliente e de registrar aluguel. Considerando a jornada de 160 horas, o funcionário realiza Aluguel 112 horas mensais.
- 20% do tempo o funcionário registra a Devolução de Carro. Considerando a jornada de 160 horas, o funcionário registra Devolução 32 horas mensais.
- 8% do tempo o funcionário cadastra um novo carro. Considerando a jornada de 160 horas, o funcionário cadastra um novo carro 12.8 horas mensais.
- 2% do tempo o funcionário cadastra uma nova categoria de carro. Uma vez que a jornada é 160 horas, o funcionário cadastra categoria de carro 3.2 horas mensais. Portanto, o Número de alugueis (NALug) possíveis de serem realizados é calculado através do total de horas que o funcionário registra aluguel dividido pelo tempo gasto na execução das tarefas Cadastrar Cliente e Registrar Aluguel (já que, como visto, em média para registrar um aluguel o funcionário cadastra cliente e registra o aluguel). Além disso, devem-se considerar os tempos para atendimento ao cliente, espera por um novo cliente e

outras situações de espera que possam ocorrer, dessa forma, considera-se que o tempo de um aluguel seja o triplo do tempo gasto na execução das tarefas Cadastrar Cliente e Registrar Aluguel. Logo, o número de alugueis realizados mensalmente é:

$$NALug = ((112 \text{ horas mensais} / (105.64 + 78.36)) \times 3600) / 3 = 730 \text{ alugueis}$$

Considerando os seguintes dados estatísticos de locação:

- Média de dias por locação = 4
- Média de locações por dia (por categoria)
 - 70 % Popular: corresponde 511 carros mensais.
 - 25% Sedan: corresponde 182 carros mensais.
 - 5% Importados: corresponde 37 carros mensais.
- Preço médio da diária por categoria:
 - Popular: R\$ 50
 - Sedan: R\$ 100
 - Importado: R\$ 150

A renda obtida por categoria considerando os dados citados a cima corresponde a:

- Popular: R\$ 50 x 511 Carros x 4 dias = R\$ 102200,00
- Sedan: R\$ 100 x 182 Carros x 4 dias = R\$ 72800,00
- Importados: R\$ 150 x 37 Carros x 4 dias = R\$ 22200,00

Portanto, a renda total ou o valor de negócio (VN) do Software Locadora de Carros Ubirajara corresponde a:

$$VN = R\$ 102200,00 + R\$ 72800,00 + R\$ 22200,00 = R\$ 197200,00$$

Após o cálculo do valor de negócio do Software Legado, o custo operacional foi obtido através do cenário descrito (R\$ 900) e, por fim, o custo-benefício foi calculado como a seguir:

$$\begin{aligned} \text{Benefício do Software Legado mensal} &= \text{Valor de negócio (VN)} - \\ &= (\text{Custo da manutenção (CM)} + \text{Custo operacional (CO)}) = 197200,00 \\ &\quad - (1680 + 900) = R\$ 194620 \text{ mensais} \\ \text{Benefício do Software Legado anual} &= R\$ 194620 \times 12 = R\$ \\ &\quad 2335440,00 \end{aligned}$$

A estimativa do custo-benefício após a Reengenharia foi feita da seguinte forma: Inicialmente, obteve-se o esforço de manutenção após a Reengenharia através do cenário da visão do desenvolvedor descrito e estimando que o número de problemas torne-se, em média, 1.5 problemas por funcionalidade (antes da Reengenharia esse valor

correspondia a 2 problemas por funcionalidade). Portanto, o esforço de manutenção após a Reengenharia representa um valor de 84 pessoas-hora e o custo de manutenção = 84 pessoas-horas x R\$ 15,00 = R\$ 1260,00.

O segundo passo realizado foi a obtenção do custo operacional após a Reengenharia: O custo operacional (CO) mensal do Software Locadora de Carros Ubirajara é R\$ 900,00 considerando as despesas de salário do funcionário e infra-estrutura necessária.

O terceiro passo realizado foi obter o valor de negócio do Software após a Reengenharia. Tal informação foi obtida pela construção do novo modelo GOMS das tarefas remodeladas.

Através do novo modelo GOMS construído, os novos parâmetros (após a Reengenharia) são mostrados na Tabela-9:

Tabela-9: Tempo de execução após a Reengenharia

Tarefa	Tempo de execução após a Reengenharia (seg)
Cadastro de Categoria de Carros	27.74
Cadastro de Carros	30.72
Cadastro de Clientes	99.78
Aluguel de Carros	54.96
Devolução de Carros	36.56

Considerando os seguintes dados sobre o trabalho do funcionário que utiliza o Software:

- Jornada de trabalho do funcionário é 160 horas mensais.
- 70% do tempo o funcionário realiza Aluguel que, em média, consiste nas tarefas de cadastrar novo cliente e de registrar aluguel. Portanto, o funcionário realiza Aluguel 112 horas mensais.
- 20% do tempo o funcionário registra a Devolução de Carro. Considerando a jornada de 160 horas, o funcionário registra Devolução 32 horas mensais.
- 8% do tempo o funcionário cadastra um novo carro. Considerando a jornada de 160 horas, o funcionário cadastra um novo carro 12.8 horas mensais.
- 2% do tempo o funcionário cadastra uma nova categoria. Portanto, o funcionário cadastra uma nova categoria 3.2 horas mensais.

Portanto, o Número de aluguéis (NAlug) possíveis é calculado através do total de horas que o funcionário registra aluguel dividido pelo tempo gasto na execução das tarefas Cadastrar Cliente e Registrar Aluguel (já que, como visto, em média para registrar um aluguel o funcionário cadastra cliente e registra o aluguel).

Além disso, devem-se considerar os tempos para atendimento ao cliente, espera por um novo cliente e outras situações de espera que possam ocorrer, conseqüentemente, considera-se que o tempo de um aluguel seja o triplo do tempo gasto na execução das tarefas Cadastrar Cliente e Registrar Aluguel.

Logo, o número de aluguéis realizados mensalmente é:

$$NAlug = ((112 \text{ horas mensais} / (99.78 + 54.96)) \times 3600) / 3 = 868 \text{ aluguéis}$$

Considerando os seguintes dados estatísticos de locação:

- Média de dias de locação = 4
- Média de locações por dia (por categoria)
 - 70 % Popular: corresponde 607 carros mensais.
 - 25% Sedan: corresponde 217 carros mensais.
 - 5% Importados: corresponde 44 carros mensais.

A renda obtida por categoria é:

- Popular: R\$ 50 x 607 x 4 = R\$ 121400,00
- Sedan: R\$ 100 x 217 x 4 = R\$ 86800,00
- Importados: R\$ 150 x 44 x 4 = R\$ 26400,00

A renda total ou o valor de negócio (VN) do Software Locadora de Carros Ubirajara corresponde a:

$$VN = R\$ 121400,00 + R\$ 86800,00 + R\$ 26400,00 = R\$ 234600,00$$

O quarto passo foi a obtenção do esforço da Reengenharia através das informações obtidas pela Fase-3:

Esforço Total da Reengenharia = 117.26 pessoas-hora

O quinto passo realizado foi a obtenção do custo da Reengenharia através das informações obtidas na Fase-3:

$$\begin{aligned} \text{Custo da Reengenharia} &= \text{Esforço Total da} \\ &\text{Reengenharia} \times \text{Custo da Reengenharia/hora} \\ \text{Custo da Reengenharia} &= 117.26 \text{ pessoas-hora} \times R\$ \\ &15,00 = R\$ 1758,90 \end{aligned}$$

Em seguida, foi obtido o fator de risco da Reengenharia pelo cenário descrito (1.0). E, por fim, o benefício após a Reengenharia foi calculado.

$$\text{Benefício} = [(234600 \times 12 - (1260 \times 12 + 900 \times 12)) \times (1 - 8.65E-3) - (1758,90 \times 1,0)] - 2335440 = 427953,828$$

O custo-benefício do Re-desenvolvimento foi calculado seguindo os mesmos passos da estimativa do custo-benefício da Reengenharia, porém, os valores para custo de manutenção e fator de risco modificaram-se devido ao Software ter de ser novamente desenvolvido sem ter um alicerce já pronto, como no caso da Reengenharia. O benefício obtido foi de R\$ 421096,70. Nesse experimento, a Reengenharia apresentou-se a opção mais vantajosa.

Fase-5: Elaboração do Contrato

O contrato elaborado conteve os seguintes documentos: lista de problemas encontrados, sugestão para mudanças, priorização das tarefas de Reengenharia, análise custo-benefício e tarefas da Reengenharia.

Reengenharia do software

A Reengenharia de software consiste nas seguinte fases: *Engenharia Reversa* + Δ + *Engenharia Avante*. Nesse experimento, a Engenharia Reversa das interfaces foi realizada através do modelo GOMS das tarefas analisadas (cadastrar categoria, cadastrar carro, cadastrar cliente, registrar aluguel e devolver carros). Através do modelo GOMS do software Legado foi possível obter a especificação da interação humano-computador de cada tarefa. Esse fato levou a considerar o modelo GOMS como uma ferramenta bastante interessante para realizar a

Engenharia Reversa de uma interface. A fase Δ foi realizada através da modificação do modelo GOMS original de maneira a deixá-lo mais eficiente e diminuindo o número de operadores existentes. Além disso, na fase Δ também foi levada em consideração os problemas de Usabilidade encontrados na Avaliação Heurística e no Percurso Cognitivo. Através desses problemas, um conjunto de modificações foi sugerido ao novo modelo GOMS obtido para cada tarefa. E, na Engenharia Avante o novo modelo GOMS foi implementado.

Avaliação de Usabilidade do software após a Reengenharia

Para avaliar a Usabilidade após a Reengenharia foi elaborado o Modelo GOMS do software gerado e aplicada a Avaliação Heurística. O tempo de execução para a tarefa Alugar Carro, estimado pelo Modelo GOMS após a Reengenharia caiu 29,8% (de 78.36 segundos para 54.96 segundos) em relação ao tempo antes da Reengenharia. Já dos 18 problemas encontrados anteriormente, apenas dois permaneceram. Observa-se que esses 2 problemas não foram solucionados devido ao não conhecimento do desenvolvedor da linguagem de programação em questão para solucionar tais problemas. Logo, a Usabilidade melhorou, pois houve diminuição do tempo de execução e diminuição no número de problemas encontrados.

CONCLUSÕES

Através deste trabalho, a viabilidade técnica da aplicação de métodos de Avaliação de Usabilidade no Processo de Planejamento da Reengenharia pôde ser verificada, pois a Usabilidade do software gerado pela Reengenharia

melhorou, de acordo com o estudo de caso realizado. O PPR-U estima o fator custo-benefício da manutenção do Software Legado, do Re-desenvolvimento e da Reengenharia e, com esses dados é possível decidir a melhor maneira de se realizar modificações no Software. Mais estudos estão sendo realizados para confirmar os resultados obtidos até agora. Aplicar o PPR-U mostra-se uma alternativa apropriada para a modificação do software através de um Processo de Reengenharia, pois o método permite diagnosticar os problemas de Usabilidade e produzir um software com melhor qualidade de uso.

REFERÊNCIA BIBLIOGRÁFICA

1. Common Industry Format for Usability Test Reports – CIF – ANSI NCITS 354-2001.
2. Kieras, David. (1999). A Guide to GOMS Model Usability Evaluation using GOMS. Univ. Michigan.
3. Lewis, C.; Wharton, C. (1997) Cognitive walkthroughs. In M. Helander, T.K. Landauer, P. Prabhu (Eds.) *Handbook of human-computer interaction, 2nd ed.* Elsevier Science. pp. 717-732.
4. Nielsen, Jacob. How To Conduct a Heuristic Evaluation. May, 2002 - 10/10/2002.
5. Ransom, Jane; Sommerville, Ian; Warren, Ian. (1998). *A method for assessing legacy systems for evolution.* Proc. Conf. on Software Reengineering. IEEE Press.
6. Sneed, Harry M. Planning the Reengineering of Legacy Systems. (1995). IEEE Software. January, 1995. pp. 24-34.
7. Tilley, S. (1995). Perspectives on Legacy System Reengineering. *Software Engineering Institute*, Carnegie Mellon University.

Investigando Aspectos de Interação em Aplicações SIG na Web voltadas ao Domínio Agrícola

Juliano Schimiguel M. Cecília C. Baranauskas Claudia Bauzer Medeiros

Instituto de Computação
UNICAMP
Av. Albert Einstein 1251, CP 6176
{juliano.schimiguel,cecilia.cmbm}@ic.unicamp.br
+55-0xx19-3788-5870

RESUMO

Sistemas de Informação Geográfica (SIG) permitem a manipulação, gerenciamento e visualização de dados geo-referenciados. O interesse por aplicações SIG tem aumentado bastante nos últimos anos. Atualmente, pode-se falar das aplicações SIG Web, onde informação geográfica dispersa em diferentes locais e a sua manipulação via SIG é disponibilizada através da Internet. Existem inúmeras categorias de aplicações SIG, em diferentes escalas e domínios, abrangendo desde temas urbanos até ambientais. A importância de SIG Web direcionados a sistemas agrícolas advém do fato de funcionarem como um ferramental útil para usuários que trabalham direta ou indiretamente no domínio: agricultores, cooperativas agrícolas, órgãos governamentais ligados à área. Dada a importância estratégica desses sistemas e o amplo espectro de perfis de usuários, este trabalho realiza um levantamento de aplicações SIG Web com ênfase no domínio agrícola e investiga aspectos de interação usuário-sistema nessas aplicações.

ABSTRACT

Geographical Information Systems (GIS) allow the manipulation, management and visualization of geo-referenced data. The interest GIS applications has increased in the last years. Currently, Web GIS applications make available through the Internet geographic information dispersed in different places. There are several categories of GIS applications, in different scales and application domains, ranging from urban application to environmental problems. The importance of Web GIS for the agricultural domain comes from the fact that they function as a useful tool for users who work direct or indirectly in the agricultural domain: agriculturists, cooperatives, government instances. Considering the strategic value of these systems and the wide range of different prospective users, this work presents a survey of Web GIS applications with emphasis in the agricultural domain, and investigates user-system interaction aspects in these applications.

VI Simpósio sobre Fatores Humanos em Sistemas Computacionais — Mediando e Transformando o Cotidiano. Hotel Bourbon, Curitiba, 17 a 20 de outubro de 2004. UFPR, CEIHC—SBC.

Palavras-Chave

Interação Humano-Computador, Sistemas de Informação Geográfica, SIG Web, Sistemas Agrícolas

INTRODUÇÃO

Um Sistema de Informação Geográfica (SIG) é um software voltado para o gerenciamento de dados geo-referenciados. O termo geo-referenciado denota dados que estão explicitamente ligados a coordenadas geográficas. Dados geo-referenciados são definidos a partir de dois componentes: posição (localização sobre a Terra) e atributos não-espaciais (características descritivas); por exemplo, atributos descritivos de um terreno incluem nome do proprietário ou valor venal. A partir do momento em que sua localização é definida usando algum sistema de coordenadas, ele passa a ser um objeto geo-referenciado [30, 31].

Um SIG serve como base a um enorme conjunto de aplicações, chamadas “geográficas”, que são todas aquelas que manipulam dados geo-referenciados e onde a visualização cartográfica é importante. Segundo Oliveira [24], as aplicações usualmente desenvolvidas em SIG definem requisitos de informação que permitem classificá-las em três categorias: urbanas, ambientais e gerenciais. As aplicações urbanas são voltadas para aspectos de infra-estrutura urbana e controle populacional, como gerência de redes (energia, telecomunicações, transportes) e distribuição de serviços públicos. As aplicações ambientais são voltadas para o aproveitamento e conservação de recursos naturais, como a modelagem da natureza (estudos climáticos, controle de agentes poluidores, análise de processos de desertificação). As aplicações gerenciais envolvem informações qualitativas sobre aplicações ambientais e urbanas. O objetivo desta terceira categoria é apoiar a formulação e o acompanhamento de políticas de desenvolvimento urbano e de uso de recursos naturais. Os sistemas agrícolas enquadram-se normalmente na categoria de aplicações ambientais.

Um SIG oferece ao desenvolvedor de aplicação um grande conjunto de funções que permitem processamento espacial dos dados e sua apresentação cartográfica, incluindo frequentemente interação via mapas. Além disso, um SIG utiliza operações de gestão de bases de dados, como consultas espaciais e análise estatística. Estas capacidades distinguem o SIG de outros sistemas de informação e fazem deste uma ferramenta valiosa para

uma vasta gama de organizações e empresas, com o objetivo de explicar eventos, prever resultados e planejar estratégias relativas a aplicações onde a dimensão espacial é importante [30, 31].

O avanço da Tecnologia de Informação (TI), os recursos dos programas voltados para o contexto de SIG e a disseminação da Internet no cotidiano, possibilitaram a criação de "mapas inteligentes" na Internet. O termo "mapa inteligente" é comumente usado por usuários SIG Web para denotar a possibilidade de interagir com um SIG e os bancos de dados subjacentes, via interfaces cartográficas. Dessa maneira, um usuário leigo na área de geoprocessamento pode usufruir destes benefícios tecnológicos com o uso puro e simples de seu *browser* padrão (visualizador Web). Esta é uma revolução na maneira de divulgar e selecionar a informação georeferenciada, que poderá democratizar o acesso dos interessados a esta tecnologia [14].

Neste contexto, definimos um SIG Web como um sistema que pode permitir a visualização e consulta a dados geográficos através da Web, de acordo com dois conceitos: SIG Web propriamente ditos e aplicações SIG Web. Um SIG Web é um sistema de software (comercial ou acadêmico) que permite a criação de aplicações SIG Web. Uma aplicação SIG Web tem por característica permitir disponibilizar visualizações de informação geográfica, podendo possibilitar alguns tipos de interação com mapas, como *zoom*, *pan*, ou consultas diversas. Essas interações são realizadas, podendo, ou não, existir a ligação da aplicação SIG Web com um sistema SIG Web e/ou um banco de dados geográfico em algum servidor SIG. Isto ocorre porque em algumas categorias de aplicações SIG Web, as imagens de mapas são capturadas de sistemas SIG Web e simplesmente colocadas na aplicação SIG Web (usando comandos de linguagens como o Html – Linguagem de Marcação de Hipertexto), configurando-se uma estrutura estática para essas imagens. Como exemplos de SIG Web, pode-se destacar o MapObjects [21] e o MapInfo [20]; e como aplicações SIG Web, o da Embrapa [9] e o da FUNCEME [13].

Dada a relevância estratégica de tais sistemas e aplicações e considerando a diversidade de usuários prospectivos possibilitadas pelo acesso a essas aplicações pela Internet, torna-se importante investigar aspectos de interação possibilitadas pelas interfaces de tais sistemas. Os objetivos deste trabalho envolvem o levantamento de aplicações SIG Web disponíveis no mercado e meio acadêmico voltadas ao domínio agrícola e a categorização das aplicações levantadas e sua análise, de acordo com critérios de conteúdo veiculado e de interação usuário-sistema.

O artigo está organizado da seguinte forma: a primeira seção apresenta uma categorização para aplicações SIG Web. A seção a seguir mostra resultados do levantamento feito para aplicações SIG Web e sua caracterização em relação a conteúdo veiculado e formas de interação possibilitadas. Por fim, discutimos resultados da análise realizada e concluímos o trabalho.

CATEGORIAS DE APLICAÇÕES SIG NA WEB

O volume e a complexidade de conteúdos representados na Web faz com que encontremos inúmeros formatos para organizar e estruturar a informação. A liberdade oferecida pela Web dificulta a atividade de definição de padrões. A W3C - *World Wide Web Consortium* [38] é um exemplo de organização que trabalha em prol do desenvolvimento de tecnologias para uso completo e eficaz do potencial da Web (especificações, *guidelines*, software, ferramentas). Em SIG Web, existe uma diversidade de aplicações, que podem ser categorizadas de acordo com o tipo de conteúdo que apresentam e o grau de interação possibilitado. Chang [5] classifica as aplicações SIG Web em cinco categorias, de acordo com um conjunto de métodos e tecnologias que viabilizem a utilização de SIG, para o público em geral [19].

Fotos de Mapas (Pré-Gerados) (FM)

Do ponto de vista do programador de aplicação, esta categoria representa o caminho mais fácil para disponibilizar a imagem de um mapa na Web. O procedimento para isto é relativamente simples. Primeiro, gera-se um mapa usando um SIG convencional; depois, o mapa é transformado em um arquivo de imagem; por último cria-se uma página HTML para tornar a imagem disponível na Web. Para o usuário, existe a possibilidade de interação com o mapa; entretanto, isso depende dos formatos de imagem que estiverem sendo utilizados (alguns formatos possuem boa resolução para possibilitar interação, é o caso do GIF; já outros não, é o caso do CGM, DWF, DXF). A Figura 1, a seguir, ilustra essa categoria de aplicação SIG Web.

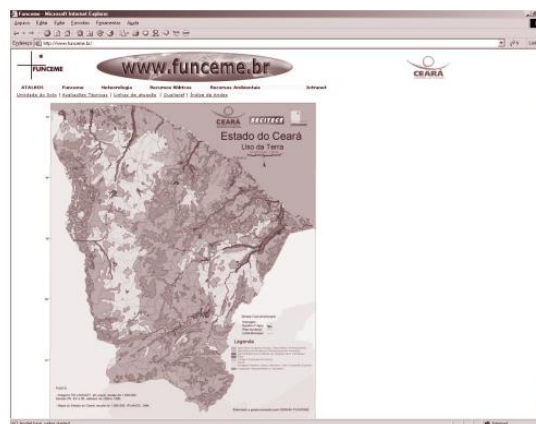


Figura 1. Exemplo da categoria FM: mapa de uso da terra do Estado do Ceará na aplicação SIG Web do FUNCEME [13]

Bibliotecas e Catálogos de Bancos de Dados Espaciais (BC)

Segundo critérios de implementação, as aplicações desta categoria possuem quatro componentes: (i) metadados (informações sobre o dado espacial), (ii) índice de mapas (mostrando localizações de outros mapas – através de hipertextos), (iii) *preview* gráfico (fotos estáticas) e (iv) dados espaciais em formato digital. Os usuários da aplicação podem fazer o *download* desses arquivos de dados espaciais, mas geralmente não podem fazer a sua

visualização no *browser* Web, porque esses arquivos possuem formatos usados em software específicos. Dessa forma, os usuários precisam ter instalado em suas máquinas o SIG correspondente ao formato selecionado, ou possuir algum tipo de *plug-in* para esse formato no *browser* Web. A Figura 2 ilustra uma aplicação desta categoria.

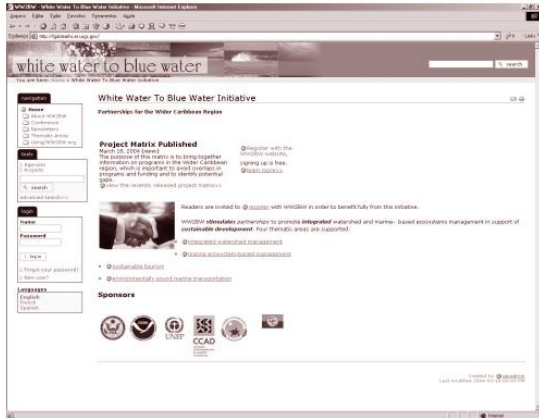


Figura 2. Exemplo de BC: página principal da aplicação SIG Web do Federal Geographic Data Committee [12]

Gerador de Mapas (GM)

Do ponto de vista de implementação da aplicação, é necessário fornecer aos usuários um formulário no *browser* Web para a entrada de especificações tais como: localização, símbolos, níveis temáticos, etc. O formulário é enviado para um servidor SIG; depois disso, a imagem resultado da solicitação realizada via formulário é gerada e, então, enviada de volta para o cliente e visualizada utilizando-se o *browser* Web. Do ponto de vista de uso, a vantagem dessa categoria é a possibilidade de se criar mapas “customizados”, de acordo com as necessidades dos usuários. Desvantagens incluem: velocidade tipicamente baixa e opções limitadas oferecidas ao usuário. Esta categoria possibilita interações como: *zoom in*, *zoom out*, *pan*, consulta de informações associadas a regiões do mapa. Na Figura 3, pode ser observado um exemplo de aplicação GM.

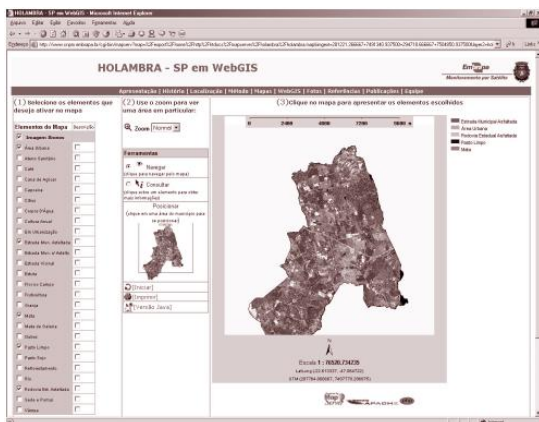


Figura 3. Exemplo de GM: aplicação SIG Web do Embrapa-Holambra, SP [16]

Browser de Mapas em Tempo Real (BM)

Do ponto de vista de implementação, provê acesso para bancos de dados espaciais e permite a usuários visualizar, consultar, recuperar e modificar mapas *on-line*. Este esquema requer servidores de dados espaciais poderosos. O servidor provê *scripts* para receber pedidos dos clientes, incluindo operações de análise espacial. Depois de processar os dados, o servidor envia a imagem e os dados para o cliente em documentos para serem visualizados no *browser* Web. Do ponto de vista do usuário, esta categoria permite modificar mapas armazenados no banco de dados no servidor SIG, ao contrário da categoria de “Gerador de Mapas”. Exemplo desta categoria está ilustrado na Figura 4, para uma aplicação SIG Web proprietária, do CPqD [29].

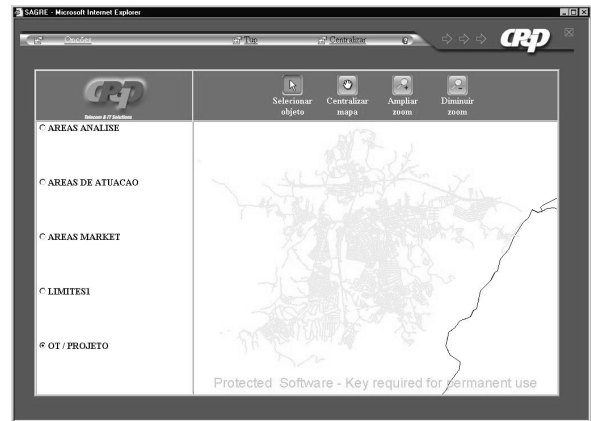


Figura 4. Exemplo de BM: aplicação SIG Web do CPqD (SagreWeb) com mapa de Atibaia, SP [29]

Imagens e Mapas em Tempo Real (IM)

Do ponto de vista de implementação, estas aplicações têm por característica usarem sensores para fornecer informações para o mapa. Ao invés de prover a visualização de um mapa estático, um mapa em tempo real é gerado a partir de um conjunto de sensores que atualizam o mapa continuamente. As imagens são capturadas de vídeo-câmeras, que comunicam-se com um servidor SIG. Isso inclui câmeras terrestres, bem como imagens geradas a partir de satélite. Do ponto de vista de uso, possibilita as interações pertinentes ao software que reproduz as imagens em tempo real. Esta categoria de aplicação foi exemplificada na Figura 5.

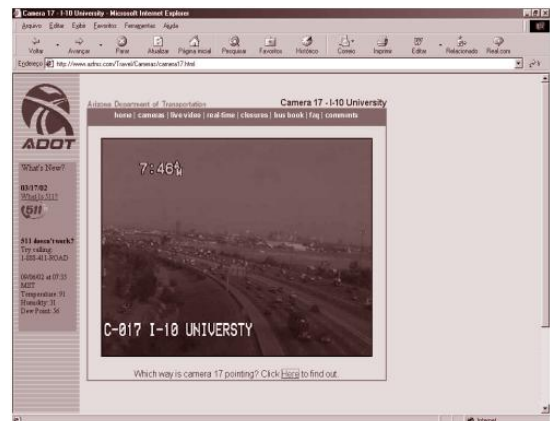


Figura 5. Exemplo de IM: vídeo ilustrado na aplicação SIG Web do Arizona Department of Transportation [2]

O trabalho de Chang [5] defende a participação da comunidade na utilização de SIG, através da Internet e da WWW. Neste contexto, o autor descreve as categorias de aplicações SIG Web que possibilitam diferentes níveis de acesso à informação geográfica. A classificação de Chang [5] é baseada nas tecnologias que possibilitam diferentes níveis de acesso à informação geográfica. Neste trabalho, utilizamos as categorias de Chang para analisar aspectos de interface e interação com elementos do domínio possibilitadas por esses sistemas.

APLICAÇÕES SIG WEB NO DOMÍNIO DE SISTEMAS AGRÍCOLAS

Entre as principais aplicações de SIG em Agricultura, podemos destacar sistemas voltados a reflorestamento, divisas de cadastro, exploração de mineração, gerenciamento de reservas ambientais, análises e prevenções climáticas, sistemas de drenagem e irrigação, análise e monitoramento ambiental, gerenciamento de recursos naturais, produtividade e qualidade, entre outros [35]. Entre algumas das aplicações SIG Web existentes no contexto agrícola, com acesso livre e gratuito, pode-se destacar os sistemas da Embrapa [9], CPTEC [7], SIMEPAR [34], Epagri [10], entre outros. A seção a seguir destaca a metodologia utilizada para o levantamento das aplicações SIG Web e sintetiza em uma tabela algumas dessas aplicações; depois disso, na seção subsequente, discutimos resultados de análise das aplicações consideradas.

Um Levantamento de Aplicações SIG Web no Domínio Agrícola

O conjunto de aplicações SIG Web analisado neste trabalho foi levantado com base nos seguintes critérios:

- (i) Em nível nacional, foi realizada uma busca tomando-se como ponto inicial aplicações SIG Web que são consideradas mais representativas do domínio; a partir destas, identificamos outras aplicações SIG Web no Brasil. Um exemplo de ponto inicial representado é a aplicação SIG Web da Embrapa – referência nacional;
- (ii) Em nível internacional, realizamos a busca por palavras chave (*agriculture application, agriculture GIS, agriculture planning*) e identificamos aplicações SIG Web que consideramos interessantes por serem consideradas de grande abrangência – continental e/ou mundial, como o *Global Hydrology and Climate Center (NASA)*.

A partir da pesquisa inicial, foram documentadas vinte e cinco aplicações SIG Web, apresentadas na Tabela 1, a

seguir. Para cada aplicação, foram destacados dois grupos de informação principais: aspectos relativos à interface de usuário e ao conteúdo tratado no domínio de sistemas agrícolas. A Tabela 1 destaca os seguintes campos independentes:

- (i) Tipo da aplicação SIG Web, de acordo com as cinco categorias de aplicações destacadas por Chang [5] – FM para “Fotos de Mapas”, BC para “Bibliotecas e Catálogos de Bancos de Dados Espaciais”, GM para “Gerador de Mapas”, BM para “*Browser* de Mapas em Tempo Real” e IM para “Imagens e Mapas em Tempo Real”. Algumas aplicações possuem características que possibilitam seu enquadramento em mais de uma categoria;
- (ii) Origem do SIG Web, com destaque para o órgão ou instituição de origem da aplicação SIG Web, bem como sua região geográfica de abrangência.

Com relação à interface de usuário, destacamos as possibilidades de interação com elementos do domínio nos seguintes campos:

- (i) *Zoom*: possibilidade de *zoom* no mapa ou em regiões do mapa;
- (ii) *Pan*: possibilidade de mover o mapa;
- (iii) *Cons*: possibilidade de consulta a informações descritivas associadas aos mapas;
- (iv) *Sel*: possibilidade de selecionar informações tabulares ou mapas através de links nas páginas Web e/ou fornecer especificações em formulários;
- (v) *Anim*: a existência ou não de possibilidade de animação em mapas;
- (vi) *Imag Satél*: existência ou não de visualização de imagens de satélite.

Com relação ao conteúdo apresentado, destacamos a presença ou não de mapas e informações nos seguintes contextos:

- (i) *Zon Agríc e Eco*: zoneamento agrícola e ecológico;
- (ii) *Mapa Veget*: vegetação;
- (iii) *Map Temp*: temperatura da região;
- (iv) *Mapa Clima*: clima da região;
- (v) *Prev Temp*: previsão de tempo;
- (vi) *Mapa Queim*: regiões de queimadas.

Tabela 1. Aplicações SIG Web levantadas no domínio de Sistemas Agrícolas

Aplicação SIG Web		Interação					Conteúdo						
Tipo SIG Web	SIG Web, Origem, Abrangência	Z o o m	P a n	C o n s	S e l	A n i m	Imag Satél	Zon Agric e Eco	Mapa Veget	Map Tem p	Mapa Clima	Prev Temp	Mapa Queim
FM	Climerh Centro Integrado de Meteorológico e Recursos Hídricos de Santa Catarina [6]				X	X	X				X	X	X
FM	CPTEC Centro de Previsão de Tempo e Estudos Climáticos do Inpe (Brasil e alguns países da América do Sul: Bolívia, Peru, Paraguai) [7]			X		X	X		X	X	X	X	X
FM	EPAGRI Empresa de Pesquisa Agropecuária de Santa Catarina [10]				X	X	X	X			X	X	X
FM	Estados do Brasil, Embrapa (Estados do Brasil) [11]	X	X		X				X				
FM	FUNCEME Fundação Cearense de Meteorologia e Recursos Hídricos (Ceará) [13]	X		X	X	X	X	X	X			X	
FM	Global Hydrology and Climate Center (NASA) [15]				X	X	X						
FM	INMET Instituto Nacional de Meteorologia (Brasil) [17]			X	X		X			X		X	X
FM	MASTER Meteorologia Aplicada a Sistemas de Tempo Regionais – USP (Brasil) [22]				X	X	X			X		X	X
FM	NEMRH Núcleo Estadual de Meteorologia e Recursos Hídricos do Maranhão (Maranhão) [23]				X		X			X	X	X	
FM	Pará 30graus Núcleo de Hidrometeorológico da Secretaria Executiva de Ciência, Tecnologia e Meio Ambiente do Estado do Pará – SECTAM (Pará) [25]			X	X	X	X				X	X	X
FM	RNA Rede Nacional de Agrometeorologia (Brasil) [28]				X	X		X		X	X		
FM	SIMEGO Sistema de Meteorologia e Recursos Hídricos do Estado de Goiás (Goiás) [33]			X	X		X				X	X	
FM	SIMEPAR Tecnologia e Informações Ambientais do Paraná (Paraná) [34]			X		X	X		X	X	X	X	X
FM	Weather, Administração Atmosférica e Oceânica Nacional dos USA (NOAA) (USA) [37]	X	X	X		X	X			X		X	
BC, FM	DHM Diretoria de Hidrometeorologia de Alagoas [8]				X		X					X	

Aplicação SIG Web		Interação					Conteúdo						
Tipo SIG Web	SIG Web, Origem, Abrangência	Z o o m	P a n	C o n s	S e l	A n i m	Imag Satél	Zon Agríc e Eco	Mapa Veget	Map Tem p	Mapa Clima	Prev Temp	Mapa Queim
BC, FM	UNEP Grid Arendal, Noruégua (várias re-giões do mundo, não possui Brasil) [36]	X	X	X	X						X		
BC, GM	Agritempo Sistema de Monitoramento Agro-meteorológico(Brasil) [3]	X	X	X	X			X		X		X	
BC, GM	PRODES Digital Inpe (Brasil) [26]				X		X	X	X				
GM	CEPAGRI Meteorologia UNICAMP (Brasil) [4]			X	X	X	X	X		X	X		
GM	HOLAMBRA, Embrapa [16]	X	X	X	X			X	X				
GM	KARS Kansas Applied Remote Sensing Program (University of Kansas) [18]		X	X	X	X	X		X				
GM	Região de Queimadas, Embrapa (Brasil) [27]				X								X
GM	Servidor de Mapas (IBGE) (Brasil) [32]	X	X	X	X			X	X		X		
IM, GM	AccuWeather USA (várias cidades do mundo) [1]	X			X	X	X			X		X	

DISCUSSÃO SOBRE OS SISTEMAS ANALISADOS

Nesta seção, discutimos resultados da análise efetuada sobre as aplicações SIG Web levantadas. Respectivamente, realizamos a discussão com relação à categoria da aplicação SIG Web; focamos o conteúdo; discutimos os aspectos de interação e sugerimos uma classificação com base em aspectos de interatividade desses sistemas.

Quanto ao Tipo da Aplicação SIG Web

A Figura 6 apresenta a categorização das aplicações SIG Web levantadas, de acordo com o modelo de Chang [5].

O gráfico ilustrado na Figura 6 mostra que a maior parte das aplicações SIG Web levantadas enquadra-se na categoria de "Fotos de Mapas", com 59% do total. Encontramos casos onde uma aplicação SIG Web poderia se enquadrar em mais de uma categoria, porque o *web site* da aplicação possuía regiões que faziam classificá-la ora em uma, ora em outra categoria. Nas categorias BC/FM, e também em BC/GM, encontramos 8% das aplicações levantadas, para cada uma; enquanto que, em IM/GM, detectamos 4% das aplicações. Para critérios de análise, nesses casos onde uma aplicação poderia se enquadrar em mais de uma categoria, a aplicação foi enquadrada na categoria considerada mais abrangente, do ponto de vista de interação. Neste caso, as aplicações que ficaram categorizadas como BC e em alguma outra categoria, optamos por colocar a aplicação na outra categoria, pois BC somente permite realizar o *download* de dados disponibilizados na aplicação e a sua

subsequente visualização através de software específico instalado na máquina do usuário. No caso da aplicação que se enquadrou em IM/GM, optamos por colocá-la na categoria GM, pois as características pertinentes à essa categoria, prevalecem dentro do *web site* da aplicação.

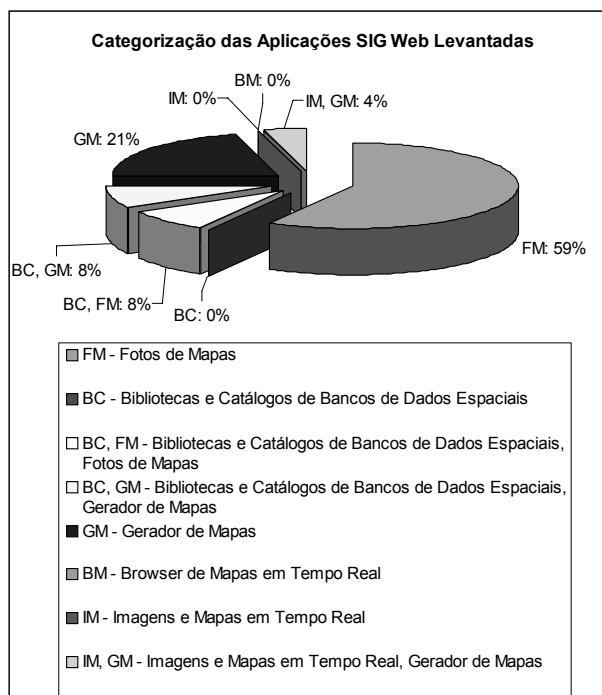


Figura 6. Categorização das aplicações SIG Web levantadas

Na seqüência do levantamento, aparece a categoria "Gerador de Mapas", com 21% das aplicações SIG Web consideradas. Existem mais possibilidades interativas nesta categoria, em relação à FM e BC, sendo possível ao usuário fornecer especificações em um formulário no *browser* web para a requisição de um mapa no servidor SIG.

Não detectamos aplicações puramente nas categorias BC, BM e IM, somente combinadas a outras categorias. As únicas categorias onde encontramos aplicações puramente pertencentes àquelas categorias foram FM e GM.

Cinco das aplicações SIG Web apresentadas são internacionais e dezenove nacionais. As aplicações nacionais apresentaram em média mais recursos de conteúdo por aplicação (2.84), em relação às internacionais (1.2). Com relação a aspectos de interação, as aplicações nacionais apresentaram em média menos recursos de interação por aplicação (3.05), relativamente às internacionais (4.2). A maior parte das aplicações nacionais levantadas aparece na categoria "Fotos de Mapas" (68.4%), o mesmo ocorrendo com as internacionais (60%).

Quanto ao Conteúdo

Algumas das aplicações apresentaram quase a totalidade dos elementos de conteúdo considerados; é o caso do CPTEC e SIMEPAR (com 85.7%) e Epagri (com 71.4%). Todos pertencentes à categoria "Fotos de Mapas". A "previsão de tempo" é a informação de conteúdo mais presente nas aplicações.

O fato de uma aplicação SIG Web ter um conteúdo abrangente não implica que ela terá grandes possibilidades interativas, pois podemos ter nesses casos, simplesmente imagens estáticas – caso das aplicações do CPTEC e do SIMEPAR. O contrário também ocorre: o fato de uma aplicação SIG Web ter grandes possibilidades interativas não implica em conteúdos variados, pois podemos ter aplicações específicas, com os diversos tipos de interação associados a um conteúdo em particular – é o caso do Unep Grid Arendal, HOLAMBRA e Servidor de Mapas (IBGE).

Um recurso interessante para ser disponibilizado em uma aplicação SIG Web é a possibilidade de consulta a informações descritivas relacionadas a localizações do mapa. Ele apareceu em 68.4% das aplicações levantadas, boa parte (61.5%) das quais na categoria "Fotos de Mapas".

Quanto à Interação

Considerando-se os recursos para interação disponibilizados na interface da aplicação, observamos que 54.1% das aplicações pesquisadas possibilitam animação de mapas, ou seja, possibilidade de visualizar mapas em movimento, onde cada *frame* (imagem) da animação representa uma data ou hora específica em que os dados ocorreram. Dessas, grande parte (76.9%) pertence à categoria mais encontrada: "Fotos de Mapas". Esse recurso é importante quando usuários desejam visualizar via mapas as mudanças no decorrer do tempo,

com o intuito, por exemplo, de tomar decisões quanto a colheitas de produtos agrícolas ou definição de datas específicas para plantio. Essa atividade é facilitada pela existência de mapas de previsão de tempo, que aparecem em 58.3% das aplicações. Um recurso não implica na existência do outro; entretanto, essa interseção ocorreu em 37.5% das aplicações, principalmente na classe "Fotos de Mapas" (88.8%).

Com relação aos tipos de interação analisados, observamos que as aplicações Unep Grid Arendal, Agritempo, IBGE Servidor de Mapas e HOLAMBRA apresentaram a maioria dos tipos de interação destacados. A primeira delas pertence à categoria "Fotos de Mapas" e as três seguintes à categoria "Gerador de Mapas". Nessas aplicações, não foram identificados recursos de animação e de visualização de imagens de satélite.

Detectamos que existe um aumento gradativo nas possibilidades de interação à medida que avançamos nas cinco categorias. Tomando-se por índice o número de possibilidades de interação relativo ao total de aplicações por classe, verificamos que as aplicações da classe FM possuem o índice de interatividade de 3.18 e as da classe GM possuem 3.5. Não foram identificadas aplicações unicamente nas classes BC, BM e IM, entretanto, foram identificados BC e IM.

Se considerarmos de forma conjunta os elementos relacionados à interação e ao conteúdo das aplicações SIG Web, observamos que as aplicações do CPTEC, da FUNCEME e do SIMEPAR, tiveram maior quantidade de recursos (66.6% dos recursos de interação e conteúdo avaliados), todas pertencentes à categoria "Fotos de Mapas". Na seqüência, temos as aplicações da Epagri, Pará 30Graus, Weather, Agritempo, do CEPAGRI e o Servidor de Mapas (IBGE), com 58.3% dos recursos. As três primeiras pertencem à categoria "Fotos de Mapas" e as outras à categoria "Gerador de Mapas".

Com relação aos tipos de interação possibilitados, detectamos que os recursos de *pan* e *zoom* em geral aparecem juntos, embora possam existir separadamente. Setenta e cinco por cento de todas as aplicações que possuem o recurso de *zoom* (33.3% das aplicações levantadas) possuem também o recurso de *pan*. Dessas aplicações que possuem tanto o recurso de *zoom* quanto o de *pan*, 50% pertencem à categoria "Fotos de Mapas" e 50% à categoria "Gerador de Mapas". Apenas uma das aplicações – KARS [18] apresentou o recurso de *pan*, mas não o recurso de *zoom*. Com base nesses dados, pode-se verificar que na grande maioria dos casos, o recurso de *pan* aparece associado ao recurso de *zoom*.

A "seleção" foi o recurso encontrado para interação mais freqüente. Dentre as aplicações SIG Web da categoria "Fotos de Mapas" (68% das aplicações levantadas), 31.2% possuem os recursos de seleção de algum tipo de informação e a consulta a dados descritivos associados aos mapas, enquanto que nas aplicações da categoria "Gerador de Mapas" (32% das aplicações levantadas), 62.5% possuem esses dois recursos.

A partir da análise realizada, verifica-se que as aplicações SIG Web mais freqüentes são as pertencentes à categoria

"Fotos de Mapas" e, em segunda instância, pertencentes à categoria "Gerador de Mapas". O fato da categoria "Fotos de Mapas" ter sido freqüente explica-se porque as aplicações desta categoria possuem um grau de sofisticação técnica inferior ao de outras categorias (a categoria com menos sofisticação técnica é a BC, pois ela propõe simplesmente a disponibilização de arquivos para *download*). Provavelmente, as aplicações pertencentes à categoria "Browser de Mapas On-Line" não sejam tão comuns pelo fato de permitirem ao usuário modificar mapas *on-line*. Neste caso, nem todos os usuários estariam habilitados a modificar dados. Essas aplicações são peculiares a sistemas chamados fechados e/ou proprietários, onde é necessário possuir um *login* e senha de acesso.

Classificação Proposta

Considerando-se os resultados, podemos estabelecer uma nova taxonomia gradativa, agora com base em aspectos de diretividade da interação com elementos do domínio em aplicações SIG Web no contexto de agricultura, a saber:

1. Bibliotecas e Catálogos de Dados Espaciais

Aplicações que possibilitam fazer o *download* de arquivos, mas sua visualização e possibilidades interativas são dependentes de algum *plug-in* ou software específico instalado na máquina do usuário.

2. Servidor de Mapas Estático

Aplicações que apresentam imagens capturadas por sensores, vídeo-câmeras, câmeras terrestres, satélites, e disponibilizadas de forma estática. As possibilidades interativas são dependentes do software de visualização. Pode também ter possibilidades interativas, como *zoom*, *pan*, consulta a dados geográficos associados ao mapa.

3. Gerador de Mapas

Aplicações em que os mapas são gerados a partir do fornecimento de especificações pelo usuário em um formulário no web *browser*. É possível realizar *zoom*, *pan*, consulta a dados geográficos associados ao mapa, selecionar camadas de dados para visualização, entre outras possibilidades;

4. Browser de Mapas On-Line

Permite visualizar, consultar, recuperar e modificar mapas *on-line*. É restrito a sistemas proprietários, pertencentes a instituições, órgãos e empresas com acesso restrito aos dados. É importante salientar que as aplicações pertencentes a esta categoria são mais viáveis para Intranets do que para a Internet, pois é muito custoso realizar atualizações em dados através da Internet.

Existem mais opções de interação quando caminhamos gradativamente de 1 a 4. Entretanto, as aplicações das camadas superiores apresentam maior sofisticação em termos técnicos, para a qual usuários casuais podem ter dificuldades. Usuários não sofisticados no uso da tecnologia Web provavelmente terão mais facilidade na

utilização de aplicações SIG Web pertencentes às primeiras camadas da taxonomia definida. A exceção é a primeira categoria: "Bibliotecas e Catálogos de Dados Espaciais", pois seu uso exige a existência de algum software específico instalado na máquina do usuário, que pode demandar conhecimentos específicos. As opções de interação são peculiares ao software instalado na máquina do usuário e não à aplicação SIG Web propriamente dita. Não obstante a isso, o grau de dificuldade de utilização de aplicações SIG Web pertencentes às primeiras categorias da taxonomia definida, poderá ser potencialmente inferior ao das demais.

CONCLUSÕES

O interesse por aplicações SIG na Web vem aumentando bastante nos últimos anos. No contexto de aplicações agrícolas, existe uma importância significativa, pelo fato destas aplicações funcionarem como um ferramental útil para pessoas que estejam envolvidas direta ou indiretamente em planejamento e exploração agrícola. No Brasil, onde o agro-negócio vem crescendo e adquirindo enorme importância no PIB, o estudo destas aplicações adquire, inclusive, relevância econômica. O uso dessas aplicações por agricultores e por profissionais que tomam decisões nessa área, poderá ser uma solução de acesso simples e rápido através da WWW, e de baixo custo, pois não é necessário, por exemplo, que o agricultor procure e consulte profissionais especializados no assunto, bastando usufruir dos recursos disponibilizados pelas aplicações SIG na Web. Entretanto, o acesso às aplicações SIG na Web é dependente dos recursos de interação possibilitados pelas interfaces dessas aplicações. O desafio desses sistemas é grande dada a diversidade de usuários que podem beneficiar-se deles.

As aplicações SIG na Web possuem conteúdo e tipos de interação diversificados. Este artigo teve por objetivo investigar tais aplicações no domínio agrícola sob esses dois aspectos. Com base nas classes de aplicações SIG Web propostas por Chang [5], verificamos que a classe de aplicações "Fotos de Mapas" é a mais comum. Aplicações que possibilitam interação *online* com mapas gerados em tempo real parecem não fazer sentido no domínio agrícola, onde imagens de satélite para determinada região têm a periodicidade de um dia.

A partir da análise realizada, chegamos a uma classificação baseada na diretividade de acesso e interação com elementos do domínio da aplicação SIG Web. Detectamos que existe um aumento nas possibilidades de interação quando se caminha da primeira até a última classe de aplicações SIG Web, na taxonomia proposta neste trabalho. O próximo passo nesta pesquisa envolve a realização de análises mais específicas em determinados sistemas, considerando-se o envolvimento direto de usuários prospectivos e critérios de qualidade em interfaces.

AGRADECIMENTOS

Agradecemos ao projeto SAI - *Advanced Information Systems* (Sistemas de Informação Avançados), PRONEX II/MCT-CNPq (Programa de Apoio à Núcleos de

Excelência) e dos projetos CNPq WEB-MAPS e AgroFlow, e ao Instituto de Computação da Unicamp.

REFERÊNCIAS

1. AccuWeather, AccuWeather USA [on line]. Disponível em: <http://www.accuweather.com/>. Último acesso: 22/03/2004.
2. ADOT, Arizona Department of Transportation [on line]. Disponível em: <http://www.azfms.com/Travel/Cameras/camera17.html>. Último acesso: 22/03/2004.
3. Agritempo, Sistema de Monitoramento Agrometeorológico [on line]. Disponível em: <http://www.agritempo.gov.br/>. Último acesso: 22/03/2004.
4. CEPAGRI, CEPAGRI Meteorologia UNICAMP [on line]. Disponível em: <http://www.cepagri.unicamp.br/portal/index.php>. Último acesso: 22/03/2004.
5. Chang, K., The Design of A Web-based Geographic Information System for Community Participation. Masters Thesis, Buffalo University, J.B. Krygier's research (1997).
6. Climerh, Centro Integrado de Meteorológico e Recursos Hídricos de Santa Catarina [on line]. Disponível em: <http://www.climerh.rct-sc.br/>. Último acesso: 22/03/2004.
7. CPTEC, Centro de Previsão de Tempo e Estudos Climáticos, INPE Instituto Nacional de Pesquisas Espaciais, São José dos Campos, SP [on line]. Disponível em: <http://www.cptec.inpe.br/>. Último acesso: 22/03/2004.
8. DHM, Diretoria de Hidro-meteorologia de Alagoas [on line]. Disponível em: <http://www.tempo.al.gov.br/>. Último acesso: 22/03/2004.
9. Embrapa, Embrapa [on line]. Disponível em: <http://www.cnpm.embrapa.br/>. Último acesso: 22/03/2004.
10. Epagri, Empresa de Pesquisa Agropecuária e Extensão Rural de Santa Catarina [on line]. Disponível em: <http://www.epagri.rct-sc.br/>. Último acesso: 22/03/2004.
11. Estados, Estados do Brasil, Cnpm, Embrapa [on line]. Disponível em: <http://www.cdbrasil.cnpm.embrapa.br/>. Último acesso: 22/03/2004.
12. Federal Geographic, Federal Geographic Data Committee [on line]. Disponível em: <http://fgdclearhs.er.usgs.gov/>. Último acesso: 22/03/2004.
13. FUNCEME, Fundação Cearense de Meteorologia e Chuvas. [on line]. Disponível em: <http://www.funceme.br/>. Último acesso: 22/03/2004.
14. Gisweb, Portal GisWeb: Sistemas de Informação Geográfica na Web [on line]. Disponível em: <http://www.gisweb.com.br/>. Último acesso: 22/03/2004.
15. Global, Global Hydrology and Climate Center (NASA) [on line]. Disponível em: <http://www.ghcc.msfc.nasa.gov/GOES/>. Último acesso: 22/03/2004.
16. HOLAMBRA, HOLAMBRA em Web-GIS, Embrapa [on line]. Disponível em: <http://www.cnpm.embrapa.br/>. Último acesso: 22/03/2004.
17. INMET, Instituto Nacional de Meteorologia [on line]. Disponível em: <http://www.inmet.gov.br/>. Último acesso: 22/03/2004.
18. KARS, Kansas Applied Remote Sensing Program, University of Kansas [on line]. Disponível em: <http://www.kars.ku.edu/>. Último acesso: 08/04/2004.
19. Krygier, J.B., Buffalo's Lower West Side. WWW Project [on line]. Disponível em: <http://www.geog.buffalo.edu/~jkrygier/>. Último acesso: 08/04/2004 (1997).
20. Mapinfo, MapInfo [on line]. Disponível em: <http://www.mapinfo.com/>. Último acesso: 22/03/2004.
21. MapObjects, MapObjects [on line]. Disponível em: <http://www.esri.com/software/mapobjects/>. Último acesso: 22/03/2004.
22. MASTER, Meteorologia Aplicada a Sistemas de Tempo Regionais – USP [on line]. Disponível em: <http://www.master.iag.usp.br/>. Último acesso: 22/03/2004.
23. NEMRH, Núcleo Estadual de Meteorologia e Recursos Hídricos do Maranhão [on line]. Disponível em: <http://www.nemrh.uema.br/>. Último acesso: 22/03/2004.
24. Oliveira, J.L. de, Projeto e Implementação de Interfaces para Sistemas de Aplicações Geográficas, tese de doutorado, IC-UNICAMP (1997).
25. Pará, Pará 30graus - Núcleo de Hidro-meteorológico da Secretaria Executiva de Ciência, Tecnologia e Meio Ambiente do Estado do Pará – SECTAM [on line]. Disponível em: <http://www.para30graus.pa.gov.br/hidrico.htm>. Último acesso: 22/03/2004.
26. PRODES, PRODES Digital Inpe [on line]. Disponível em: <http://www.obt.inpe.br/prodes/>. Último acesso: 22/03/2004.
27. Queimadas, Região de Queimadas, Cnpm, Embrapa [on line]. Disponível em: <http://www.queimadas.cnpm.embrapa.br/>. Último acesso: 22/03/2004.
28. RNA, Rede Nacional de Agrometeorologia [on line]. Disponível em:

- <http://masrv54.agricultura.gov.br/rna/>. Último acesso: 22/03/2004.
29. SagreWeb, Aplicação SIG Web do CPqD – Centro de Pesquisa e Desenvolvimento, Campinas, SP.
 30. Schimiguel, J., Interface 3D de Aplicações SIG como Espaço de Comunicação, tese de mestrado, IC-UNICAMP (2002).
 31. Schimiguel, J., Baranauskas, M.C.C., Medeiros, C.M.B., Modelando a Interface de Aplicações em SIG como Espaço de Comunicação. IV Workshop Brasileiro sobre Fatores Humanos em Sistemas Computacionais. Florianópolis, SC (2001).
 32. Servidor, Servidor de Mapas – IBGE [on line]. Disponível em: <http://www1.ibge.gov.br/mapserver/>. Último acesso: 22/03/2004.
 33. SIMEGO, Sistema de Meteorologia e Recursos Hídricos do Estado de Goiás [on line]. Disponível em: <http://www.simego.sectec.go.gov.br/>. Último acesso: 22/03/2004.
 34. SIMEPAR, Tecnologia e Informações Ambientais [on line]. Disponível em: <http://www.simepar.br>. Último acesso: 22/03/2004.
 35. Tosi, F. de A., GPS na Agricultura [on line]. Disponível em: <http://www.gpsglobal.com.br/Artigos/Agricola/SIG.html>. Último acesso: 08/03/2004.
 36. UNEP, UNEP Grid Arendal, Noruega [on line]. Disponível em: <http://www.grida.no/>. Último acesso: 08/03/2004.
 37. Weather, Administração Atmosférica e Oceânica Nacional dos USA (NOAA) [on line]. Disponível em: <http://weather.noaa.gov/>. Último acesso: 08/03/2004.
 38. W3C, W3C World Wide Web Consortium [on line]. Disponível em: <http://www.w3.org/>. Último acesso: 08/03/2004.

O Uso de Conhecimento Prévio do Domínio em Programação por Demonstração

Raquelina Ritter de Moura Penteado

Sérgio Roberto Pereira da Silva

Departamento de Informática — Universidade Estadual de Maringá
Av. Colombo, 5790, zona 07, 87020-900, Maringá-PR, Brasil
{raque,srsilva}@din.uem.br

RESUMO

Neste artigo propomos que aplicações extensíveis que empregam o mecanismo de programação por demonstração implícita podem ter seu processo de criação de extensões melhorado quando o conhecimento prévio do domínio da aplicação é considerado. O uso deste conhecimento permite que as repetições executadas pelos usuários na interface da aplicação possam ser detectadas mesmo quando suas ações não forem consecutivas ampliando a usabilidade das aplicações extensíveis. Ele também permite a exibição de generalizações dos tipos: constante, variável e *loops* simples, aninhados, condicionais e dependentes aos usuários. Para o processo de detecção de repetições de ações utilizamos o conceito de máquinas de estado e, para o processo de inferência de generalizações empregamos o algoritmo de aprendizado *Find-S* estendido para o uso de conhecimento prévio do domínio da aplicação. Disponibilizamos este conhecimento na aplicação empregando a arquitetura do Modelo Semiótico para aplicações extensíveis.

ABSTRACT

In this paper, we propose that extensible applications that make use of the implicit programming by demonstration approach could have its extension creation process improved when application domain background knowledge is considered. The use of such knowledge allows for the detection of end-users actions executed in the application interface even when their actions are not consecutive raising the usability of the application. It also allows the exhibition of generalizations of type: constant, variable, and simple, nested, conditional and dependent loops for the users. For the detection of action repetition process, we use the state machine concept, and for the creation of generalizations process, we make use of the Find-S learning algorithm extended for the use of application domain background knowledge. We make this kind of knowledge available in the application by the use of the Semiotic Model for extensible applications architecture.

Palavras chaves

Programação por Demonstração, Aplicações Extensíveis, Uso de Conhecimento Prévio do Domínio.

1. INTRODUÇÃO

Na interação cotidiana dos usuários com as aplicações interativas, em geral, são executadas ações repetitivas. Isto nos indica que os usuários poderiam trabalhar de uma forma mais produtiva se estas ações fossem automatizadas, isto é, os usuários poderiam encontrar uma maior usabilidade nas aplicações interativas.

Entendemos por usabilidade um conceito que se refere à qualidade da interação de sistemas com os usuários e que depende de vários fatores, tais como [4]:

- A facilidade do aprendizado do sistema: tempo e esforço exigido para que o usuário atinja um determinado nível de desempenho.
- A facilidade de uso do sistema: avalia o esforço físico e cognitivo do usuário durante o processo de interação com o sistema.
- A satisfação do usuário: avalia se o usuário gosta e sente prazer em trabalhar com o sistema.
- A flexibilidade: avalia a possibilidade de o usuário acrescentar e modificar as funções e o ambiente iniciais do sistema.
- A produtividade: avalia se o sistema permite ao usuário ser mais produtivo do que seria se não o utilizasse.

Para automatizar as ações repetitivas, as aplicações interativas comerciais têm disponibilizado mecanismos para a criação de extensão às suas funcionalidades, sendo os mais comuns: o gravador de macros e a linguagem de *script*. Com o gravador de macros a aplicação grava uma seqüência fixa de ações, executadas na linguagem de interface da aplicação pelo usuário, que pode ser replicada posteriormente. Este mecanismo não exige conhecimento de técnicas de programação por parte do usuário. Com a linguagem de *script* os usuários descrevem suas intenções por meio de uma linguagem de programação e, portanto, eles terão um maior poder de expressão, podendo criar extensões mais sofisticadas. Entretanto, este mecanismo requer conhecimentos de técnicas de programação que muitos usuários geralmente não possuem. Além do mais, estes dois mecanismos apresentam uma mesma limitação, ambos interrompem o fluxo de trabalho dos usuários.

O mecanismo de programação por demonstração (*PbyD*) permite a combinação da interface simples do gravador de macros com a expressividade de uma linguagem de *script* [1]. Este mecanismo, como o gravador de macros, também possibilita que os usuários leigos demonstrem as tarefas que desejam automatizar por meio da linguagem de interface, porém, os programas aprendidos podem

conter estruturas de controle, tais como: iteração, condicional e abstração [7]. O mecanismo de *PbyD* pode ser classificado como explícito ou implícito [17]. No primeiro, os usuários devem demonstrar explicitamente o início e o término da seqüência de ações que desejam automatizar; e no segundo, o mecanismo captura as intenções dos usuários de uma forma autônoma detectando repetições nas ações executadas pelos usuários. O mecanismo implícito apresenta vantagem sobre o explícito, pois este último também interrompe o fluxo de trabalho dos usuários.

As implementações atuais que utilizam o mecanismo de *PbyD* implícito detectam tanto seqüências de ações repetitivas consecutivas quanto não-consecutivas [16], porém estas seqüências devem ser executadas de uma forma idêntica, ou seja, devem possuir as mesmas ações e estas ações devem ser executadas da mesma maneira.

Por meio da detecção de repetições, as aplicações *PbyD* implícitas procuram inferir as generalizações para as tarefas dos usuários na tentativa de automatizar suas atividades. As aplicações *PbyD* implícitas atuais, até onde vai nosso conhecimento, apresentam somente generalizações do tipo constante, *loop* simples e *loops* aninhados. Não sendo, portanto, capazes de apresentar generalizações que atendam a todas as intenções reais dos usuários.

Na tentativa de minimizar as limitações encontradas no mecanismo de *PbyD* implícito, propomos o uso de conhecimento prévio do domínio da aplicação no processo de criação de extensões às aplicações que utilizam este mecanismo. Mais especificamente propomos empregar este conhecimento no processo de detecção de repetições nas ações e no processo de inferência de generalizações. Deste modo, será possível fazer que o processo de detecção de repetições identifique seqüências de ações repetitivas mesmo quando alguma seqüência for interrompida por algum ruído e, também, que o processo de inferência de generalizações considere além das estruturas do tipo *loop*, desvios condicionais e variáveis em suas generalizações. Logo, usuários leigos poderão encontrar uma maior facilidade para criar extensões em suas aplicações extensíveis, podendo automatizar grande parte de suas tarefas repetitivas cotidianas.

Assim, na seção 2 deste artigo, apresentamos uma visão geral de nossa proposta. Na seção 3, descrevemos o processo de detecção de repetições e, na seção 4, delineamos o processo de criação de generalizações, ambos desenvolvidos com a intenção de facilitar o processo de criação de extensões por parte dos usuários. Para concluir, na seção 5, comparamos nosso trabalho com algumas aplicações existentes, discutindo os resultados alcançados e apresentamos alguns trabalhos futuros.

2. O USO DE CONHECIMENTO PRÉVIO DO DOMÍNIO DA APLICAÇÃO EM PROGRAMAÇÃO POR DEMONSTRAÇÃO

Nesta seção vamos descrever, de forma geral, como as aplicações que empregam o mecanismo de *PbyD* implícita podem fazer uso de conhecimento prévio do

domínio da aplicação. Para isto é necessário entender o processo de criação de uma extensão nas aplicações *PbyD* atuais.

Na realização de uma extensão uma aplicação *PbyD* deve cumprir as seguintes etapas: 1) capturar automaticamente as ações realizadas pelos usuários na sua interface, 2) detectar as tarefas dos usuários presentes nesta seqüência de ações, 3) detectar as repetições neste conjunto de tarefas, 4) inferir as generalizações possíveis sobre o conjunto de repetições detectado e 5) apresentá-las aos usuários. Por **tarefa** entenda-se um conjunto de ações que operam sobre o mesmo objeto do domínio de uma aplicação, e por **tarefas repetitivas** as tarefas que possuem as mesmas ações, realizadas consecutivamente ou não, mesmo que estas sejam executadas sobre objetos diferentes. Assim, as tarefas “selecionar” e “alterar o tamanho da fonte” de uma ‘tabela’ e de um ‘parágrafo’ em um editor de texto formam uma tarefa repetitiva, pois possuem as mesmas ações (“selecionar” e “alterar a fonte”).

A introdução do uso de conhecimento prévio do domínio da aplicação no processo de criação de extensões nas aplicações *PbyD* requer o emprego de quatro agentes inteligentes de software para realizar as etapas acima citadas. Para disponibilizar o acesso ao conhecimento prévio do domínio da aplicação a estes agentes consideramos uma adaptação da arquitetura de software proposta no Modelo Semiótico para aplicações extensíveis [3]. Este modelo propõe o emprego de uma base de conhecimento do design da aplicação (BCDA) que contém conhecimento do domínio e do design da aplicação. Esta base, em geral, é modelada pelo designer de software durante a construção de uma aplicação. Nela deverão estar representados os objetos do domínio, que são visíveis pelo usuário, e as ações que poderão ser executadas sobre eles, com os seus respectivos parâmetros. Além disso, deverão ser modeladas as relações estáticas existentes entre estes objetos (particularmente as relações de herança e partonomia). Para ser empregada em aplicações *PbyD* será necessário que o designer de software defina também uma máquina de estado para cada objeto do domínio, a qual será utilizada no processo de detecção de repetições. A arquitetura modificada pode ser visualizada na Figura 1.

O processo de criação de extensões para aplicações *PbyD*, proposto neste trabalho, pode ser visualizado na Figura 2. Nele podemos ver os quatro agentes de software propostos e a base de conhecimento e os históricos empregados no processo. Os agentes de software propostos devem trabalhar simultaneamente em *background* para a consecução do processo de criação de extensões. A tarefa de cada agente está assim descrita:

- **Agente observador** : captura as ações executadas na interface pelos usuários, as formata e as armazena em um histórico de ações. Para esta formatação o agente observador deve acessar o conhecimento prévio do domínio da aplicação, armazenado na BCDA, para consultar quais parâmetros considerar na ação. Este conhecimento é importante devido ao fato de que a mesma ação pode ser executada em objetos diferentes, logo parâmetros diferenciados podem ser

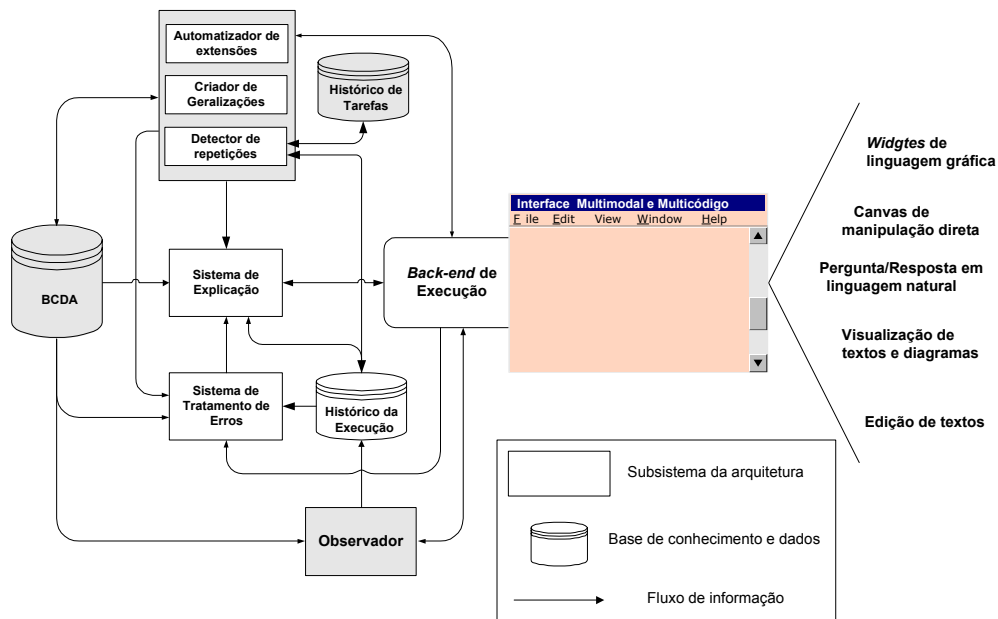


Figura 1 – Arquitetura do Modelo Semiótico para Aplicações Extensíveis [3] modificada com a introdução dos agentes proposto neste trabalho.

exigidos. Por exemplo, o formato padrão (conjunto de parâmetros relevantes) da ação “selecionar” para o objeto ‘parágrafo’ poderá conter os parâmetros “tipo” e “tamanho da fonte” do ‘parágrafo’. Já para o objeto ‘tabela’ o formato padrão poderá ser diferente contendo parâmetros como o “número de linhas e colunas” e o “formato da tabela”.

- **Agente detector de repetições:** conforme o agente observador atualiza o histórico de ações, o agente

detector é ativado para verificar se alguma tarefa foi finalizada. Para isto, ele consulta as máquinas de estados existentes na BCDA. Com o término de uma tarefa, o agente detector atualiza o seu histórico de tarefas e procura por tarefas repetitivas. Se houver repetições, generalizações poderão ser criadas pelo agente criador de generalizações.

- **Agente criador de generalizações:** com repetições detectadas, este agente deve inferir generalizações

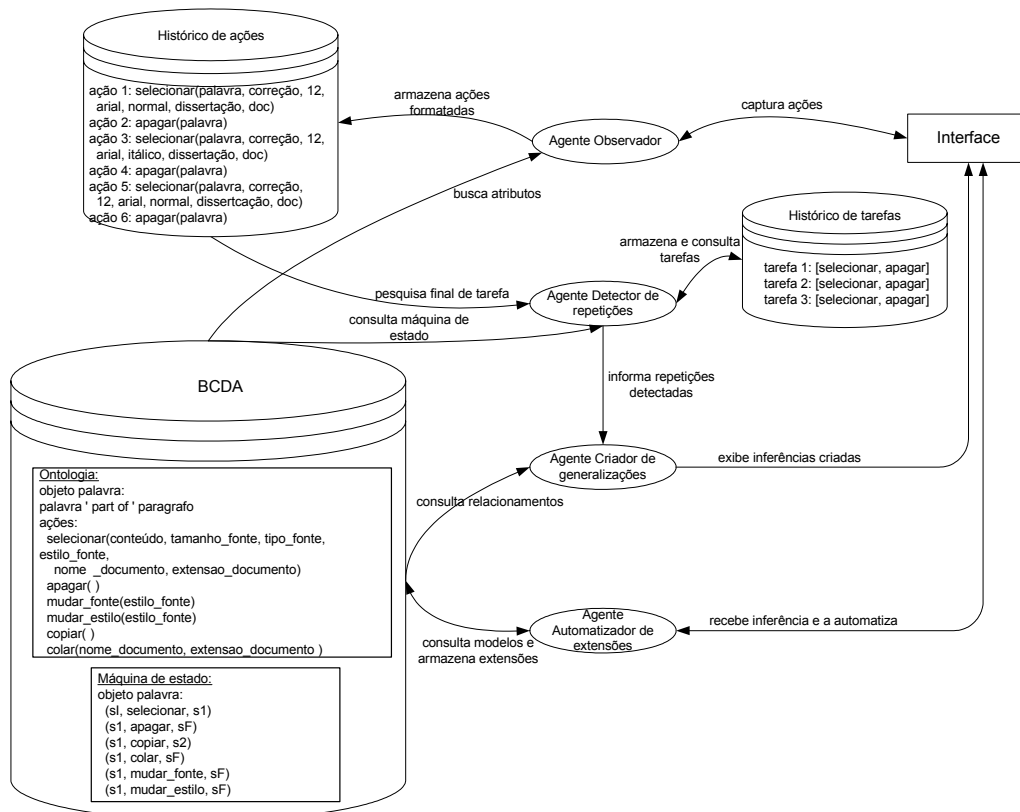


Figura 2 – Processo de criação de extensões em uma aplicação *PbyD* implícita.

que serão exibidas aos usuários na tentativa de automatizar suas tarefas. As generalizações devem ser exibidas aos usuários para que haja um equilíbrio entre o poder dos usuários e da aplicação. Este cuidado é importante, pois, de acordo com [13], uma aplicação *PhyD* implícita não deverá automatizar uma generalização sem a autorização explícita do usuário, visto que nenhum usuário gosta de perceber que tarefas estão sendo realizadas sem seu conhecimento e aprovação.

- **Agente automatizador de extensões:** após os usuários escolherem uma generalização exibida, este agente deve automatizá-la. Neste processo o agente cria o código de extensão automaticamente e realiza um teste semântico da extensão com os usuários, antes de executá-la. Até o momento da sua execução uma extensão é considerada volátil, pois os usuários podem utilizá-la e descartá-la posteriormente. Caso os usuários tenham interesse em utilizá-la posteriormente, a extensão poderá se tornada permanente. Para isto, o agente deve criar os elementos de interação necessários para a ativação da extensão na interface e para adquirir os dados necessários a sua execução e a inseri-la na BCDA, tornando-a uma nova funcionalidade da aplicação.

Para apresentarmos um exemplo de como o processo de criação de extensões trabalha vamos considerar a seguinte situação: em um editor de texto, um usuário seleciona e apaga consecutivamente três vezes a palavra “correção”. Conforme o processo descrito na Figura 2, o agente observador captura cada ação que o usuário realiza na interface da aplicação, a formata e a inclui no histórico de ações. Ao ser executada uma nova ação, o agente detector verifica se uma tarefa é finalizada para inseri-la no histórico de tarefas. Neste exemplo, ao término da segunda, da quarta e da sexta ação, por meio de consultas à máquina de estado do objeto ‘palavra’ na BCDA, o agente detecta que tarefas foram finalizadas. Ao final da segunda tarefa, ou seja, da quarta ação, o agente detector verifica que tarefas repetitivas estão sendo executadas, pois há tarefas no histórico de tarefas que possuem as mesmas ações (tarefa 1 e 2). Com a terceira tarefa finalizada, o agente detector verifica que três tarefas repetitivas consecutivas foram realizadas e o agente criador de generalizações é acionado. Com as tarefas consideradas repetitivas, as tarefas 1, 2 e 3, o agente criador de generalizações irá inferir três generalizações (apresentadas na Figura 3). Sendo assim, o usuário poderá selecionar alguma delas para que o agente automatizador de extensões automatize o trabalho do usuário. A primeira generalização permite a exclusão de uma palavra “correção” a cada ativação. Já as outras apresentam uma maior cobertura por considerar *loops* (*For each...*).

Nas duas próximas seções detalhamos o processo de detecção de repetições e o de criação de generalizações. A descrição completa do processo de criação de generalizações para aplicações *PhyD* implícitas pode ser encontrado em [14].

```

Generalizações:
Generalização número 1:
selecionar(_Palavra,correção,tamanho_fonte =
  12, _tipo_fonte =
  arial,_Estilo_fonte,nome_documento =
  dissertação, extensão_documento = doc)
apagar(_Palavra)
Generalização número 2:
For each _Palavra of the _Parágrafo do
selecionar(_Palavra,correção,tamanho_fonte =
  12, _tipo_fonte =
  arial,_Estilo_fonte,nome_documento =
  dissertação, extensão_documento = doc)
If (_Estilo_fonte = normal) OR
  (_Estilo_fonte = itálico) then
  apagar(_Palavra)
Generalização número 3:
For each _Palavra of the _Parágrafo do
selecionar(_Palavra,correção,tamanho_fonte =
  12, _tipo_fonte=arial,
  _Estilo_fonte,nome_documento =
  dissertação,extensão_documento = doc)
apagar(_Palavra)

Informe o número da generalização que você
deseja automatizar: _

```

Figura 3 – Exemplos de generalizações criadas pelo agente criador de generalizações.

3. O PROCESSO DE DETECÇÃO DE REPETIÇÕES

Para que repetições possam ser detectadas, o agente detector deverá identificar o término de uma tarefa e, em seguida, procurar por tarefas repetitivas à tarefa finalizada no histórico de tarefas. Para detectar o término de uma tarefa, o agente detector deverá consultar na BCDA a máquina de estado do objeto que estiver sendo considerado na tarefa e verificar se o nó final desta máquina será alcançado com as ações pertencentes à seqüência em questão (somente assim uma tarefa será considerada finalizada). Esta máquina de estado é particular de cada tipo de objeto e modela os estados pelos quais um objeto pode passar durante a execução de uma tarefa. Uma máquina de estado deverá possuir um nó inicial (sI), um nó final (sF), um conjunto de arestas que indicam as ações que podem ser executadas sobre o objeto e um conjunto de nós intermediários (s1, s2, ..., sN) que indicam os estados que a tarefa pode assumir. Uma tarefa é definida por meio das interdependências entre estas ações. Na Figura 4 temos uma parte da máquina de estado do objeto ‘palavra’. Note que uma tarefa poderá ser tanto um conjunto de ações interdependente (“selecionar”, “copiar”, “colar”) quanto uma única ação (“apagar”).

Na modelagem das máquinas de estado dos objetos o designer deverá considerar em cada máquina todas as

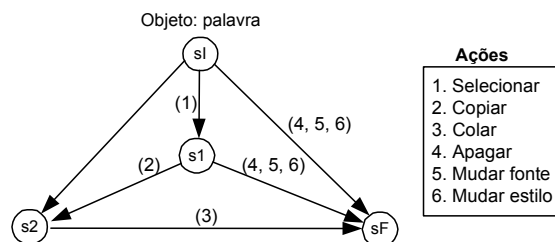


Figura 4 - Máquina de estado do objeto palavra de um editor de texto

ações que poderão ser executadas sobre um determinado objeto, e todas as seqüências de ações válidas. No entanto, não deverá inserir *loops* na máquina, pois o agente detector os considerará automaticamente ao identificar tarefas repetitivas na interface de uma aplicação. Recomenda-se que o designer considere como ações interdependentes somente aquelas em que uma ação dependa incondicionalmente da anterior. Por exemplo, para uma tabela ser “colada” ela deverá necessariamente ser “copiada” ou “recortada” antes.

No caso da existência de duas tarefas serem consideradas em uma única tarefa, como a ação “apagar” na seqüência (“selecionar”, “copiar”, “colar”, “apagar”), o processo de detecção poderá ficar comprometido, pois o agente detector considerará como tarefas similares somente àquelas que possuem toda a seqüência de ações idêntica. Deste modo, se após esta seqüência um usuário “selecionar”, “copiar” e “colar” uma ‘tabela’ não ocorrerá à detecção de uma tarefa repetitiva, pois esta tarefa não possuirá as mesmas ações que a anterior. Isto poderá tornar o processo de detecção muito esporádico e, conseqüentemente, fraco, pois não ocorreriam as devidas automatizações no trabalho do usuário.

Considerando a máquina de estado apresentada na Figura 4, após um usuário “selecionar” uma ‘palavra’ ocorrerá a introdução da ação “selecionar” no histórico de ações e, o agente detector verificará na máquina de estado do objeto ‘palavra’ se o nó final será alcançado. Neste caso, o nó alcançado será um nó intermediário (s1) e o agente não executará o processo de detecção de tarefas repetitivas, pois a tarefa não será finalizada. Porém, se a ‘palavra’ for “apagada” após ser “selecionada”, o processo de detecção será ativado, pois com esta seqüência de ações o nó final (sF) da máquina de estado será atingido, finalizando a tarefa.

É importante ressaltar que durante o processo de verificação do término de uma tarefa, o agente também deverá consultar na BCDA se o objeto em questão herdar características de outro objeto. Se isto ocorrer, a máquina de estado do objeto pai também deverá ser considerada no processo, pois só assim todas as tarefas pertencentes àquele objeto serão consideradas. Por exemplo, se uma tarefa estiver sendo executada sobre o objeto ‘quadrado’ no domínio de um editor gráfico, e este objeto herdar características de outro objeto, por exemplo, do objeto ‘figura’, o agente detector deverá procurar detectar o término de uma tarefa tanto na máquina de estado do objeto ‘quadrado’ quanto na máquina de estado do objeto ‘figura’, pois o objeto ‘quadrado’ herda características do objeto ‘figura’.

Toda vez que uma nova ação for inserida no histórico de ações, além do agente detector verificar se uma tarefa será finalizada com esta nova ação, ele deverá atribuir um identificador único à ação e armazenar seus parâmetros em uma lista de parâmetros para que eles possam ser considerados no processo de criação de generalizações. Toda nova ação inserida no histórico de ações poderá:

1) **Ser interdependente** da última ação da tarefa corrente, resultando em duas possíveis situações:

a) A tarefa corrente ser finalizada. Neste caso, a ação será inserida na lista que armazena as ações da tarefa corrente, a tarefa corrente será concluída e incluída no histórico de tarefas da aplicação, e o agente detector procurará por tarefas repetitivas.

b) A tarefa corrente não ser finalizada. Neste caso, a ação será simplesmente inserida na lista de ações da tarefa corrente, para que a tarefa possa ser finalizada posteriormente.

2) **Não ser interdependente** da última ação da tarefa corrente, caso em que a ação executada representará um **ruído** para a tarefa corrente, resultando em três situações possíveis:

a) Existe uma tarefa pendente e a ação é interdependente desta tarefa. Neste caso, a nova ação será adicionada na lista de ações da tarefa pendente, podendo ou não finalizar esta tarefa. Se a tarefa pendente for finalizada ela deverá ser inserida no histórico de tarefas e o agente detector procurará por tarefas repetitivas. Porém, para que a ação seja considerada na tarefa pendente, além de ser interdependente da última ação da tarefa pendente, os objetos considerados tanto na ação quanto na tarefa pendente deverão ser os mesmos. Esta restrição existe para garantir que a ação seja uma continuação da tarefa pendente.

b) Existe uma tarefa pendente e a ação não é interdependente desta tarefa. Neste caso, não ocorrerá nenhum processamento, pois o agente considerará somente uma tarefa pendente por vez. Seria possível considerar mais de uma tarefa pendente ao mesmo tempo, no entanto não consideramos esta situação neste artigo.

c) Não existe tarefa pendente. Neste caso, uma nova tarefa pendente será criada. Para isto, a lista de ações da tarefa pendente será igualada a lista de ações atual da tarefa corrente e a lista de ações da tarefa corrente conterá somente a nova ação, isto é, o **ruído**. A nova tarefa pendente poderá ser finalizada posteriormente com um novo ruído em outra tarefa.

Quando uma tarefa for finalizada o agente detector iniciará a identificação de repetições, na qual poderá detectar tarefas repetitivas ou seqüências de tarefas repetitivas. Em ambos os casos, as tarefas ou seqüências poderão ser consecutivas ou não-consecutivas. Na tarefa de identificação este agente usará duas listas de tarefa, a lista seqüência-tarefas será usada para armazenar as tarefas finalizadas recentemente que possuem tarefas repetitivas no histórico de tarefas, e a lista tarefas-repetitivas será usada para armazenar as tarefas que já estavam no histórico de tarefas e foram consideradas repetitivas. Deste modo, os diferentes tipos de repetições serão detectadas da seguinte forma:

1) Tarefas repetitivas consecutivas:

As repetições deste tipo poderão ser identificadas após a realização de, no mínimo três tarefas. Deste modo, considerando o contexto em que as tarefas T_1 , T_2 e T_3 forem realizadas consecutivamente, sendo as tarefas T_1 e T_2 similares. Ao término da tarefa T_2 a lista tarefas-

repetitivas será igual a T_1 e a lista seqüência-tarefas será igual a T_2 . Assim, poderá ocorrer de duas situações:

- a) A terceira tarefa for similar às anteriores. Caso em que o agente detector poderá concluir que três tarefas repetitivas estão sendo realizadas consecutivamente, sendo elas T_1 , T_2 e T_3 .
- b) A terceira não for similar às anteriores. Caso em que o agente detector poderá concluir que duas tarefas repetitivas foram executadas consecutivamente, sendo elas T_1 e T_2 .

2) Tarefas repetitivas não-consecutivas:

As repetições deste tipo poderão ser identificadas quando duas tarefas repetitivas forem executadas não-sequencialmente e uma seqüência de tarefas repetitivas não for formada. Por exemplo, considerando o contexto em que as tarefas T_1 , T_2 , T_3 e T_4 forem executadas consecutivamente, sendo as tarefas T_1 e T_3 similares. Ao término da tarefa T_3 a lista tarefas-repetitivas será igual a T_1 e a lista seqüência-tarefas igual a T_3 . Neste caso, se ao término da tarefa T_4 o agente não detectar nenhuma tarefa similar a ela, ou detectar tarefa(s) similar(es) à ela que não seja a tarefa T_2 , o agente poderá concluir que duas tarefas repetitivas foram executadas não-consecutivamente, sendo elas T_1 e T_3 .

3) Seqüência de tarefas repetitivas consecutivas:

As repetições deste tipo poderão ser identificadas após a realização de, no mínimo, três seqüências de tarefas. Deste modo, considerando o contexto em que as tarefas T_1 , T_2 , T_3 , T_4 e T_5 forem realizadas consecutivamente, sendo a tarefa T_1 similar a T_3 e a tarefa T_2 similar a T_4 . Ao término da tarefa T_4 a lista tarefas-repetitivas será igual a seqüência $[T_1, T_2]$ e a lista seqüência-tarefas será igual a seqüência $[T_3, T_4]$. Assim, poderá ocorrer de duas situações:

- a) A terceira seqüência de tarefas for considerada similar às anteriores. Neste caso, se ao término da tarefa T_5 o agente identificará que as tarefas T_1 e T_3 são similares a T_5 , ele, independente da próxima tarefa, concluirá que estarão ocorrendo duas seqüências de tarefas repetitivas consecutivas, sendo elas $[T_1, T_2]$ e $[T_3, T_4]$.
- b) A terceira seqüência de tarefas não for considerada similar às anteriores. Neste caso, ao término da tarefa T_5 o agente identificará que ela não será similar à T_1 e T_3 . Sendo assim, ele poderá concluir que estarão ocorrendo duas seqüências de tarefas repetitivas consecutivas, sendo elas $[T_1, T_2]$ e $[T_3, T_4]$.

4) Seqüência de tarefas repetitivas não-consecutivas:

As repetições deste tipo poderão ser detectadas quando seqüências de tarefas repetitivas forem executadas não-sequencialmente. Por exemplo, considerando o contexto em que as tarefas T_1 , T_2 , T_3 , T_4 , T_5 , T_6 , T_7 e T_8 forem executadas consecutivamente, sendo que existe similaridades entre as tarefas: T_5 e T_1 ; T_6 e T_2 ; T_7 e T_3 . Assim, a lista tarefas-repetitivas igual a $[T_1, T_2, T_3]$ e a lista seqüência-tarefas será igual a $[T_5, T_6, T_7]$. Neste caso, se ao término da tarefa T_8 o agente verificar que T_4 não é similar a T_8 , a seqüência será quebrada e uma seqüência de tarefas repetitivas não-consecutivas será detectada, sendo ela composta de $[T_1, T_2, T_3]$ e $[T_5, T_6, T_7]$. Isto poderá

ocorrer em duas situações. Primeiro, se a tarefa T_8 não possuir tarefas similares. Segundo, se T_8 possuir tarefas similares e T_4 não estiver entre elas.

Assim, sempre que ocorrer uma detecção de tarefas repetitivas o agente criador de generalizações deverá ser acionado, procurando inferir o maior número de generalizações possíveis, considerando somente as repetições detectadas. Por outro lado, o agente detector deverá aguardar o processo de criação de extensão ser finalizado e o histórico de ações ser alterado para considerar uma nova ação e iniciar seu processo de detecção novamente.

É interessante notar que ao ser detectada uma seqüência de tarefas repetitivas mais de um tipo de objeto poderá estar sendo considerado na detecção, pois mais de uma tarefa está sendo considerada na repetição. Caso isto aconteça, o agente detector deverá agrupar as tarefas conforme o tipo do objeto manipulado e informar ao agente criador de inferências cada grupo isoladamente. Isto é necessário para que o agente criador de inferência possa considerar as particularidades de cada tipo de objeto.

4. O PROCESSO DE CRIAÇÃO DE GENERALIZAÇÕES

Na etapa de criação das generalizações, as aplicações *PhyD* podem empregar um mecanismo *ad hoc* ou um mecanismo baseado em técnicas de aprendizado de máquina. Segundo [7], as aplicações desenvolvidas de uma forma *ad hoc* contam com regras heurísticas para descreverem quando uma generalização é apropriada, como o *Eager* [2], o *Peridot* [12], o *Cima* [10] e o *Tinker* [8]. Estas heurísticas representam um sério problema, pois fazem com que os sistemas *PhyD* (como os sistemas especialistas baseados em regras nas quais eles são modelados) sejam fracos, difíceis de construir e difíceis de estender. Por exemplo, o *Peridot* usa um conjunto de regras ações-condições dependentes do domínio para inferir restrições e *loop*. Em contrapartida, as aplicações baseadas em técnicas de aprendizado de máquina contam com um generalizador do histórico de ações, que possibilita a criação de generalizações mais robustas e independentes de domínio. Deste modo, neste trabalho adotamos os mecanismos baseados em técnicas de aprendizagem de máquina e nos baseamos no algoritmo de aprendizado *Find-S* [11], modificado para inclusão de conhecimento prévio do domínio da aplicação, para desenvolvemos o processo de criação de generalizações. Assim, ao receber as tarefas que foram consideradas repetitivas pelo agente detector de repetições, presentes nas listas seqüência-tarefas e tarefas-repetitivas, o agente criador de generalizações agrupará as ações destas listas em duas seqüências de ações e tentará inferir generalizações, empregando para isto o algoritmo *Find-S* modificado. Por exemplo, sendo as tarefas T_1 e T_2 repetitivas, e cada uma possuindo as ações “ A_1 ” e “ A_2 ”, o agente criará as seqüências $[A_1, A_2]_1$ e $[A_1, A_2]_2$. Então, ele deverá considerar, primeiramente, a ação A_1 (e seus parâmetros) das duas seqüências, procurando inferir uma generalização como, por exemplo, a transformação de um parâmetro constante em uma variável, que satisfaça as duas ações, e assim por diante, até que todas as ações sejam consideradas. Logo, para cada ação A_i será criada

uma sub-hipótese H_i que representa a generalização desta ação.

Sempre que ocorrerem generalizações, o parâmetro da ação que foi generalizado, e seus valores, deverão ser armazenados em uma lista que conterá as variáveis criadas dinamicamente. Estas informações serão utilizadas para classificar estas variáveis e também possibilitarão que novos tipos de generalizações sejam criados posteriormente.

Após cada hipótese H_i ter sido criada para cada ação das seqüências, o agente criador de generalizações deverá verificar na lista de variáveis quais parâmetros foram generalizados para formatar as variáveis geradas. A formatação de uma variável determina o tipo da variável, realizando sua classificação segundo o tipo do parâmetro generalizado. Se o parâmetro for o objeto manipulado, o agente deverá pesquisar na BCDA se este objeto herda características de algum outro objeto. Se isto ocorrer, a variável deverá ser classificada como sendo do tipo do objeto pai. Este procedimento possibilita que a generalização possa considerar um conjunto maior de objetos. Caso contrário, a variável será do mesmo tipo que o parâmetro generalizado. Durante a formatação de uma variável, o agente deverá considerar as variáveis formatadas anteriormente no mesmo processo de criação de generalizações. Isto será necessário porque quando duas variáveis possuem os mesmos valores, elas apresentarão uma dependência entre si e, assim sendo, a última deverá ser classificada segundo a classificação da primeira.

Os passos realizados pelo agente criador de generalizações são apresentados no algoritmo da Figura 5. Nele podemos ver que após todas as sub-hipóteses H_i serem formatadas, elas deverão ser agrupadas para a criação da hipótese principal. Esta hipótese representa a generalização do conjunto de ações das tarefas repetitivas e deverá ser apresentada ao usuário para automatização de seu trabalho. No entanto, a disponibilidade de conhecimento prévio do domínio da aplicação, por meio da BCDA, possibilita que usemos esta hipótese como ponto de partida para a geração de um conjunto de generalizações mais rico em estruturas de controle. Deste modo, poderão ser geradas generalizações que utilizam: 1) comandos condicionais, quando objetos ou atributos diferentes forem usados com a mesma ação; 2) comandos de repetição, quando ocorrer dos objetos envolvidos nas tarefas apresentarem um relacionamento de herança ou de partonomia; e também 3) comando com repetições aninhadas, quando dois objetos diferentes forem considerados em uma mesma repetição e eles possuírem o relacionamento de partonomia, isto é, um objeto fazer parte do outro.

Para classificarmos os tipos de inferências que podem ser criadas nesta etapa, consideramos os tipos de repetições analisados por [17]. Os tipos de generalizações que o agente criador de generalizações pode gerar são:

1) **Constante:** é criada quando todas as iterações de repetições são idênticas. Um exemplo é a inserção de tabelas com um número específicos de linhas e colunas em um editor de texto. Se a generalização abaixo fosse executada em um editor de texto uma

tabela com dez linhas e três colunas seria inserida em um texto. Este tipo de generalização é equivalente às macros atualmente empregadas nas aplicações comerciais.

```
inserir(_Tabela, numero_linhas = 10,
numero_colunas = 3)
```

2) **Variável simples ou dependente:** é criada quando as iterações de repetições apresentam variações. Um exemplo de generalização variável simples é a primeira generalização criada no exemplo da Figura 2. Porém, para a execução desta generalização o usuário deverá informar ao agente automatizador de extensões, por meio de um diálogo de aquisição de dados, o estilo da fonte da palavra ‘correção’ que ele deseja “apagar”. A generalização variável dependente é semelhante à simples, porém as variáveis criadas nelas são dependentes de uma característica do objeto manipulado. Um exemplo seria o usuário “salvar” mensagens de uma ‘caixa de entrada’ de um *e-mailer* com o ‘nome do remetente’. Logo, a generalização criada será:

```
selecionar(_Mensagem, _Remetente)
salvar_como(_Mensagem, _Remetente,
extensão = txt)
```

Se esta generalização fosse executada uma mensagem seria armazenada em um texto com o nome do remetente da mensagem.

Generalizações interativas, isto é, generalizações que exigem a criação de um diálogo de aquisição de dados entre a aplicação e o usuário para que elas possam ser ativadas, são interessantes quando várias tarefas são consideradas na repetição, pois caso contrário o número de ações para ativar a inferência será o mesmo, ou até maior, do que o número de ações que o usuário deve realizar para a execução da tarefa repetitiva. É importante ressaltar que qualquer tipo de generalização, exceto a constante, pode exigir este diálogo de aquisição de dados.

3) **Loop simples ou dependente:** é criado quando as iterações compreendem as mesmas ações, mas são repetidas sobre um conjunto de objetos. Para a detecção deste conjunto o agente criador de generalizações consulta a BCDA para averiguar se há o relacionamento de herança ou o de partonomia entre os objetos considerados na repetição detectada. Um exemplo de *loop* simples é a terceira inferência criada no exemplo da Figura 3. Por meio do relacionamento de partonomia entre o objeto ‘palavra’ e ‘parágrafo’ existente na ontologia do domínio na BCDA, o agente pode considerar o conjunto de ‘palavras’ que fazem parte dos ‘parágrafos’. Já, a inferência *loop* dependente será criada quando uma generalização variável dependente for criada e o objeto considerado na generalização fizer parte de um outro objeto do domínio. A generalização abaixo poderá ser criada se na ontologia do domínio de um *emailer* o agente averiguar que o objeto ‘mensagem’ tem o relacionamento de partonomia com o objeto ‘pastas’. Se esta generalização fosse executada todas as mensagens pertencentes à ‘pastas’ seriam

```

Criar-generalizações
entrada:
    seqüência-tarefas      % são as tarefas da lista sequencia-tarefas
    tarefas-repetitivas   % são as tarefas da lista tarefas-repetitivas utilizadas no
                          % algoritmo anterior com suas respectivas ações, objetos
                          % manipulados e parâmetros

variáveis locais:
    ATi,                % atributo (objeto manipulado ou parâmetro) de uma ação
    Hi,                % sub-hipótese
    lista-variáveis,      % lista que armazena atributos generalizados
    hipótese-principal,   % variável que armazena a hipótese principal criada
    lista1, lista2,    % listas utilizadas para formar uma estrutura condicional
    S1,                % listas com as ações da lista seqüência-tarefas
    S2,                % listas com as ações da lista tarefas-repetitivas
    ATj'              % variável que indica o tipo do atributo (objeto ou parâmetros)
    Vi,                % variável que indica um elemento da lista-variáveis
    I1, I2            % se dois objetos são considerados em uma mesma repetição,
                          % primeira variável indica generalizações criadas para o primeiro
                          % objeto e a segunda, generalizações para o segundo objeto

Início
% Criar generalização principal
S1, S2 = criar-seqüências-ações(seqüência-tarefas, tarefas-repetitivas)
Para cada ação Ai de S1 faça
    inicializar a sub-hipótese Hi com a ação Ai                % Hipótese mais restrita
    Para cada atributo ATj em Hi faça
        Se ATj não é satisfeito por ATk de Ai em S2 com j = k então
            ATj' = relaxar (ATj) em Hi para satisfazer ATk de Ai em S2
            inserir([ATj', valor(ATj), valor(ATk)], lista-variáveis)
        % Classificação das variáveis da sub-hipótese Hi gerada acima
        Para cada Vi da lista-variáveis faça
            Se ATi = objeto-manipulado então
                Se objeto-manipulado possui relacionamento de herança então
                    tipo-variável = tipo(superclasse)
                Senão Se Vi de ATi já foi considerada como variável em outra sub-hipótese então
                    tipo-variável = tipo(variável-anterior)
                Senão
                    tipo-variável = tipo(ATi)

            inserir(Hi, hipótese-principal)
% Criar generalizações com condicionais, se possível
Se lista-de-variáveis ≠ Vazio então
    Para cada Vi da lista-de-variáveis faça
        adicionar(formatar(valor1), lista1)
        adicionar(formatar(valor2), lista2)
    %Criar inferência
    concatenar('If', lista1, 'or', lista2, 'then', hipótese-principal)
% Criar generalizações com repetições, se possível
Se relacionamento-herança(objetos-manipulados na repetição) ou
    relacionamento-partonomia(objetos-manipulados na repetição) então
    % Criar generalização
    concatenar('For each', tipo-objeto(objetos-manipulados), 'of the',
        objeto que é container ou superclasse do tipo-objeto,
        'do', hipótese-principal)
% Criar generalizações com repetições aninhadas, se possível
Se dois objetos foram considerados numa mesma repetição então
    Se o segundo objeto faz parte do primeiro então
        Para cada generalização G1 criada para o primeiro objeto faça
            Para cada generalização G2 criada para o segundo objeto faça
                Se tipo(G1) = tipo(G2) então
                    Agrupar generalizações

```

Figura 5- Algoritmo do processo de criação de generalizações.

armazenadas na forma de texto com o nome de seu remetente.

```

For each _Mensagem of the _Pastas do
    selecionar(_Mensagem, _Remetente)
    salvar_como(_Mensagem, _Remetente,
        extensão = txt)

```

4) **Loop condicional:** é criado quando as iterações são idênticas, mas são repetidas sobre um conjunto de objetos que exibem propriedades diferentes. Um exemplo de *loop* condicional é a segunda generalização inferida no exemplo da Figura 3. Esta generalização, quando for executada, apagará a

palavra ‘correção’ que atender todas as restrições impostas pela ação “selecionar” e possuir o ‘estilo da fonte’ igual a ‘normal’ ou ‘itálico’. Este tipo de generalização pôde ser criado porque os atributos dos objetos manipulados foram considerados e existiam diferenças entre eles.

5) **Loops aninhados:** é criada quando as iterações são idênticas, mas são embutidas em dois *loops*. Este aninhamento ocorrerá quando dois objetos forem considerados em uma mesma repetição e o segundo fizer parte do primeiro. Um exemplo ocorrerá um usuário realizar tarefas repetitivas como abrir um

‘documento’ de um editor de texto e “mudar o estilo da fonte” de um ‘parágrafo’ do texto. Sendo assim, o agente detector perceberá que repetições estão sendo executadas e que os dois objetos considerados nela, o ‘documento’ e o ‘parágrafo’, possuem um relacionamento de partonomia. Neste caso, na BCDA, o objeto ‘documento’ deve apresentar o relacionamento de partonomia com o objeto ‘pasta corrente’ e o objeto ‘parágrafo’ com o objeto ‘documento’. Logo, o agente criador de generalizações inferirá a seguinte generalização:

```
For each _Documento of the _Pasta_corrente do
  abrir(_Documento, nome=carta, _Extensão)
  For each _Parágrafo of the _Documento do
    selecionar(_Parágrafo, estilo=normal)
    mudar_estilo(_Parágrafo, estilo=itálico)
```

Se esta generalização fosse executada todos os ‘parágrafos’ com o estilo de fonte ‘normal’ de todos os ‘documentos’ que fizerem parte da ‘pasta corrente’ passariam a ter o estilo ‘itálico’.

Após todas as generalizações serem inferidas elas deverão ser exibidas ao usuário para que ele possa entender o modelo conceitual da aplicação facilitando a realização de extensões futuras [9]. A exibição deverá ser de uma forma não intrusiva, para não interromper o trabalho dos usuários, ou seja, para não comprometer a usabilidade da aplicação. Para isto, sempre que houver novas generalizações deverá ocorrer uma sinalização discreta aos usuários como ocorre no APE [16]. Caso contrário, a atenção do usuário será desviada bruscamente e o seu fluxo de trabalho será quebrado. Por exemplo, em um ambiente como o *Windows* uma janela poderia aparecer minimizada na barra de ferramentas e sinalizar por meio de mudanças de cores da janela minimizada para indicar que novas generalizações foram criadas para automatizar o trabalho dos usuários. Se um usuário possuir o interesse de utilizar este recurso ele poderá maximizar a janela e selecionar alguma generalização em uma lista que conterá todas as generalizações que foram criadas naquele momento para a repetição mais recente, caso contrário poderá continuar executando o seu trabalho normalmente.

5. DISCUSSÃO

A aplicação APE [16], até onde vai nosso conhecimento, é a aplicação *PbyD* implícita de maior relevância existente na literatura atual. Deste modo, vamos considerá-lo para comparar os resultados alcançados na pesquisa que desenvolvemos. Para esta comparação, primeiramente, consideramos o processo de detecção de repetições e, em seguida, o de criação de generalizações.

No processo de detecção de repetições o APE é capaz de detectar tanto ações quanto seqüências de ações, sendo elas consecutivas ou não-consecutivas. Porém, é incapaz de detectar repetições se as ações não forem executadas exatamente na mesma ordem, por utilizar um algoritmo simples de detecção de padrões em *strings*, o KMR [5]. Assim, ele detecta somente repetições exatas. O processo de detecção de repetições apresentado neste trabalho também emprega um algoritmo simples que procura detectar a igualdade entre as ações das tarefas realizadas

por um usuário na interface de uma aplicação. Além de considerar todos os tipos de repetições citados acima, ele também detecta repetições “ruidosas”, ou seja, seqüências de ações que foram interrompidas por algum ruído (ação). Entretanto, as ações das repetições devem ser as mesmas. Esta vantagem advém do fato de considerarmos as interdependências entre as ações (tarefas) que podem ser executadas em um determinado objeto do domínio, isto é, do emprego de conhecimento prévio do domínio da aplicação.

Um outro ponto que vale a pena ressaltar em nosso processo de detecção é a minimização do trabalho do agente detector. Com o uso de conhecimento prévio, ou seja, a noção de tarefas, o agente precisa ativar o seu processo de detecção de tarefas repetitivas somente quando uma tarefa for finalizada, e não quando uma ação for executada, como ocorre nas aplicações *PbyD* implícitas existentes.

No processo de criação de generalizações tanto o APE quanto o nosso trabalho se basearam em algoritmos de aprendizado de conceitos, o APE no algoritmo de eliminação de candidatos e o nosso no algoritmo *Find-S* [11]. O APE é capaz de criar três tipos de generalizações: constante, *loops* e *loops* aninhados. Já o processo de criação de generalizações apresentado neste artigo é capaz de criar sete tipos de generalizações, sendo eles: constante, variável simples e dependente e *loop* simples, aninhados, condicional e dependente. Logo, haverá uma maior chance da intenção real dos usuários ser atendida, pois um maior espectro de situações cotidianas poderá ser reconhecido. Porém, deve-se ter o cuidado para que um número reduzido de generalizações seja exibido aos usuários, para que o processo de criação de extensões não comprometa a produtividade dos usuários. As generalizações do tipo *loop* condicional e *loop* variável somente puderam ser considerados devido ao uso de conhecimento prévio do domínio da aplicação que tornou possível ao processo de criação de generalizações considerar dissimilaridades entre os valores dos parâmetros das ações e dos objetos manipulados pelas ações.

É importante salientar que existem duas limitações em nosso trabalho atual. Primeira, no processo de detecção de repetições, somente uma tarefa pendente é considerada de cada vez. Segunda, no processo de criação de generalizações, no máximo duas repetições que possuam atributos diferentes podem ser consideradas em cada processo de generalização. Acreditamos que estas limitações possam ser solucionadas na continuação deste trabalho, exigindo alguns aprimoramentos na implementação dos agentes inteligentes.

5.1 CONCLUSÃO

Conforme procuramos demonstrar ao longo deste artigo, o uso de conhecimento prévio do domínio da aplicação nos processos de detecção de repetições e de criação de generalizações permite detectar um número maior de repetições, e também permite que generalizações mais elaboradas (considerando condicionais e a dependência entre as variáveis) sejam disponibilizadas aos usuários. Sendo assim, podemos dizer que a usabilidade de uma aplicação extensível, que utiliza o mecanismo de *PbyD*

implícito apresentará uma evolução quando empregamos conhecimento prévio do domínio em seu mecanismo de criação de extensões e os usuários poderão encontrar uma maior facilidade para automatizarem as suas tarefas cotidianas, visto que as generalizações geradas estarão mais próximas de sua intenção original.

No entanto, com o uso de conhecimento prévio do domínio da aplicação o processo de detecção de repetições pode ir além das detecções de repetições que consideramos neste artigo, ou seja, da detecção de repetições que possuem as mesmas ações executadas na mesma ordem, com ruído ou não. De acordo com [17], repetições não-ordenadas e com ações diferentes entre as suas tarefas poderão ser detectadas, considerando um histórico estruturado em multi-níveis como proposto em [6]. Ambas os tipos possibilitariam um maior número de detecções e conseqüentemente um maior número de criação de generalizações para a automatização das tarefas dos usuários. A detecção de repetições com ações diferentes possibilitaria a criação de *loops* condicionais e *loops* dependentes mais elaborados, isto é, não considerando somente os parâmetros das ações. Sendo assim, o detector poderá considerar ações de alto nível em seu processo e detectar repetições em iterações que possuam ações iguais em ordens diferentes ou que possuam ações diferentes.

Além disso, uma aplicação também poderia ordenar as generalizações conforme as preferências dos usuários podendo assim aumentar a sua produtividade, melhorando conseqüentemente a usabilidade da aplicação. Para isto o agente criador de generalizações poderia fazer uso de algum outro tipo de aprendizado, por exemplo, o aprendizado Bayseano [15], para que as preferências dos usuários pudessem influenciar a ordem de apresentação das generalizações. Se um usuário na maioria das vezes aceitasse uma generalização do tipo *loop* condicional, sempre que o agente fosse exibir generalizações ele poderia colocar este tipo em primeiro lugar, pois haveria uma chance maior dela ser escolhida e o trabalho do usuário ser automatizado de uma forma mais rápida.

Estas possibilidades deverão ser exploradas na continuidade deste trabalho, assim como a eliminação de algumas limitações menores não exploradas por restrições de tempo.

REFERÊNCIAS

1. Cypher, A. Watch What I Do: Programming by Demonstration. Cambridge : The MIT Press, 1993.
2. Cypher, A. Eager: Programming repetitive tasks by demonstration. In: Cypher, A. (ed) Watch What I Do: Programming by Demonstration. Cambridge : The MIT Press, 1993. p. 205-217.
3. Da Silva, S. R. P. Um Modelo Semiótico para Programação por Usuários Finais. 2001. Tese (Doutorado) - PUC-Rio, Rio de Janeiro, 2001.
4. De Souza, C.S.; Leite, J.C.; Prates, R.O.; Barbosa, S.D.J. “Projetos de Interfaces de Usuário:

Perspectivas Cognitivas e Semióticas”. Anais do XIX Congresso Nacional da Sociedade Brasileira de Computação. Jornada de Atualização em Informática. SBC’99. Julho de 1999.

5. Karp, R. M.; Miller, R. E.; Rosenberg, A. L. Rapid Identification of Repeated Patterns in Strings, Trees and Arrays. Proc. of the Fourth Annual Symposium on Theory of Computing. Denver, ACM Press, 1972. p 1-3.
6. Kosbie, D. S.; Myers, B. A. A System-Wide Macro Facility Based on Aggregate Events: A Proposal. In: CYPHER, A (ed) Watch What I Do: Programming by Demonstration. Cambridge : The MIT Press, 1993. p. 433-444.
7. Lau, T.; Weld, D. Programming by Demonstration: An Inductive Learning Formulation. Proc. of the 4th International Conference on Intelligent User Interfaces, Los Angeles, p. 145-152, 1999.
8. Lieberman, H. Tinker: A Programming by Demonstration System for Beginning Programmers. In: CYPHER, A. (ed) Watch What I Do: Programming by Demonstration. Cambridge : The MIT Press, 1993. p. 49-64.
9. Malsby, D.; Witten, I. H.; Nevill-Manning, C. G. Interacting with learning agents: Implications for ML from HCI. In: Workshop on Machine Learning Meets Human-Computer Interaction, p. 51-58, 1996.
10. Malsby, D.; Witten, I. H. Cima: An Interactive Concept Learning System for End-User Applications. Applied Artificial Intelligence, 11, 1997. p. 653-671.
11. Mitchell, T. Machine Learning. The McGraw-Hill, 1997. 414 p.
12. Myers, B. A. Peridot: Creating User Interfaces by Demonstration. In: Cypher, A. (ed) Watch What I Do: Programming by Demonstration. Cambridge : The MIT Press, 1993. p. 125-153.
13. Norman, D. A. How might people interact with agents. In: Communications of the ACM, 37(7), p. 68-71, 1994.
14. Penteado, R. R. M. O Uso de Conhecimento Prévio do Domínio em Aplicações na Programação por Demonstração, 2004. Dissertação (Mestrado) – UEM, Maringá, 2004.
15. Russel, S.; Norvig, P. Artificial Intelligence: a modern approach, 2. ed.. Prentice-Hall, 2002.
16. Ruvini, J. D.; Dony, C. Learning Users’ Habits to Automate Repetitive Tasks. In: Lieberman, H. Your Wish Is My Command: Programming By Example, Boston : The Morgan Kaufmann, 2001. p. 271-295.
17. Ruvini, J. D. The Challenges of Implicit Programming by Example. In: IUI’04, 2004, Madera, Funchal, Portugal, p. 13-16.

TFlex: Um Simulador de Teclado com Múltiplos Modos de Varredura

Marcio Juliato

Eduardo H. Tanaka

Christian Baudet

Miguel Galves

Thiago T. Coelho

Heloísa Vieira da Rocha

Instituto de Computação (IC) – Universidade Estadual de Campinas (UNICAMP)
Avenida Albert Einstein, 1251, Caixa Postal 6176, CEP13084-971 Campinas, SP – Brasil

{marcio.juliato, etanaka, christian.baudet, miguel.galves, thiago.coelho, heloisa}@ic.unicamp.br

RESUMO

Atualmente, existem inúmeros periféricos de entrada para computadores. Contudo, nem todas as pessoas podem utilizar os dispositivos mais comuns como o teclado e o mouse. Este é o caso de algumas pessoas portadoras de deficiências físicas e motoras. Para este público, uma alternativa é um simulador de teclado. Todavia, os simuladores de teclado atuais apresentam restrições que nem sempre podem ser superadas, não garantindo um bom nível de autonomia e independência dos usuários. Portanto, propõe-se um novo modelo de simulador de teclado, denominado TFlex, sem as restrições apresentadas até então em softwares similares. Pelo modelo proposto, um usuário portador de necessidades especiais pode ter completa independência no uso do computador utilizando esse simulador de teclado. Além disso, tentativas para melhorar a eficiência de uso foram implementadas, como diferentes modos de varredura e layouts de teclado.

ABSTRACT

Nowadays, there is a number of computer input peripherals, however, not every person can use the most common input devices such as keyboard and mouse. This is the case of some people with physical and motor disabilities. For these people a keyboard simulator is a good alternative, but the current keyboard simulators have restrictions, which not always can be overcome, not assuring a good user's independence level. Therefore, a new keyboard simulator model is proposed, called TFlex, without the restrictions presented in others similar software. With the proposed model, a user with special needs could have total independence in computer while using TFlex. Besides, approaches to improve user's efficiency were developed, with different scan modes and keyboard layouts.

Palavras Chave

Teclado Virtual, Simulador de Teclado, Tecnologias Assistivas

INTRODUÇÃO

A Organização Mundial de Saúde (OMS) estima que 10% da população mundial é portadora de deficiência. No Brasil, dados do censo demográfico de 2000 do IBGE [4] apontam para mais de 14% da população como de portadores de deficiência, sendo que há mais de 7 milhões de pessoas com algum tipo de deficiência motora e quase um milhão e meio de portadores de deficiência física.

Apesar desses números serem altos, ainda há poucas iniciativas para integrar as pessoas portadoras de deficiência na sociedade. Ruas e avenidas não permitem a locomoção de usuários de cadeiras de rodas pela falta de guias rebaixadas. Nas bibliotecas, é muito difícil encontrar livros em Braille ou falados para permitir a “leitura” dos cegos. Soma-se a este descaso o preconceito e a imagem negativa que a sociedade faz da pessoa portadora de deficiência, considerando-a um cidadão especial no sentido negativo (não “normal”), incapaz de trabalhar ou estudar, conduzindo a ações segregadoras ou paternalistas.

Da mesma forma que o mundo físico ainda é uma barreira para os portadores de deficiência, o mundo virtual também o é. As tecnologias de informação e comunicação (TICs) contribuem ainda mais para a exclusão. Contudo, assim como em um prédio é possível construir rampas e elevadores para os usuários de cadeiras de rodas, nos computadores existem adaptações ou recursos capazes de promover a acessibilidade, conhecidas como tecnologias assistivas. Hogetop e Santarosa [3] definem tais tecnologias como o conjunto de recursos que, de alguma maneira, contribuem para proporcionar maior independência e qualidade de vida para os portadores de deficiência, contribuindo também para a inclusão.

Dentre as diferentes tecnologias assistivas existentes, encontra-se o simulador de teclado. Frequentemente, os simuladores de teclado apresentam na tela do computador uma representação de um teclado físico, isto é, um teclado virtual.

Contudo, um simulador de teclado se diferencia de um teclado virtual. Um teclado virtual apenas apresenta as teclas de um teclado na tela do computador e é utilizado, por exemplo, em websites de bancos para aumentar a segurança na entrada de dados, permitindo a entrada de dados através de dispositivos de apontamento como um mouse. Já um simulador de teclado, além de permitir a

escrita via dispositivo de apontamento, possui um sistema de varredura entre as teclas mostradas na tela e permite a usuários que possuem restrições motoras graves o uso de praticamente todas as opções disponíveis em um teclado com o pressionamento de uma única tecla do teclado físico ou um *switch* ou comutador (que nada mais é do que um botão ou chave), bastando que a varredura se encontre na tecla desejada. A Figura 1 mostra alguns comutadores [5] que podem ser utilizados junto com simuladores de teclado e outras tecnologias assistivas de software.



Figura 1. Exemplo de comutadores.

Além da varredura, outros recursos que frequentemente estão incluídos em simuladores de teclado são sistemas de predição de palavras e “leitor” de palavras, que lê aquilo que acabou de ser escrito. Entretanto, a maioria dos simuladores de teclado existentes atualmente sofre restrições diversas, mesmo sabendo que sua implementação é considerada simples.

ESTADO DA ARTE

Existem inúmeros simuladores de teclado no mercado e outros dispositivos de hardware e software que podem substituir o teclado físico. A seguir, serão analisados alguns dos mais populares entre eles.

Teclado Virtual do Windows XP

Apesar do nome, o Teclado Virtual presente no Windows XP possui um sistema de varredura simplificado. Ele é apresentado na tela como um teclado padrão QWERTY com 101/102 teclas, contendo todas as teclas de um teclado (incluindo acentos, teclas de função F1-F12, Caps Lock, Shift, Alt, Ctrl e até mesmo a tecla do menu Iniciar do Windows). As vantagens deste teclado sobre os demais é que ele acompanha o sistema operacional, não sendo preciso a compra e/ou a instalação de outros softwares, e é totalmente integrado ao sistema operacional, funcionando com praticamente qualquer software para Windows.

Há como configurar o teclado para não mostrar o teclado numérico, reduzindo os estímulos visuais e a quantidade de itens a serem percorridos pela varredura, e também é possível acionar uma opção para emitir som de clique quando se pressiona uma das teclas. Mas, diferente de um teclado físico comum em que se pode enxergar que no número 1 está o ponto de exclamação (!) quando se pressiona o Shift, no teclado virtual este e outros símbolos só irão aparecer após o pressionamento do Shift. A varredura acontece primeiro entre as linhas do teclado. Selecionada uma linha, a varredura se dá então entre as

colunas/teclas desta linha. A Figura 2 mostra a tecla *j* sendo percorrida pela varredura.



Figura 2: Teclado Virtual do Windows XP.

Quanto à varredura do teclado, só é possível configurar o tempo de varredura entre 0,5 e 3 segundos e de que dispositivo (joystick, porta serial, ou uma tecla do teclado físico, como a barra de espaço) deve-se capturar o sinal que indica a seleção da opção sendo varrida. A própria Microsoft admite que este teclado é limitado e que, para uso freqüente, o usuário deveria procurar outros teclados com mais funcionalidades.

Teclado Amigo

O Teclado Amigo é um software nacional, desenvolvido por pesquisadores do Núcleo de Computação Eletrônica da Universidade Federal do Rio de Janeiro (NCE-UFRJ), sendo distribuído gratuitamente pela Rede SACI [11] como “Kit SACI 2”. O Teclado Amigo é um conjunto de programas que possui diferentes teclados virtuais com varredura. Estes programas são uma calculadora, um editor de textos, e um teclado básico que permite apenas a escrita de pequenas frases.

A velocidade da varredura no teclado segue uma escala numérica de zero a quatro (quanto maior o número selecionado, maior a velocidade). Porém, ao abrir a janela de configuração de velocidade de varredura, não fica visível qual é a velocidade atualmente em uso. Na Figura 3 observa-se o Teclado Básico. A disposição das teclas é, no mínimo, inédita e única. Separou-se entre subgrupos contendo vogais, consoantes, números e símbolos diversos (pontuação, sinais matemáticos, parênteses e outros). Também há teclas com vogais já acentuadas. Ilustrando falta de consistência, há alguns botões com letras em maiúsculas e outros com letras em minúsculas e sem nenhuma razão para tal diferenciação.



Figura 3. Teclado básico do Teclado Amigo.

O Teclado Básico e o Editor de Textos do Teclado Amigo possuem abreviações nem sempre fáceis de serem inferidas (ZUPT, ABRV, FALA, por exemplo). Além de não funcionar com outros programas, outro ponto negativo de todo o conjunto de programas do Teclado Amigo é que, para indicar qual é a linha ou tecla que está sendo varrida, os programas movem o cursor do mouse,

praticamente impedindo o uso do mouse concomitantemente com estes programas.

Click-N-Type

Outro software distribuído gratuitamente, o Click-N-Type da Lakefolks [6] é um teclado com varredura similar ao Teclado Virtual do Windows XP e funciona em qualquer programa deste sistema operacional. O Click-N-Type, apesar de ser em Inglês, possui um pacote que traduz a maioria das mensagens e textos nos menus para uma mistura entre o Português de Portugal e do Brasil. Também, há muitas opções para customização.

Além da tradicional possibilidade de configuração da velocidade da varredura (entre 0 e 100, sendo 100 o mais rápido), é possível configurar o layout do teclado e escolher quais teclas devem ser mostradas e a ordem das teclas (alfabética, QWERTY, ou qualquer outra). Junto com o software, acompanham layouts de teclados QWERTY de 101 e 102 teclas e um teclado de tamanho reduzido, indicado principalmente para uso com o sistema de varredura. Há para download outros layouts, inclusive o ABNT2 utilizado no Brasil.

O sistema de varredura é o que o difere dos softwares anteriores. No Click-N-Type, existe um retângulo de contorno vermelho (cor padrão, possível de ser alterada) que se movimenta por blocos de teclas do teclado. Quando se pressiona um botão do teclado ou o botão esquerdo do mouse ou ainda um comutador ligado ao computador, uma linha vermelha passa a se mover horizontalmente dentro do retângulo. Outro pressionamento faz uma outra linha vermelha se mover verticalmente. Mais um pressionamento faz com que a tecla do teclado virtual na intersecção das linhas vertical e horizontal seja “clificada”, como ilustrado na Figura 4.

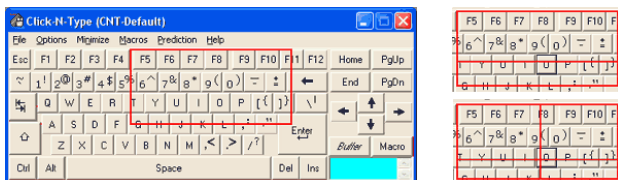


Figura 4. Telas do Click-N-Type.

Percebe-se que este tipo de varredura acaba produzindo mais pressionamentos de botões do que o do Teclado Virtual do Windows XP ou dos programas do Teclado Amigo. Outro ponto negativo é que o retângulo de varredura se move entre blocos de teclas e nem sempre as teclas ficam totalmente dentro do retângulo, exigindo que o usuário tenha uma certa coordenação motora para acertar a intersecção na tecla desejada. E, assim como o Teclado Amigo, enquanto a varredura estiver ativada no Click-N-Type, o cursor do mouse fica desabilitado. Apesar de não incluir em seu pacote básico, existe um meio de instalar um sistema de predição de palavras em Português.

WiVik

Outro simulador de teclado com alguma similaridade com o Click-N-Type, o WiVik [14] tem como diferencial a possibilidade de “ler” letras, palavras ou frases na medida em que são escritas, dando um feedback sonoro ao usuário. Este software, assim como o Click-N-Type,

permite customizar totalmente o teclado virtual, inclusive colocando botões para, por exemplo, mover e redimensionar janelas, varrer opções de menus e atalhos como copiar e colar (Ctrl+C e Ctrl+V, respectivamente).

O WiVik é um software totalmente em Inglês que não oferece nenhuma forma de alteração do idioma. Também, o software pode não funcionar em todos os programas para Windows, como salienta o fabricante. Por fim, a licença para um único usuário custa US\$ 350,00.

Simulador de Teclado – ST(NIEE)

Este simulador de teclado é um dos pioneiros no Brasil. A interface é baseada em modos “semi-gráficos” (gráficos gerados a partir de caracteres de texto), tipicamente comuns em programas baseados no sistema operacional MS-DOS. Trata-se, na verdade, de uma adaptação de um projeto da Universidade Técnica de Madri realizada por uma equipe do Núcleo de Informática na Educação Especial da Universidade Federal do Rio Grande do Sul [9].

Por se tratar de um software desenvolvido há muito tempo, logo na instalação o usuário é questionado sobre qual a porta serial será usada pelo acionador. Caso tenha sido feita uma escolha por engano ou mesmo se queira instalar um acionador em outra porta serial, será necessário reinstalar o software. A varredura implementada no Simulador de Teclado do NIEE é a de linhas por colunas, mas as letras são dispostas na tela seguindo um layout diferente do QWERTY. E, como não havia espaço na tela do MS-DOS para dispor todas as opções, existem sub-janelas para escolha entre vogais com acentos, números e símbolos matemáticos, teclas de função (F1-F12), e teclas para navegação no texto (home, end, page up, page down, etc.).

A pequena área reservada para ver a escrita do texto no Simulador de Teclado do NIEE só possui espaço para uma única linha, fazendo com que o usuário use muito das teclas de navegação pelo texto. A Figura 5 mostra uma tela deste simulador de teclado. Outro ponto negativo, por causa da idade do software, é que ele só consegue interagir com outros programas do MS-DOS. Mesmo assim, ao menos uma característica presente neste simulador de teclado é rara em outros mais recentes: a varredura por todas as opções de configuração, permitindo que o usuário possua independência e autonomia para configurar o software.

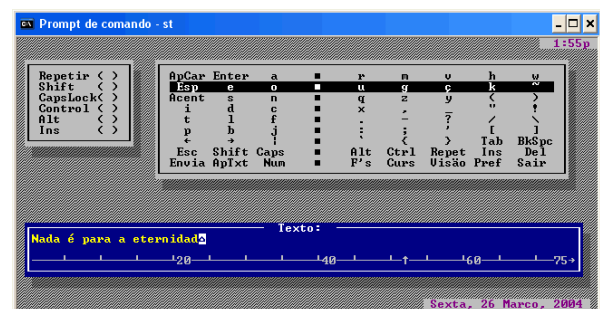


Figura 5. Simulador de Teclado do NIEE.

Dasher

O Dasher [7] não pode ser considerado nem um teclado virtual nem um simulador de teclado. Contudo, a sua finalidade é a mesma: permitir a escrita no computador através de um meio alternativo ao teclado físico. Ele é uma interface para entrada de textos que utiliza um modelo de predição de palavras (em diversos idiomas, inclusive português) para tornar a escrita eficiente. Sua tela pode ser observada na Figura 6.

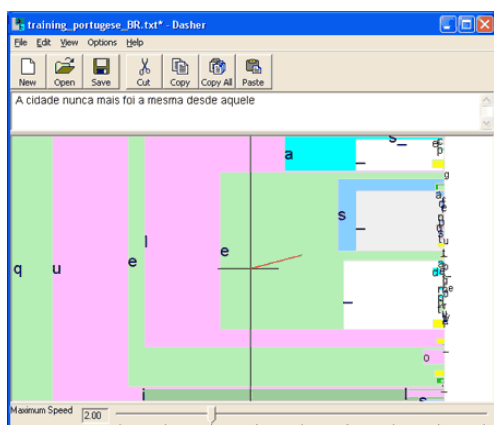


Figura 6. Tela do Dasher.

O software pode ser utilizado via mouse ou qualquer outro dispositivo que substitua o mouse, como um *eye tracker* (o autor do software afirma que a versão para *eye tracker* possibilita escrever tão rápido quanto por digitação normal, sendo também mais rápido do que teclados virtuais e simuladores de teclado). Outra vantagem do Dasher é que ele é um software livre e seu código-fonte pode ser modificado e distribuído.

A principal desvantagem do Dasher é que ele exige que o usuário tenha condições de mover um dispositivo de apontamento. Entretanto, para o público comum dos simuladores de teclado (pessoas com restrições motoras, principalmente nos membros superiores), o único tipo de dispositivo de apontamento que garantidamente atende a esta exigência seria um *eye tracker*, que é um dispositivo de rastreamento ocular de preço elevado.

Via Voice e outros sistemas de reconhecimento de voz

É possível substituir o teclado e o mouse através de um sistema de reconhecimento de voz. A maioria destes softwares é de uso pessoal, já que vai se adaptando à voz do usuário. Depois de um período em que o software é treinado para reconhecer e entender a fala do usuário, praticamente todos os comandos do computador podem ser dados através do microfone. Há dois problemas no uso destes sistemas. Primeiro, a língua utilizada é predominantemente o Inglês. Segundo, nem todos são gratuitos, como é o caso do ViaVoice, da IBM.

A PROPOSTA

Como visto, os simuladores de teclado e teclados virtuais existentes hoje são projetos que limitam o uso por pessoas portadoras de deficiência física ou motora. E mesmo outros softwares como o Dasher e o ViaVoice, que podem substituir um teclado convencional, também possuem restrições. Estas restrições podem ser devido à

falta de integração com o sistema operacional, impossibilitando o uso junto com outros programas, interferências com o mouse e o teclado físicos, ausência de varredura nas janelas de configuração, layouts inapropriados e custo elevado.

Entretanto, nota-se que o funcionamento de um simulador de teclado é bem simples e a sua implementação, a princípio, não pode ser considerada muito complexa, mas com enormes desafios de interface.

Assim, propõe-se o desenvolvimento de um simulador de teclado, o TFlex [12], para uso por pessoas portadoras de deficiência física ou motora, tendo o mesmo, recursos configuráveis que aumentam a facilidade e eficiência de uso.

Pelas condições culturais e sócio-econômicas do país, passa a ser interessante a distribuição do mesmo sem qualquer custo ao usuário, e em Português. Outro fator importante é que ele não seja difícil de instalar, configurar, aprender e usar.

A forma mais comum e tradicional de varredura em simuladores de teclado é a baseada em linhas e colunas, como se o teclado se comportasse como uma matriz. Neste tipo de abordagem, primeiro seleciona-se a linha em que a tecla desejada se situa, sendo em seguida realizada uma busca entre as colunas desta linha até que o caractere almejado seja encontrado.

Porém, para se chegar a um caractere desta matriz não necessariamente deve-se seguir essa abordagem tradicional, sendo possível adotar outro tipo de varredura. Uma aproximação razoável seria seguir um algoritmo de busca binária para encontrar, através de divisões sucessivas, a tecla desejada.

No projeto do TFlex, esses dois modos de varredura foram implementados, sendo os mesmos descritos nas seções a seguir. Além disso, foram estudados e implementados layouts alternativos ao QWERTY, como o alfabético, por exemplo, com o intuito de agilizar o processo de escrita e/ou facilitar a localização do caractere a ser digitado.

LAYOUTS DO TECLADO

O layout QWERTY surgiu no final do século XIX, há 130 anos, junto com a invenção das primeiras máquinas de escrever. Desde então, é considerado o padrão universal para teclados não só de máquinas de escrever como também de computadores e outros dispositivos, embora outros layouts tenham sido propostos no século XX. O nome QWERTY vem das 6 primeiras letras dispostas na primeira linha do teclado [2].

Por ser considerado um padrão universal, utilizar o layout QWERTY no TFlex facilitaria o reconhecimento da disposição das teclas por usuários que já conhecem este layout. Mas o TFlex poderia utilizar um layout alfabético (com as teclas de letras dispostas em ordem alfabética), facilitando a aprendizagem e o uso por pessoas pouco acostumadas com um layout QWERTY. Ainda, poder-se-ia pensar em um layout, baseado na frequência de uso de caracteres na língua portuguesa. Ou seja, distribuir no layout do teclado os caracteres mais utilizados de tal

forma que não seria preciso esperar tanto para selecionar um caractere, de acordo com o tipo de varredura, seja ela tradicional ou binária.

Este último layout, sem dúvida, exigiria um tempo de aprendizagem maior para que os usuários localizassem os caracteres, uma vez que esse seria diferente do layout mais comum, o QWERTY, e também não disponibilizaria as teclas em uma ordem pré-definida, como no caso do alfabético.

O layout baseado em frequência de uso foi inicialmente pensado para a abordagem com busca binária, pois poder-se-ia distribuir os caracteres de forma a manter sempre no primeiro nó sendo varrido aqueles caracteres que são mais freqüentes. Portanto, depois de aprendido, possivelmente seria o layout que levaria a uma eficiência maior (aquele em que se escreveria com maior agilidade).

A BUSCA TRADICIONAL

A primeira abordagem concebida para desenvolver o TFlex foi a de varredura tradicional, em que o sistema de varredura é o mais simples possível: selecionando-se primeiro uma linha de teclas e, dentro desta linha, uma coluna (uma tecla).

Contudo, como se observa no caso do Teclado Virtual do Windows XP e do Simulador de Teclado do NIEE, o tamanho da tela é um fator que limita o número de linhas por colunas que podem ser mostradas pelo teclado virtual. Assim, é preciso ou desenhar teclas virtuais de tamanho reduzido o suficiente para que todas caibam na tela ou então trabalhar apenas com um conjunto de teclas em um dado momento.

Considerado estes aspectos, a varredura tradicional no TFlex foi implementada de forma a considerar 3 modos de operação: varredura em Grupos, varredura em Linhas e varredura em Colunas. O teclado varre primeiro os 4 grupos existentes: letras e símbolos, teclas de navegação (Home, End, etc.), números e teclas especiais (Alt, Ctrl, etc.). Quando o grupo desejado é encontrado, como mostrado na Figura 7, o usuário pressiona uma tecla (*Espaço*, no protótipo implementado) e a varredura começará a percorrer as linhas do grupo selecionado, como pode ser observado na Figura 8. Da mesma maneira, quando a linha desejada for alcançada, e um botão for pressionado, a varredura passará para o modo de colunas como mostra a Figura 9. Finalmente, quando o botão desejado for selecionado, um dos três eventos a seguir pode ocorrer:

- O caractere relacionado com a tecla será enviado para a caixa de texto, ou
- A janela relacionada com a tecla será exibida, ou
- A função relacionada com a tecla será executada, como por exemplo, a função que alterna as letras em maiúsculas e minúsculas.

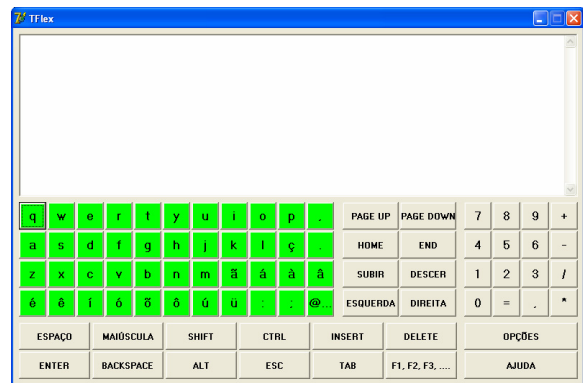


Figura 7. Varredura por grupos. Seleção do grupo de letras.

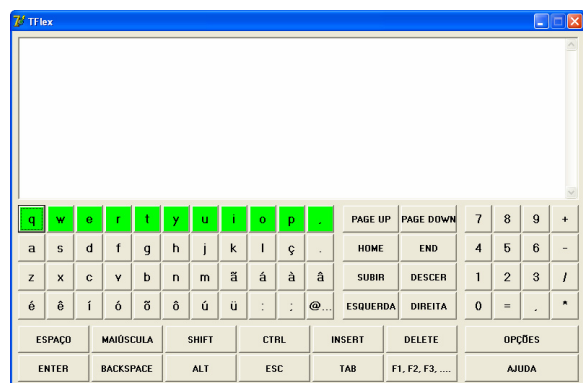


Figura 8. Varredura por linhas no grupo de letras.



Figura 9. Varredura por colunas no grupo de letras.

Se a varredura percorre duas vezes um conjunto de botões sem que nada seja pressionado, ela automaticamente retorna ao modo anterior. Por exemplo, se todas as linhas de um grupo são percorridas duas vezes, a varredura volta a percorrer todos os grupos a partir do grupo em que a linha se encontrava. Para usuários que possuem menores restrições motoras, foi implementado também um comando de volta. Ao pressionar a tecla *Esc*, a varredura volta ao nível anterior.

A BUSCA BINÁRIA

Além da varredura tradicional, foi implementada no TFlex a varredura do teclado utilizando-se busca binária.

Por que utilizar varredura binária?

Tentando buscar alternativas para a varredura tradicional, propôs-se investigar simuladores de teclado com diferentes modos de busca pelos caracteres a serem digitados pelo usuário, como uma baseada no algoritmo

de busca binária. Nesse tipo de implementação divide-se sucessivamente o teclado em duas partes, sendo que o usuário sempre escolhe uma das divisões, a qual novamente é dividida em duas partes. Dessa forma seleciona-se, no final das divisões, o caractere desejado.

Para justificar a proposição de um novo meio de acesso às letras, faz-se necessário analisar como a maioria dos simuladores de teclado atuais funciona. Em várias implementações tem-se o teclado apresentado na tela e uma linha de teclas é mostrada em contraste (em azul no caso do Teclado Virtual do Windows XP). O usuário deve então selecionar a linha que possui a tecla que se deseja pressionar. Uma vez selecionada a linha, as letras da mesma são varridas em contraste, uma após a outra, até que o usuário pressione ou uma tecla do teclado, ou um botão do mouse, ou outro tipo de acionador/comutador, para selecionar a letra desejada. A velocidade de varredura é configurável, adequando-se assim a capacidade motora do usuário. Tomemos como base para os cálculos de nossa hipótese, um tempo de 0,8 segundos em que uma linha, ou uma tecla permanece em destaque, permitindo que o usuário a selecione. Tomando como base os teclados analisados anteriormente, os quais possuem, em média, cerca de 7 linhas, e que cada linha possui em torno de 15 teclas (colunas). Suponhamos ainda que o usuário seja capaz de selecionar a linha na primeira varredura, ou seja, ele tem tempo hábil para escolher a linha desejada, sem que uma segunda varredura seja efetuada naquela mesma linha. Dessa forma, temos um tempo de seleção médio de linhas de $0,8 \text{ s} \times 3,5 \text{ linhas}$, nos resultando em um tempo total de seleção de 2,8 s. Considerando o tempo médio de seleção da coluna, ou seja, $0,8 \text{ s} \times 7,5 \text{ colunas}$, nos resulta 6 s para que a letra seja selecionada dentro de uma coluna. No total, para que o usuário digite uma letra, são necessários, no mínimo, $2,8 + 6 = 8,8 \text{ s}$.

Considerando-se a busca binária, temos ao invés de linhas e colunas, a alternância entre dois grupos de teclas. Podemos estimar o tempo gasto para a digitação de um caractere quando utilizando esse tipo de abordagem. Para realizar os cálculos, como no teclado tradicional, manteremos em 0,8 s o período de tempo em que cada grupo fica em destaque, e consideraremos o melhor caso, no qual o usuário não espera por outro período para selecionar o grupo de caracteres desejado. Dessa forma, temos que, para um teclado de 104 teclas, serão necessárias $\log_2(104) = 6,7$ interações. Tomemos como base os mesmos 0,8 s de período. Isso nos resulta em um tempo médio de acesso para um determinado caractere $6,7 \text{ int.} \times 0,8 \text{ s} = 5,36 \text{ s}$.

Assim, concluímos teoricamente que a abordagem de busca binária seria algo minimamente interessante de ser implementada. Entretanto, nota-se que para selecionar um caractere utilizando a busca binária, necessita-se de um número maior de interações. Dessa maneira, nota-se que existe uma relação de compromisso entre a rapidez de digitação e o número de interações necessárias. Assim, tomamos o teclado binário como algo interessante para pessoas capazes de realizar um número maior de interações, e tomando proveito desse maior grau de

liberdade obter-se-ia uma maior velocidade na digitação. Para pessoas com maior comprometimento motor continua certamente sendo mais recomendada a utilização do teclado utilizando a busca tradicional.

Implementação da varredura binária

Antes de iniciar a implementação discutiu-se a validade de se utilizar a expansão do grupo de caracteres selecionado, onde a cada iteração da varredura, esse se expandiria para ocupar toda a tela do teclado. Entretanto, decidiu-se que essa não seria uma abordagem interessante pelo fato das teclas ficarem modificando suas posições a cada iteração, obrigando o usuário a procurar pela tecla desejada cada vez que selecionasse um grupo. Mantendo as teclas em suas posições originais, o usuário pode manter o foco de sua visão sempre em um mesmo ponto do teclado, facilitando assim a escolha da tecla desejada e em consequência, agilizando a velocidade de digitação.

Com o intuito de reduzir o campo de busca, optou-se por realizar, como na busca tradicional, uma seleção de um grupo de caracteres de funções parecidas antes de se realizar a varredura binária.

A Figura 10 mostra a seleção do grupo que contém letras, símbolos e pontuação. Após a seleção do grupo, o mesmo é dividido em 2 subgrupos, entre os quais o foco se alterna. A Figura 11(a) mostra a divisão esquerda em destaque. Para que o usuário selecione um grupo, utiliza-se a tecla *Espaço*, como na varredura tradicional. Caso o usuário tenha errado na seleção, ele pode utilizar a tecla *Esc* para voltar para a seleção de grupos, ou ainda esperar que o sistema o faça automaticamente após 3 alternâncias de foco.

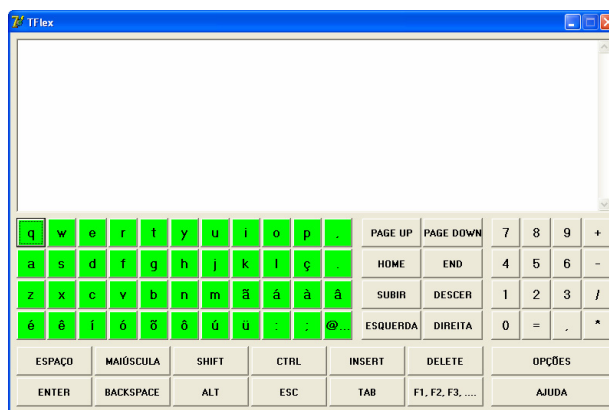


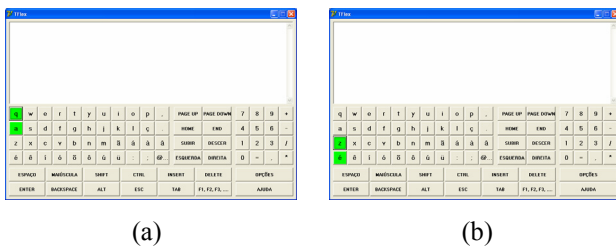
Figura 10. Seleção do grupo de letras, símbolos e pontuação.



(a)

(b)

Figura 11. (a) Seleção da seção esquerda da divisão binária na segunda iteração; (b) Terceira iteração.



(a)

(b)

Figura 12. (a) Divisões em metade Inferior e (b) Superior.



Figura 13. Seleção do caractere.

Suponhamos que a tecla a ser digitada seja a letra “a”. O usuário continua selecionando a metade esquerda, como mostrado na Figura 11(b), até chegar a uma única coluna. Nota-se que até esse momento o campo de busca foi dividido sempre verticalmente, em metades esquerda e direita. A partir desse momento, como não se adotou a abordagem de movimentação de teclas por motivos supracitados, o campo de busca será dividido horizontalmente. Ou seja, divide-se a coluna em metades superior e inferior, como mostrado e nas Figuras 12(a) e (b). O usuário continua com as iterações de divisão até chegar ao caractere desejado, como pode ser observado na Figura 13. Em seguida, a busca reinicia no grupo selecionado anteriormente, a fim de que o usuário possa iniciar a busca por um outro caractere, facilitando assim a seleção de teclas de um mesmo grupo, quando na digitação de um texto, por exemplo, caracteres. Esse comportamento tem o intuito de aproveitar o tema da digitação do usuário, facilitando a digitação futura de elementos de mesmo tema dos anteriores. Caso isso não fosse levado em conta, o usuário que acabou de digitar um número, por exemplo, teria de esperar pela varredura passar pelos grupos de caracteres especiais, letras, movimentação até chegar novamente nos números.

COMPARAÇÃO ENTRE MODELOS

Para analisar a acessibilidade dos dois modelos implementados, primeiro tentamos levantar na literatura *guidelines* que nos possibilitasse fazer uma inspeção de usabilidade, como a avaliação heurística [10], mas com especificidades relativas a características de acessibilidade, que eram as que nos interessava neste primeiro teste.

Atualmente, existem diversos guias de acessibilidade, tanto para software quanto para Internet, havendo uma preponderância maior de guias para a Internet. Dentre estes guias, destacam-se os da WAI/W3C para desenvolvimento de conteúdo para a Web acessível [1] e ferramentas de autoria para Web [13], além das recomendações de acessibilidade da Microsoft [8] para seus sistemas operacionais.

Desenvolver um software acessível a partir de guias como o da WAI/W3C para ferramentas de autoria para a Web pode ser considerada uma tarefa difícil, uma vez que tais guias podem conter informações abstratas, em alto nível, dizendo apenas o que deve ser feito, e não o como, e muitos não levam em conta o contexto de uso do software em questão (se na educação, em um ambiente de trabalho, etc.). Já recomendações como as da Microsoft dependem de ferramentas de desenvolvimento exclusivas da empresa, e são voltadas principalmente para a implementação de software usável com determinadas tecnologias assistivas (leitores de tela, por exemplo). Ainda, estes guias possuem mais dois pontos negativos. Primeiro, nenhum deles considera testes com usuários para se indicar o nível de acessibilidade. Segundo, nenhum guia é aceito universalmente.

Desta forma, avaliar a acessibilidade de um software apenas verificando sua conformidade com tais guias não seria apropriado. E, infelizmente, ainda não existem metodologias formais amplamente difundidas voltadas exclusivamente para a avaliação de acessibilidade.

Optamos, portanto, em iniciar a avaliação dos dois tipos de varredura implementados a partir de testes de usabilidade nos quais a tarefa consistia na digitação de um texto simples: “Parreira levará a campo os mesmos jogadores que golearam a Hungria por 4 a 1.”. Para isso nos baseamos no layout QWERTY, de modo a tomar proveito da experiência que os usuários possuíam nesse layout.

Para a realização dos testes utilizamos 5 usuários de diversas áreas do conhecimento, sendo os mesmos não portadores de qualquer deficiência. Todo o grupo já possuía certa experiência com o uso de computadores, e nenhum havia utilizado simuladores de teclados, bem como não foi realizado qualquer treinamento com os usuários antes da aplicação dos testes. Mantivemos o intervalo de chaveamento de grupos em 0,8 s, de modo a adequar a uma ampla gama de usuários.

O usuário 2 somente testou o TFlex com busca binária, enquanto que o usuário 3 somente utilizou a busca tradicional. O intuito desse tipo de teste foi a de se eliminar a influência do aprendizado nos resultados do teste, uma vez que o usuário poderia aproveitar a experiência adquirida da utilização de um teclado bem como relembrar a disposição das teclas para agilizar a digitação.

Nos outros casos, os usuários testaram os dois modos de varredura, sendo que para os usuários 1 e 4, iniciou-se com a busca tradicional sendo em seguida utilizada a binária. No caso do usuário 5 aplicou-se o teste na ordem inversa. Há uma ressalva quanto ao usuário 5, já que ele faz parte da equipe de desenvolvimento do TFlex, entretanto o resultado desse teste foi interessante pois pudemos analisar os efeitos que a experiência traz na digitação. O tempo e o número de toques que cada usuário utilizou na digitação do texto nos testes estão organizados nas Tabelas 1 e 2.

	Tipo de Busca (Minutos)	
	Tradicional	Binária
Usuário 1	14	23
Usuário 2	-	18
Usuário 3	15	-
Usuário 4	11	17
Usuário 5*	10	12

Tabela 1. Tempo (em minutos) para digitação de um texto elementar.

	Tipo de Busca (Número de toques)	
	Tradicional	Binária
Usuário 1	269	597
Usuário 2	-	824
Usuário 3	Não computado	-
Usuário 4	242	656
Usuário 5*	250	555

Tabela 2. Toques necessários na digitação de um texto elementar.

Nota-se que o aprendizado, como esperado, tem uma influência direta na eficiência de digitação. Observa-se que o usuário 1 utilizou menos toques que o usuário 2 devido a sua experiência prévia com a busca tradicional. Nota-se que o usuário 5 obteve o menor número de toques na busca binária devido a sua ampla experiência com esse tipo de varredura, assim como o tempo utilizado para a digitação foi o menor de todos. Por outro lado, o usuário 3, por sua falta de experiência com a busca tradicional utilizou o maior tempo para digitar o texto.

Um aspecto a ser observado é que a experiência influencia significativamente na eficiência de uso, sendo assim, é de se esperar que o usuário necessite de um tempo de aprendizado. Futuros testes serão conduzidos no sentido de se avaliar esta curva de aprendizagem com relação aos dois tipos de varredura. Principalmente, com relação ao da varredura binária onde a princípio a eficiência pretendida somente será alcançada quando o usuário entender este algoritmo de busca. E isso de alguma forma deverá ser possibilitado por características da interface que salientem os aspectos envolvidos no processo de busca, principalmente com relação a subdivisão e alternância entre grupos.

Através desses testes, utilizamos as opiniões dos usuários para aprimorar o TFlex. Uma das características já alteradas no protótipo foi com relação ao modo de varredura binária. Neste modo de varredura ao pressionar a tecla *Esc* o TFlex retornava a varredura para a seleção de grupos, diferentemente da varredura tradicional, onde a varredura retorna para o modo anterior. Tomemos o

exemplo da varredura das colunas de uma linha para procurar um determinado caractere, mas o caractere se encontra na linha anterior. Ao se pressionar *Esc* a busca retoma a varredura por linhas, fazendo com que o usuário possa se recuperar rapidamente do erro. Esse comportamento ocasiona na busca binária um número superior de iterações no caso do usuário cometer um erro de seleção na varredura.

Outros pontos importantes e que estão sendo analisados dizem respeito à divisão dos grupos e o posicionamento de teclas com elevado uso, como a tecla de ESPAÇO, por exemplo, que devido a sua alta frequência de uso possivelmente deverá ser posicionada junto ao grupo de caracteres e prevê-se a necessidade de incluí-la no teclado com alguma redundância.

Um dos resultados mais efetivos desses testes iniciais foi que a consistência espacial deve ser garantida como uma das características básicas de usabilidade. Por exemplo, o erro mais comum dos usuários no teste foi com relação às teclas acentuadas que estão divididas em mais de uma linha. E sempre a seleção era feita na linha que iniciava um caractere, dada a característica da varredura tradicional ser por linhas. Assim o usuário iniciava a busca de um caractere em uma determinada linha, ainda sem levar em consideração o acento desejado, e quando direcionava sua atenção para o acento do caractere, percebia que o mesmo se encontrava em outra linha.

E finalmente, pode-se concluir que aumentando-se a amplitude dos testes a partir de um melhor refinamento do protótipo, teremos coletado uma série de resultados que esperamos poder compor na forma de um primeiro conjunto de *guidelines* a serem consideradas no design de produtos de software com funcionalidades e objetivos análogos aos do TFlex.

CONCLUSÕES

As tecnologias assistivas são importantes dispositivos capazes de ajudar no acesso aos recursos computacionais pelas pessoas portadoras de deficiência. Uma destas tecnologias é conhecida por simulador de teclado, um teclado virtual mostrado na tela do computador que possui um sistema que varre as teclas.

Os modelos atuais de simuladores de teclado possuem certas restrições que dificultam a sua adoção por usuários portadores de deficiência motora ou física. Por estas razões, o TFlex foi proposto e está em processo de desenvolvimento, podendo ser obtido gratuitamente em [12].

Apesar de testes com os primeiros protótipos do TFlex terem sido feitos apenas com usuários não portadores de deficiência, alguns aspectos foram notados, como a necessidade de se rever a disposição de teclas específicas (Enter, Backspace, Espaço, etc.) e de se configurar o tempo da varredura de acordo com a habilidade do usuário. Utilizamos voluntários não portadores de necessidades especiais em um primeiro momento com o objetivo de descobrir os principais problemas de usabilidade e refinar o protótipo.

Há de se considerar, também, que os usuários que realizaram os testes com o TFlex estavam acostumados com um modelo direto de entrada de dados no computador, através do teclado físico tradicional. Provavelmente, isto influenciou nos testes e a sensação generalizada de pouca eficiência em escrever com um simulador de teclado. Salienta-se, mais uma vez, que o objetivo principal do TFlex (e de outros simuladores de teclado) é a de se permitir a escrita por aqueles que não podem usar nem o teclado nem o mouse. Ainda assim, o TFlex possui abordagens que tentam melhorar a eficiência, como a proposta de varredura baseada na busca binária e os layouts alfabético e por frequência de uso. Além disso, concluímos que algumas modificações simples podem ser realizadas na busca binária de modo a melhorar a forma de recuperação em caso de erro de digitação.

Planeja-se, com o desenvolvimento de um protótipo mais refinado, testar o TFlex com usuários portadores de deficiência a fim de se comparar de forma mais realista tanto os modos de varredura (tradicional e binária) quanto os layouts de teclado propostos (QWERTY, alfabético e por frequência de uso).

Nota-se ainda que os refinamentos posteriores do TFlex devem levar em consideração a relação de compromisso entre manter um layout de teclado que seja eficiente, mas ao mesmo tempo, mantenha o aspecto de um teclado físico. Além disso, deve-se levar em consideração o tamanho das teclas do TFlex de forma a facilitar a sua leitura, bem como manter um tamanho de janela suficientemente pequeno para não atrapalhar a visualização de outras janelas de programas que por ventura sejam utilizadas junto com o TFlex.

Por fim, espera-se que este projeto resulte em um software de tecnologia assistiva que sirva como uma alternativa às pessoas que não podem utilizar os periféricos tradicionais de entrada de dados no computador.

AGRADECIMENTOS

Primeiramente gostaríamos de agradecer a professora Heloísa V. da Rocha do Instituto de Computação da UNICAMP pelo suporte e apoio ao desenvolvimento desse projeto, bem como no auxílio e dedicação aos testes realizados. Não podemos deixar de agradecer a todos os

voluntários que se prontificaram a testar o TFlex, como também pelas preciosas críticas e determinação dos pontos problemáticos do projeto.

REFERÊNCIAS

- 1.Chisholm, W.; Vanderheiden, G. Jacobs, I. Web Content Accessibility Guidelines 1.0, maio/1999. <http://www.w3.org/TR/WCAG10/>.
- 2.Hartman, J. How the Typewriter Got Its Keys. <http://www.kith.org/logos/words/upper/Q.html>
- 3.Hogetop, L; Santarosa, L. M. C. Tecnologias Assistivas/Adaptativas: viabilizando a acessibilidade ao potencial individual. set/2001. <http://www.nied.unicamp.br/~proinesp>.
- 4.IBGE. Censo Demográfico 2000 - Tabulação Avançada. População residente, por situação do domicílio e sexo, segundo o tipo de deficiência, Brasília/DF, 2003. <http://www.ibge.gov.br/estatistica/populacao/censo2000>
- 5.Inclusive Technology. Access Devices. <http://www.inclusive.co.uk/catalog/hardware.shtml>.
- 6.Lakefolks. Click-N-Type Virtual Keyboard. <http://www.lakefolks.org/cnt/>.
- 7.Mackay, D. The Dasher Project. <http://www.inference.phy.cam.ac.uk/dasher/>.
- 8.Microsoft. MSDN Library - User Interface Design and Development – Accessibility. <http://msdn.microsoft.com/at>.
- 9.NIEE. Simulador de Teclado. <http://www.niee.ufrgs.br/software/soft1.html>.
- 10.Nielsen, J. Usability Engineering. Academic press Limited, 1993.
- 11.SACI. O Projeto Teclado Amigo. <http://www.saci.org.br/pub/kitsaci2/projeto.html>.
- 12.TFlex. Projeto TFlex. <http://www.nied.unicamp.br/~tflex/>
- 13.Treviranus, J; McCathieNevile, C; Jacobs, I; Richards, J. Authoring Tool Accessibility Guidelines 1.0 – W3C Recommendation, fev/2000. <http://www.w3.org/TR/ATAG10/>.
- 14.WiViK. What's WiViK?. <http://www.wivik.com/Description/description.html>.

Um método de avaliação para interfaces baseadas em mapas

Maria Lúcia Arraes Seixas

Clarisse Sieckenius de Souza

Semiotic Engineering Research Group
Departamento de Informática, PUC-Rio – Brazil
<http://peirce.inf.puc-rio.br/>
{malu, clarisse}@inf.puc-rio.br

RESUMO

A integração da informação geográfica e dos mapas digitais nos Sistemas de Informação Geográfica (SIGs) produz soluções para vários grupos de usuários. Porém, vem crescendo o número de usuários não especializados que utilizam SIGs na *Web* ou em *desktop* para localizar endereços, planejar rotas, planejar viagens, encontrar lugares, e assim por diante. Estas aplicações menos complexas e destinadas a qualquer pessoa precisam ser avaliadas, pois da qualidade da interação depende seu sucesso comercial. Apesar desta necessidade, praticamente não encontramos trabalhos avaliando estas aplicações. Neste trabalho propomos um método de Inspeção Semiótica para Interfaces baseadas em Mapas (ISIM) capaz de analisar o efeito de representações usadas nestas interfaces quanto a capacidade de comunicar a informação geográfica. O ISIM emprega técnicas analíticas para definir o contexto da aplicação e o perfil do usuário, bem como para interpretar os resultados obtidos. Emprega também uma técnica empírica, de testes e entrevistas com usuários, para comparar parte dos dados obtidos com os das técnicas analíticas. O ISIM, inicialmente proposto para avaliação de aplicações cuja parte da interface é baseada em mapa, procura identificar uma classe de problemas básicos decorrentes de interação, porém significativos do ponto de vista de utilização e interpretação de mapas de um modo geral.

ABSTRACT

The integration of digital maps and geographical information in Geographical Information Systems (GIS) produces solutions for several groups of users. However, it is increasing the number of non expert users that use GIS in the Web or in his desktop to find addresses, to plan routes, to plan trips, to find places, and so on. These less complex applications that are destined to anybody also need to be evaluated, because its commercial success depends of the quality of the interaction. In spite of this need, practically there is no works evaluating these applications. In this work we propose a semiotic inspection method (whose acronym in Portuguese is ISIM) that analyzes the effect of representations in communicate the geographic information in map based

VI Simpósio sobre Fatores Humanos em Sistemas Computacionais — Mediando e Transformando o Cotidiano. Hotel Bourbon, Curitiba, 17 a 20 de outubro de 2004. UFPR, CEIHC—SBC.

interfaces. ISIM uses analytical techniques to define the context of the application and the user's profile, as well as to interpret the results. ISIM also uses an empirical technique, of tests and interviews with users. ISIM, initially proposed for evaluation of applications whose part of the interface is based on map, identifies a class of basic problems.

Palavras Chaves

Avaliação de Interfaces, Engenharia Semiótica, Interfaces baseadas em Mapas.

INTRODUÇÃO

Sistemas de Informação Geográfica (SIGs) são sistemas de informação construídos com a finalidade de armazenar, analisar e manipular dados geográficos. Os dados geográficos representam objetos e fenômenos em que a localização geográfica é uma característica inerente e indispensável para tratá-los [2]. A tecnologia de SIG integra operações de banco de dados, como consultas, com a apresentação e análise de dados através de mapas. Apesar de os avanços nas pesquisas de sistemas de informação geográfica estarem sendo rapidamente incorporados em produtos de SIG, “sabe-se pouco sobre como os mapas funcionam nestes contextos ou sobre as ferramentas apropriadas a se oferecer” [13].

A integração da informação geográfica e dos mapas digitais nos SIGs produz soluções para vários grupos de usuários. Entre eles estão os não especialistas: pessoas com formações em diversas áreas do saber, com diferentes conhecimentos computacionais e pertencentes a diferentes culturas. A utilização de mapas interativos para estes usuários, principalmente para aqueles acostumados a usar mapas estáticos ou em papel, é uma tarefa que exige alguma habilidade, pois além de terem de interpretar as informações cartográficas eles precisam interagir com a interface. Um dos maiores obstáculos para os usuários não especialistas é ter de interagir com uma interface onde estão embutidas linguagens, conceitos e visões do mundo que refletem a arquitetura do sistema ao invés de dar suporte às suas atividades [27]. Para conhecer mais sobre este tipo de interação é importante que se possam usar métodos de avaliação de interface e interação apropriados.

As interfaces baseadas em mapas (IMs) são importantes já que a maioria dos SIGs as utiliza para apresentação de dados e interação com o sistema. Os SIGs também usam outros tipos de interface (e.g. gráficos, tabelas, menus, diálogos), mas para sua avaliação é possível aplicar

métodos genéricos já conhecidos. Neste trabalho propomos um método que se aplica especificamente às interações que envolvem diretamente a visualização (ou apresentação) e a interpretação de mapas.

Nas IMs o usuário pode interagir modificando a apresentação do mapa, variando os elementos que serão visualizados, a forma como serão apresentados e até mesmo o contexto de uso. Assim, o objetivo clássico da cartografia — produzir um “mapa ótimo” — fica bastante relativizado em um ambiente onde nem a pessoa que faz (ou determina as especificações gerais para) os mapas visualizados, nem as pessoas que os utilizam sabem o que querem ou precisam ver nos mapas até o momento de vê-los [13].

O método que propomos adota uma perspectiva semiótica para analisar o efeito da apresentação (ou visualização¹) na comunicação da informação geográfica em interfaces baseadas em mapas. Para tanto usamos a teoria da Engenharia Semiótica [5,6,8] e, mais especificamente, o conceito de Contínuo Semiótico, proposto no âmbito da avaliação de aplicações extensíveis [7]. Partimos da hipótese que rupturas ou falhas na semiose do usuário, decorrentes de apresentações não relacionadas com tarefas e contextos, correspondem a rupturas no contínuo semiótico entre as linguagens de representação ou descrição da interface. O contínuo semiótico é uma propriedade formal de linguagens com que o usuário tem contato, e contribui para orientar o foco de atenção e interpretação do usuário.

Os SIGs a que o método proposto aqui se aplica são os que permitem que o usuário tenha acesso a vários mapas co-referentes, em várias escalas, durante a realização de suas tarefas. As tarefas neste tipo de sistema podem ser variadas, uma vez que o universo de aplicações e de usuários pode ser muito diversificado. Para ilustrar o método, serão utilizadas inicialmente tarefas para encontrar caminhos e lugares a partir de uma origem e um destino. Neste tipo de aplicação grande parte da interação depende de mapas e há um problema importante na área de IHC (Interação Humano-Computador) no que diz respeito às formas e aos efeitos das operações de *zoom*.

Até o momento, não encontramos métodos de avaliação específicos para este tipo de interação. Muitas das pesquisas sobre avaliações de interfaces baseadas em mapas encontrados atualmente utilizam métodos e técnicas tradicionais de IHC, pautados em teorias cognitivas ou em conhecimentos heurísticos, que são adequados somente para a avaliação de alguns aspectos do sistema. Características específicas de interação com mapas, como a percepção de orientação, de navegação, a identificação de objetos e a interpretação da simbologia são comumente esquecidas ou tratadas apenas superficial e assistematicamente pelas técnicas tradicionais.

O objetivo de nosso método de inspeção semiótica de interfaces baseadas em mapas (ISIM) é verificar como a

¹ Neste trabalho usaremos “apresentação” e “visualização” como termos intercambiáveis.

apresentação através de elementos do mapa e da mudança da escala pode ser interpretada e, por conseguinte, como pode afetar a tarefa do usuário em relação à percepção de orientação e de navegação, identificação de objetos e interpretação da simbologia. Por exemplo, para que a mudança nos mapas não atrapalhe a conclusão da tarefa do usuário, o foco da tarefa precisa ser mantido durante a interação. Portanto, neste caso seria importante saber: Como manter o foco da tarefa durante a interação com mapas? Quais elementos precisam ser mantidos visíveis para que o usuário não se perca? Como modificar a apresentação dos objetos de modo que o usuário consiga manter a referência do objeto?

Considerando um ambiente onde existem ferramentas como *zoom* e *pan* que permitem que os usuários modifiquem o que estão vendo, é importante também verificar como estas ferramentas contribuem para a realização de uma tarefa, já que durante a interação com mapas o usuário pode fazer várias mudanças de escala (*zoom*) e deslocamentos (*pan*) para visualizar uma região do mapa. Estas mudanças podem facilitar a realização das tarefas, porém, muitas vezes, fazem com que o usuário fique perdido durante a interação.

Nas próximas seções deste trabalho apresentaremos: uma breve descrição dos trabalhos relacionados a nosso tema de pesquisa; os conceitos de Engenharia Semiótica sobre os quais se fundamenta nossa proposta; a descrição ilustrada do método; e finalmente uma conclusão apontando os trabalhos futuros que planejamos realizar.

TRABALHOS RELACIONADOS

Há no momento grande motivação para as pesquisas sobre SIGs relacionadas com o desenvolvimento [22] e a avaliação de aplicativos [3,12,18,19,29]. Várias delas estão voltadas para a visualização² de dados espaciais ou geovisualização³ [10,14,21,24,26,28]. Porém, vem crescendo o número de usuários não especializados que utilizam SIGs na *Web* ou em *desktops* para localizar endereços, planejar rotas, planejar viagens, encontrar lugares, e assim por diante. Estas aplicações menos complexas e destinadas a qualquer pessoa precisam ser avaliadas, pois da qualidade da interação depende seu sucesso comercial. Entretanto, praticamente não encontramos trabalhos avaliando estas aplicações, que ao contrário do que acontece em sistemas gerais para

² Segundo [26], “visualização é uma técnica de análise de dados que conta com a habilidade humana para reconhecer padrões em ambientes computacionais flexíveis que apóiam a exploração interativa de dados em tela. Visualização é particularmente útil e apropriada quando se conhece pouco sobre o conjunto de dados para aprender sobre suas características, fazer descobertas e formar hipóteses sobre relações entre os seus atributos.”

³ Geovisualização integra visualização, cartografia, análise de imagens, visualização de informações, análise exploratória de dados e SIG para oferecer teoria, métodos e ferramentas para exploração visual, análise, síntese e apresentação de dados espaciais [14].

geovisualização devem atender a objetivos e atividades bastante contextualizados e culturalmente determinados. Os poucos trabalhos que há não usam métodos específicos que possam identificar problemas bastante comuns, e potencialmente graves, na apresentação dos mapas e interação sobre eles. Apesar de alguns autores [14,24] reconhecerem que o uso de princípios tradicionais de usabilidade não considera características específicas destes ambientes, há uma grande escassez de alternativas que efetivamente avancem sobre as questões fundamentais da interação sobre mapas.

A escolha da engenharia de usabilidade é problemática porque, embora ela forneça um conjunto de geral de diretrizes para examinar a eficiência de ambientes computacionais, no caso da maioria das aplicações de SIGs investigadas pelos pesquisadores até aqui, não se sabe "a priori" a que conjunto específico de tarefas ou atividades dos usuários as visualizações irão servir. Isto vai de encontro a um dos pressupostos fundamentais da engenharia de usabilidade, pois ela requer que se tenha em mãos modelos de tais atividades e tarefas. Por basear-se em modelos propostos por teorias cognitivas (como a da Engenharia Cognitiva [16]) onde o comportamento racional, baseado em metas e planos, é a base para a concepção do que é IHC, a engenharia de usabilidade tem pouca estabilidade na avaliação de interações genéricas sobre visualizações e, menos ainda, na avaliação da qualidade das próprias visualizações. Este é, por excelência, o domínio da semiótica.

ENGENHARIA SEMIÓTICA

A Engenharia Semiótica atribui ao designer e ao usuário o mesmo papel durante o processo iterativo: o de interlocutores um do outro. As interfaces dos sistemas são vistas como artefatos de metacomunicação. Esta perspectiva difere das abordagens cognitivas, onde o software é avaliado do ponto de vista de como os usuários o entendem e usam. Na abordagem semiótica o foco está na comunicação, que se desdobra em dois níveis inter-relacionados. Primeiramente a comunicação unilateral e integral, do designer para o usuário, sobre o que ele (designer) fez e por que, baseado em que premissas sobre o usuário, seu contexto, seus gostos, preferências, capacidades e valores. Esta comunicação de primeiro nível se realiza e desdobra através de uma comunicação de segundo nível, entre usuário e sistema (a que a maioria das teorias concorrentes de IHC toma por constituir a única comunicação que se passa). O sistema representa o designer, na hora da interação com o usuário, e faz com que a mensagem unilateral e integral "passe" para o usuário através de conversas e interações dialogadas, típicas de todas as situações em que pessoas usam computadores para uma variedade de fins.

Para se avaliar a interação em um software com base na Engenharia Semiótica é necessário avaliar esta conversa entre o preposto do designer (*i.e.* o sistema) e o usuário. O sucesso com que um usuário será capaz de interagir com uma aplicação e realizar suas tarefas está fortemente relacionado à qualidade da comunicação (de fato à metacomunicação, ou seja: da comunicação integral através da comunicação desdobrada em conversas) das intenções do

designer e dos princípios de interação que guiaram o design da aplicação [20]. Esta é uma das diferenças fundamentais entre o método que estamos propondo e os métodos de avaliação comumente utilizados que não enxergam a interação dessa forma. A outra é a de aprofundar a qualidade dos próprios mapas, em especial daqueles que são co-referentes (*i.e.* que se referem a uma mesma geo-região), para o quê nos valem do princípio do Contínuo Semiótico.

Contínuo Semiótico em Mapas

O princípio do contínuo semiótico, proposto em conjunto com o de abstração interpretativa em [7] é uma propriedade que se aplica à relação entre duas linguagens computacionais. A seguir podemos ver como foram propostos originalmente.

Princípio da Abstração Interpretativa

Uma Linguagem de Interface com o Usuário (UIL) é uma abstração interpretativa de uma ou mais linguagens computacionais subjacentes (que são necessárias para determinar a semântica da interação efetiva) se os usuários conseguem entender completamente todos os signos da UIL devido:

- aos padrões e combinações signícas que eles encontram enquanto interagem com a aplicação;
- às explicações disponíveis sobre a UIL (conforme estão disponibilizadas no módulo de ajuda *online* e tutoriais, *screen tips* e documentação da aplicação em geral); e
- à sua própria experiência com computadores (trazendo à tona o conhecimento associado usado nas metáforas da interface, conhecimento do domínio da aplicação e puro senso comum).

O princípio da abstração interpretativa fala sobre a interpretação dos signos de uma linguagem computacional. Dizemos que esta linguagem computacional abstrai, para ser interpretada, todas as outras camadas de linguagens existentes sob ela. Logo, uma linguagem computacional é uma Abstração Interpretativa de uma ou mais linguagens se um intérprete humano for capaz de interagir com a primeira, sem conhecimento algum das especificações expressas nas linguagens subjacentes.

Princípio do Contínuo Semiótico

Duas linguagens são semioticamente contínuas se o seu acoplamento pragmático, ao se traduzir uma linguagem na outra, for sempre preservado. No caso de aplicações extensíveis, a parte extensível da UIL (UILx) é semioticamente contínua com a Linguagem de Programação para o Usuário Final (EUPL) se:

- os usuários conseguem extrair sentido dos signos da UILx e interagir com a UILx sem qualquer conhecimento da EUPL ou de sua existência;
- os usuários são capazes de compreender os signos da EUPL e gerar textos na EUPL sem qualquer conhecimento de outras linguagens de mais baixo nível, tais como linguagens de programação, APIs de sistemas operacionais, etc.;

- existe um construtor para um texto pragmaticamente válido na EUPL; e
- qualquer usuário que conheça a UILx e a EUPL sempre pode traduzir uma instância arbitrária de um texto da EUPL para uma combinação válida, realizada ou potencial, de signos da UILx.

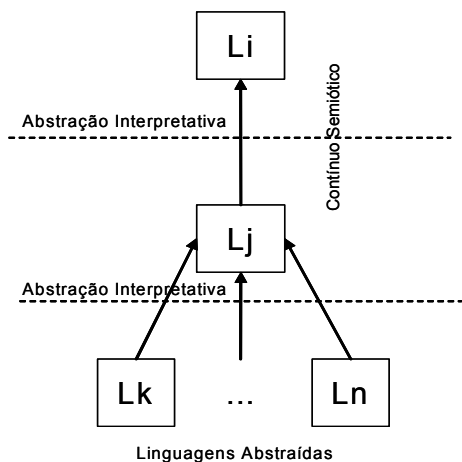


Figura 1. Representação visual dos princípios de Abstração Interpretativa e Contínuo Semiótico

O efeito do contínuo semiótico (Figura 1) é identificar linguagens de programação cujos programas o usuário sempre consegue traduzir em estados, eventos e interações de interface. Se a linguagem de programação deixa o usuário fazer programas, bem-formados, que não podem ser traduzidos em fenômenos de interação, ela não é semioticamente contínua com a linguagem de interface.

Uma característica importante dos mapas interativos é que a sua apresentação pode ser modificada durante a interação. Apesar de teoricamente estarmos mexendo no mesmo objeto, neste caso o mapa, o fato de a sua apresentação estar variando é análogo a estarmos produzindo (ou gerando) *n* mapas, sendo o primeiro mapa apresentado no estado inicial do sistema e o *n*-ésimo mapa o último produzido ao final das interações. Especialmente no caso de *zooming*, se pudéssemos ver os *n* mapas produzidos por variações de escala, teríamos uma seqüência de imagens como no exemplo na Figura 2. Nela, a seqüência de mapas apresentados correspondem a sucessivas abstrações onde o primeiro mapa, contendo mais detalhes, é seguido por outro, uma abstração do primeiro, e assim sucessivamente.



Figura 2. Contínuo Semiótico nas variações de escalas.

Para interpretarmos esta seqüência de mapas sob um ponto de vista semiótico precisamos inicialmente entender o que é um signo e semiose. Um signo é uma relação triádica formada pelo *representamen*, que é a representação (perceptível para um intérprete) do objeto que a determina; pelo objeto que é aquilo que existe física

ou conceitualmente e possibilita que haja uma representação para ele; e pelo interpretante que é aquilo que a presença de uma representação faz surgir na mente do intérprete [17]. O processo de apreensão de um signo é chamado de semiose. Esta interpretação mental é pragmática e, portanto, sujeita a contingências da situação de interpretação. O interpretante é ele mesmo um signo, que pode ser determinado por fatores contingentes e disparar uma cadeia indefinidamente longa de interpretações. A Figura 3 exemplifica uma cadeia semiótica produzida durante a interpretação de uma seqüência de mapas.

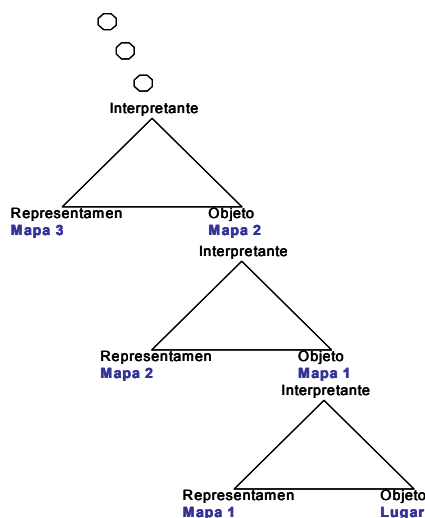


Figura 3. Semiose em seqüências de mapas.

O efeito do contínuo semiótico em seqüências de visualizações de mapas deve ser o de fazer com que a interpretação de um Mapa 1, ou seja, o signo produzido como interpretante do Mapa 1 esteja representado de alguma forma em um Mapa 2. Em outras palavras, o Mapa 2 deve poder ser reconhecido como um *representamen* da interpretação do Mapa 1 em uma cadeia semiótica contínua. Por exemplo, a princípio, uma seqüência de mapas em escalas sucessivas, como na Figura 2, é um exemplo de contínuo semiótico, pois a transição entre as visualizações mantém referências abstraídas da visualização anterior.

No caso de mapas sucessivos onde se faz não uma variação de escala, mas uma variação temática (como, por exemplo, a alternância de um mapa geopolítico para um mapa geológico, co-referente), algo análogo se passa. Porém, este exemplo revela uma reformulação da noção de Contínuo Semiótico quando passamos de linguagens de extensão e interface para linguagens alternativas de visualização ou apresentação. Tomando o exemplo de dois mapas co-referentes, um geopolítico e um geológico, é interessante notar que tanto o geopolítico pode ser uma abstração interpretativa do geológico, quanto *vice-versa*. Ou seja, neste caso, a determinação semântica específica de qual é a base não é tão importante quanto no contexto em que o princípio foi originalmente enunciado. Assim, propomos que, para efeitos do método que iremos apresentar – o ISIM – o princípio do Contínuo Semiótico seja reformulado para:

As representações de dois mapas, M1 e M2, são semioticamente contínuas se:

- os usuários conseguem extrair sentido dos signos de M1 e interagir com M1 sem qualquer conhecimento de M2 ou de sua existência;
- os usuários são capazes de compreender os signos de M2 e interagir com M2 sem qualquer conhecimento de M1 ou de sua existência;
- qualquer usuário que conheça M1 e M2 sempre pode *mapear* uma localização de M1 em M2 e *vice-versa*.

O importante desta reformulação do princípio no contexto de IM é que a noção original de *tradução pragmaticamente acoplada de um texto em EUPL para signos reais ou potenciais da UILx* é substituída pela noção de *mapeamento de uma localização de M1 em M2 e vice-versa*. Esta reformulação tem, como vimos, a vantagem de relaxar a restrição de que um mapa tem de ser a origem semântica absoluta de todos os outros em uma cadeia de mapas produzidos. Assim, pouco importa por qual *visão* ou *tipo de mapa* o usuário tenha iniciado a sua interação, a essência do contínuo semiótico, que é a de um fio condutor na interpretação de signos co-referenciados, se preserva.

Quando uma interação resulta em visualizações que não são semioticamente contínuas, há uma ruptura e a pessoa não consegue interpretar corretamente a nova apresentação. Por exemplo, na Figura 4 o usuário tem uma apresentação A1 de uma área da cidade de São Francisco, Califórnia. Em uma situação onde o usuário vai fazer uma visita à igreja Saints Peter and Paul Catholic Church, visível no mapa, e depois quer encontrar lugares próximos para serem visitados. De acordo com o módulo de ajuda, o usuário deve encontrar e selecionar o ponto mapa, neste caso a igreja, e depois clicar em Find Nearby Places no menu Tools ou na barra de ferramentas padrão, selecionar as categorias que deseja ver (restaurantes e lugares), definir o raio da busca (no exemplo é de 1,5 km) e a ordem em que os lugares aparecerão (distância ou alfabética). A ação leva a uma nova apresentação A2 onde justamente o ponto de referência da atividade e contexto do usuário desaparece, levando o usuário a questionar: Onde estou? Ou Onde está a Igreja?

CARACTERIZAÇÃO DO PROBLEMA - UM CASO ILUSTRATIVO

Este caso foi escolhido após uma série de estudos e observações de casos reais de uso feitos com usuários onde os resultados mostraram evidências sobre as dificuldades que os eles têm de efetuar mudanças de escalas (*zoom in*, *zoom out*) de uma maneira adequada e produtiva. Vários autores (por exemplo, [11,15,25]) discutiram as dificuldades de se projetar (e subsequentemente de se usar) as facilidades de “*zoom*” nas interfaces de SIGs.

O *zoom* tem o efeito de mudar a escala do mapa. Ele pode ser feito livremente pelo usuário quando este seleciona uma região do mapa e pede para mostrá-la numa escala

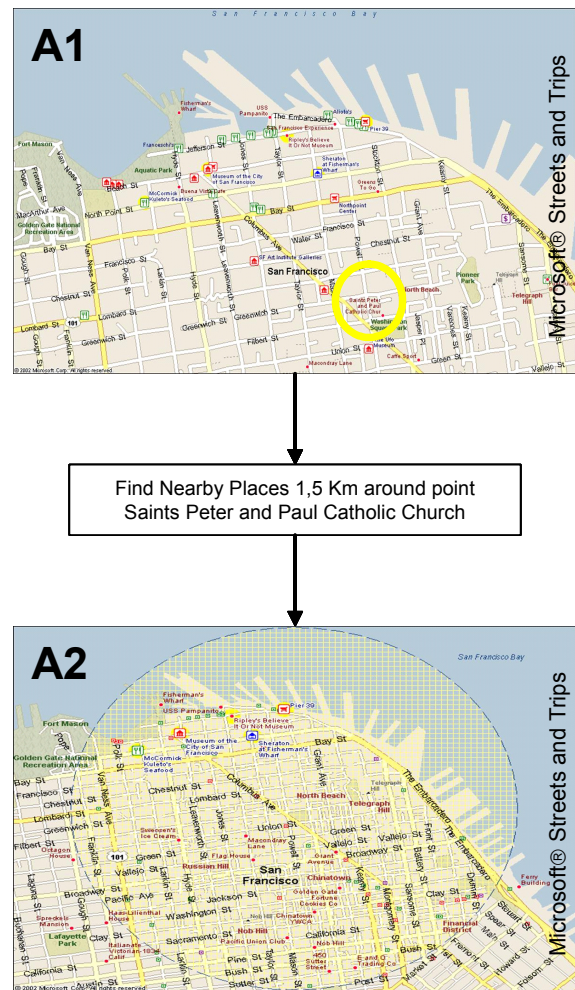


Figura 4. Quebra do Contínuo Semiótico.

menor, por exemplo, através de uma ferramenta própria. Ele também pode ser feito pelo sistema quando o usuário realiza uma ação que pode afetar a apresentação atual.

Um exemplo de *zoom* realizado pelo sistema é o que acontece automaticamente quando o usuário faz uma busca de um objeto no mapa e o sistema calcula que uma boa visualização e identificação do objeto encontrado não são possíveis na escala corrente que o usuário está utilizando. Neste caso, há uma mudança “automática” de escala, para cuja compreensão o usuário precisa, claro, dar-se conta de que ela ocorreu e reinterpretar o que está vendo face ao contexto anterior e a suas intenções na interação. Um outro exemplo é o da Figura 5 onde podemos ver um caso em que um usuário modifica a escala do mapa através de uma barra deslizante e o mapa resultante não contém a região anteriormente marcada com uma legenda, uma referência colocada pelo usuário. Esta modificação na escala, apesar de ter sido solicitada pelo usuário foi realizada pelo software utilizando um critério que não está bem claro. O que guiou a mudança de escala? Qual foi a referência adotada? Por que não foi o objeto marcado pelo usuário? Por que não foi um ponto no centro do mapa anterior ao *zoom*?

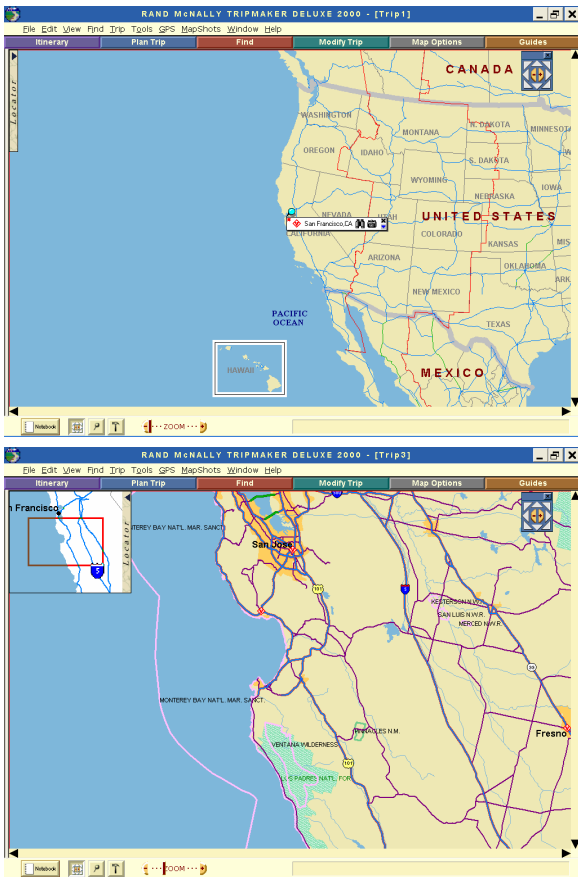


Figura 5. Mudança de escala

O zoom efetuado pelo sistema faz parte de uma decisão de design e seu comportamento pode ser previsto. Ele precisa levar em conta o tipo de tarefa que está sendo executada, o contexto atual, as preferências do usuário e acima de tudo não fazer com que o usuário se perca ao mudar a apresentação do mapa. Uma mudança de escala mais adequada pode ser vista na Figura 6, onde a referência marcada pelo usuário continua visível, deixando claro que região do mapa está sendo mostrada.

A perda da referência anterior não é a única dificuldade que o usuário pode encontrar com a mudança de escala. Por exemplo, mudanças na simbologia podem não deixar claro o que estão referenciando. Como pode ser visto na Figura 7 a mudança de escala alterou a apresentação dos objetos no mapa, uma necessidade devido à redução do espaço para apresentá-los, porém haviam objetos representados em vermelho, os hotéis, e em verde, os parques, e na nova escala todos ficaram vermelhos. O que será que houve com os parques? Sumiram? Os hotéis e parques agora têm a mesma representação? A escolha da simbologia adotada também é uma decisão de design e ela precisa ser coerente para que possa comunicar ao usuário o seu significado.

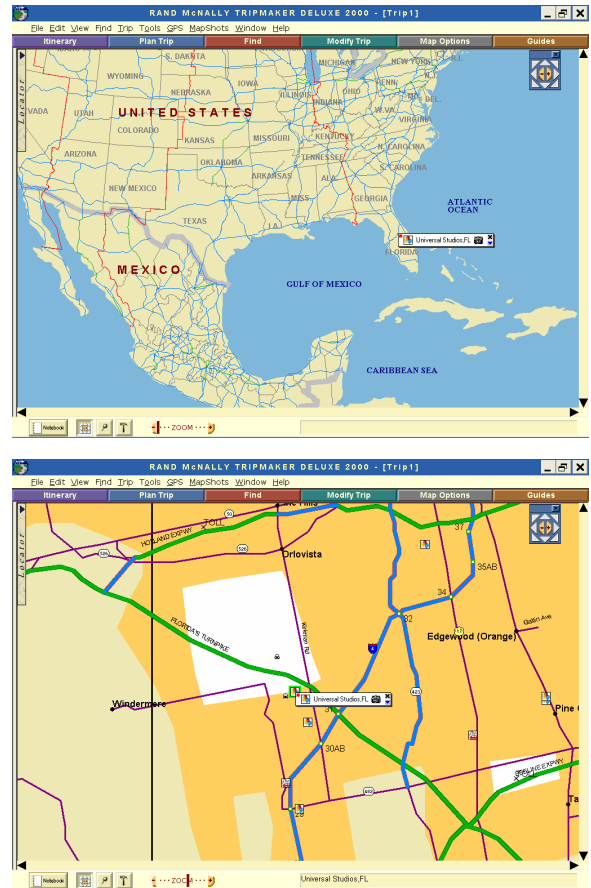


Figura 6. Mudança de escala com referência

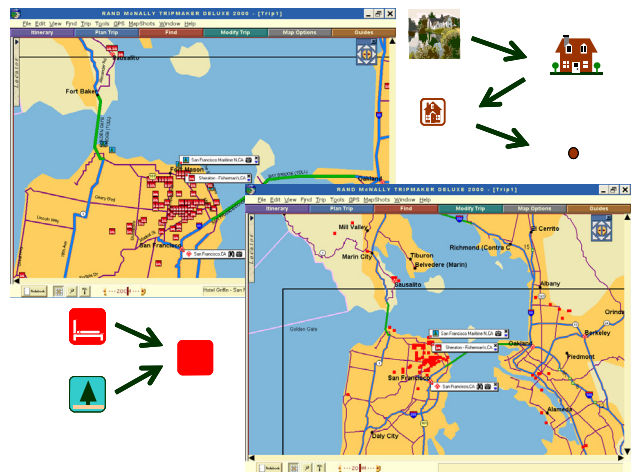


Figura 7. Modificação na apresentação dos objetos.

A mudança de escala é apenas uma das interações que pode afetar a tarefa do usuário. Qualquer ação que mude a apresentação do mapa também pode (ver exemplo da Figura 4). A possibilidade de se mudar a apresentação nos mapas interativos os torna muito mais interessantes que mapas em papel, mas também os torna muito mais difíceis de serem avaliados, pois não basta analisarmos os

signos estáticos, precisamos também analisar os signos dinâmicos decorrentes da interação⁴.

MÉTODO

O método que está sendo proposto avalia a qualidade da interação baseada em mapas, com vistas à realização de tarefas de localização ou traçado de trajetos. Nesta avaliação, os signos (a maioria deles cartográficos) que aparecem nos mapas para interação desempenham um papel particularmente importante, uma vez que fazem parte da linguagem de saída (apresentação), entrada (comando e controle) ou ambas, na interface do sistema.

O método emprega técnicas analíticas para definir o contexto da aplicação e o perfil do usuário, bem como para interpretar os resultados obtidos. Emprega também uma técnica empírica, de testes e entrevistas com usuários, para comparar parte dos dados obtidos com as técnicas analíticas.

O método é realizado em 5 etapas (Figura 8). Na primeira etapa é realizada uma inspeção na interface do sistema e no conteúdo do módulo de ajuda (*help*) [4]. O resultado é uma leitura da metacomunicação do Designer. Na segunda etapa, baseado nos resultados da primeira, é estabelecido o contexto de uso da aplicação. Definimos quais os usuários típicos do sistema e quais os tipos de tarefas que eles irão querer resolver com o sistema. Como resultado geramos alguns cenários de uso. O contexto amplia os possíveis cenários de uso, pois permite que sejam feitas suposições sobre outras utilizações relacionadas à tarefa. Na terceira etapa são feitos os testes com os usuários baseados nos cenários estabelecidos na segunda etapa. A interação dos usuários com o sistema é gravada para uma análise posterior. Na quarta etapa são inspecionadas as seqüências de mapas gerados, a fim de se identificarem rupturas (se houver) no contínuo semiótico entre as várias representações utilizadas pelos usuários. Na quinta etapa interpretamos o retrato do contínuo semiótico entre os mapas baseado na relação entre a e a tarefa que estava sendo realizada e a apresentação resultante da interação com o sistema.

1ª Etapa – Inspeção Semiótica dos Signos

A análise semiótica dos signos das interfaces dos mapas que sugerem o contexto para o qual foi abstraído (pretendido) deve ser realizada a partir de 3 pontos de observação que representam a ordem em que os usuários comumente exploram as aplicações computacionais [4,8]:

- (a) Uma leitura superficial da aplicação, captando os significados mais salientes dos *signos estáticos*;

- (b) uma visita exploratória da aplicação, captando os significados mais salientes dos *signos dinâmicos*;
- (c) uma leitura da documentação on-line oferecida⁵, trazendo informações ou explicações associadas aos signos estáticos e dinâmicos analisados em (a) e (b).

Este tipo de análise assegura apenas que as interpretações do analista são plausíveis e, dependendo diretamente de sua experiência, prováveis. O melhor indicador da qualidade da análise é a semiose do analista, que deve ser explicitada e justificada. Uma semiose bem explicitada e justificada oferece uma oportunidade valiosa de se investigarem processos interpretativos ricos em informações para aqueles que têm por objetivo comunicar idéias.

Durante esta etapa são verificados quais os elementos do mapa que podem ser vistos pelos usuários, modificados durante uma mudança de escala, modificados por uma interação do usuário, enfim, tudo que pode aparecer, desaparecer e ser modificado. Estes elementos são o foco da nossa análise, são eles que, ao sofrerem alguma alteração, vão nos dar indícios de como está o contínuo semiótico para o usuário.

2ª Etapa – Contexto da Aplicação

No caso das interfaces baseadas em mapas, para estabelecer o contexto de uso da aplicação, precisamos identificar:

1. Quem são os usuários e quais os seus papéis;
2. Quais os tipos de informações (diferentes, adicionais) disponibilizadas para as diferentes situações que os usuários podem se encontrar;
3. Quais os elementos de interface e de apresentação dos objetos no mapa que são relevantes para as diferentes situações.

Como resultado desta análise podemos responder as seguintes perguntas:

- Qual o contexto para um determinado mapa?
- Quais são os elementos do mapa que são decisivos para fazê-lo dependente de um contexto particular?
- Quais os elementos de um mapa que sugerem o contexto para o qual ele foi pretendido?

⁴ Segundo da Silva et al.(2003) os signos da interface podem ser estáticos ou dinâmicos. Eles são estáticos, quando estão associados a uma primeira interpretação do usuário sobre um objeto disponível na aplicação, ou dinâmicos, quando resultantes de uma interação onde o usuário procura o real significado do objeto.

⁵ Os métodos de avaliação de software comumente utilizados não dão a devida importância à leitura da documentação *online* fornecida. Do ponto de vista da Engenharia Semiótica, a leitura desta metalinguagem em uma avaliação é importante, pois ela consiste em um canal privilegiado de comunicação do designer para o usuário, onde o designer tem a oportunidade de expressar a mensagem pretendida para os usuários [23].

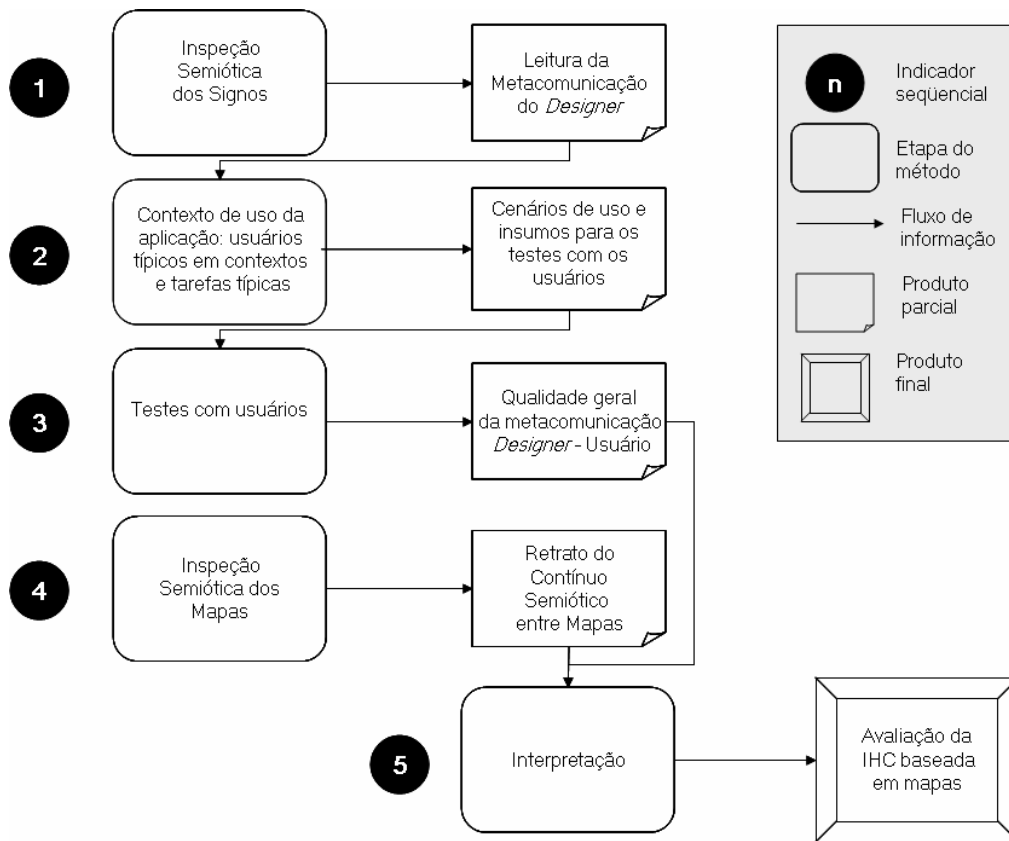


Figura 8. Método para avaliação de mapas

Ao final desta etapa temos as informações necessárias para construirmos os cenários de uso do sistema. São desenvolvidos apenas alguns cenários para testar os pontos de possíveis rupturas de comunicação identificados. Para cada cenário estabelecemos quais os elementos relevantes que estão disponíveis no sistema e o que precisa estar visível ou referenciado. Estes elementos uma vez identificados servirão como base para se analisar as apresentações durante a Inspeção Semiótica dos Mapas.

3ª Etapa - Testes com Usuários

O que se quer avaliar na inspeção semiótica são as mudanças nas apresentações dos mapas que ocorrem durante as interações dos usuários. Queremos inicialmente entender como a forma de apresentação dos objetos nos mapas pode afetar a realização de uma tarefa.

Os testes com usuários são uma forma de se verificar como as pessoas irão reagir aos vários mapas apresentados durante a realização da tarefa.

O ideal nesta etapa é realizar um questionário pré-teste para obter informações sobre o perfil do usuário. Após o teste o usuário também pode responder um questionário aberto sobre as suas impressões do sistema como facilidades, dificuldades etc. Finalmente, o avaliador que esteve acompanhando a interação deve entrevistar o usuário para confirmar qualquer problema que ele tenha identificado na interação do usuário. Esta entrevista é feita com o objetivo de esclarecer pontos que poderiam deixar dúvidas durante o processo de inspeção. Isto

ajudará o avaliador a casar a sua semiose com semiose do usuário.

4ª Etapa - Inspeção Semiótica dos Mapas

Nesta etapa o que se quer saber é se a variação dos elementos do mapa durante a realização de uma tarefa ou mudança de escala produz uma ruptura na continuidade. As perguntas relevantes são:

- A apresentação após uma ação do usuário mostra ou indica os elementos necessários para a tarefa?
- Existem outras apresentações disponíveis para a mesma tarefa?
- As apresentações possíveis (*zoom*, etc.) mantêm os elementos visíveis ou há alguma indicação para as suas existências?
- Nas apresentações os elementos aparecem, somem ou se modificam?
- A modificação da apresentação quebra o contínuo semiótico?

Para responder estas perguntas analisamos a gravação da interação do usuário com o sistema. A análise consiste, basicamente, em verificar se houve mudança na apresentação do mapa após uma ação do usuário. Se houve, verificamos como se apresentam os mapas, antes e após a interação, com relação a:

- que parte da tarefa o usuário se encontra;
- mudança de escala;

- modificação dos elementos apresentados de um modo geral; e
- modificação dos elementos que representam os objetos relacionados à tarefa.

Devemos nesta fase levar em consideração que existem dois tipos de ações. As relacionadas diretamente com as modificações que podem ocorrer nos mapas, como por exemplo, para encontrar um lugar o usuário digita o nome do lugar numa caixa de texto ou interage com o mapa até que o lugar apareça no mapa modificando sua apresentação anterior. As ações relacionadas indiretamente como, por exemplo, preenchimento de caixas de texto, interações com menus ou configurações que não afetam as apresentações.

Todas as ações dos usuários que podem estar relacionadas com rupturas na comunicação designer-usuário são registradas e comentadas. Porém, apenas as que afetam diretamente as apresentações dos mapas é que serão analisadas e interpretadas posteriormente.

Estas variações na apresentação dos objetos são abstrações e precisam estar de acordo com as possíveis abstrações feitas pelos usuários ao realizar a tarefa. O desejável é que estas modificações que ocorrem normalmente durante a interação com os mapas contribuam para a realização da tarefa. Porém, quando encontramos modificações no mapa que fazem com que a sua apresentação não seja semioticamente contínua com a apresentação anterior, classificamos a ruptura em uma das seguintes classes de problemas:

1. Orientação (Onde eu estou?)
2. Navegação (Aonde posso /devo ir?)
3. Identificação de objetos (O que é isto? Onde está? Onde estava?)
4. Interpretação da simbologia (O que isto representa?)

Problemas de orientação podem ocorrer quando o usuário perde (deixa de ver) alguma referência usada para se localizar no mapa. Problemas de navegação geralmente ocorrem quando o sistema não provê signos indicadores do contexto atual. Alguns exemplos destes tipos de signos são: o mapa índice, o indicador de direção, o indicador de orientação, o uso de legendas e outros signos que auxiliam os usuários a identificarem a sua posição e o que há a sua volta. O problema da identificação dos objetos pode estar relacionado com a sua forma (de representação ou apresentação) ou a sua localização como, por exemplo, alguma coisa que deveria aparecer, mas não apareceu, ou alguma coisa que deveria estar presente, mas sumiu. Problemas de identificação de objetos podem levar a problemas de orientação, pois se o usuário não consegue identificar os objetos a sua volta, ele pode ficar perdido. A interpretação da simbologia geralmente é um problema encontrado nas escalas menores, onde não há muito espaço para apresentação e os símbolos podem acabar se sobrepondo, dificultando a compreensão do usuário.

5ª. Etapa – Interpretação

Nesta etapa queremos poder responder às seguintes perguntas:

- Existe uma quebra de continuidade entre o uso pretendido (pelo designer) e o uso real do mapa?

Se existe, precisamos identificar:

- Como manter o foco da tarefa durante a interação com mapas?
- Quais elementos precisam ser mantidos visíveis para que o usuário não se perca?
- Como modificar a apresentação dos objetos de modo que o usuário consiga manter a referência do objeto?

As respostas para tais perguntas vêm da análise das rupturas na etapa anterior. Dependendo dos tipos de problemas encontrados, um especialista em IHC pode identificar soluções para o *redesign* da aplicação.

Apesar de registrarmos os problemas ocorridos e classificá-los, a interpretação é feita independentemente da quantidade de problemas (ou classes de problemas) encontrados. Uma única ocorrência de um problema significativo de interação deve ser tratada com a devida importância. Por este ser um tipo de avaliação subjetiva a interpretação deve ser feita por um especialista em IHC.

CONCLUSÕES

A interação nos *desktops* GIS é pobremente projetada e suas funcionalidades são raramente testadas. Apesar de algumas aplicações serem direcionadas para atender a objetivos e atividades bastante contextualizados, a concepção de que as aplicações tradicionais de SIGs tratam de tarefas não muito bem estruturadas parece estar presente em quase todas as interfaces.

O ISIM, inicialmente proposto para avaliação de aplicações em que parte da interface é baseada em mapas, procura identificar uma classe de problemas básicos decorrentes de interação, porém significativos do ponto de vista de utilização e interpretação de mapas de um modo geral. As classes de problemas que o método pretende avaliar são apenas um subconjunto de possibilidades. Durante a interação com mapas o usuário pode querer colocar marcadores, adicionar legendas, desenhar no mapa, medir distâncias etc. Estas interações que podem ser feitas nas IMs estão relacionadas com uma outra classe de problemas: a manipulação direta dos mapas.

Apesar dos métodos de inspeção fazerem uma maior cobertura da interface do que os testes com tarefas pré-selecionadas, as avaliações empíricas com usuários podem trazer a tona novos problemas que não foram encontrados durante a inspeção.

Os filmes dos testes com os usuários também servem para roteirizar a inspeção, contornando o fato de que ela pode ser completamente aberta (uma das principais dificuldades de falar em IHC para GIS). Porém, a roteirização a partir de 1, 2 ou *n* filmes, são bastante limitadas. São só 1, 2 ou *n* caminhos dentro de um

conjunto aberto de possibilidades. Os métodos pautados em teorias semióticas são métodos que potencializam sempre, em variadas direções, a semiose do avaliador, e aí toda semiose é um processo com ilimitadas possibilidades. A análise semiótica faz, então, com que os filmes sejam um fator coesivo e vantajoso.

Para contornar o subjetivismo absoluto do avaliador os dados da inspeção são comparados com o usuário em ação e com as entrevistas pós-teste. As entrevistas são, particularmente, enriquecedoras, pois acrescentam novos pontos de vista para que possamos revisitar o filme com outras questões em mente.

O método desenvolvido pode ser utilizado com outras técnicas ou métodos de avaliação. Por ser um método para avaliar características específicas das interfaces baseadas em mapas, ele não dá ênfase a outros problemas de comunicabilidade ou de usabilidade que possam ser encontrados na interface.

Como trabalhos futuros podemos pensar em:

- Acrescentar ao método outras classes de problemas específicos de SIGs;
- Estender o método, não só para avaliar mas para guiar o desenvolvimento de novas aplicações;
- Testar o método em outros tipos de aplicações de SIGs;
- Testar método em aplicações semelhantes de visualização, mas não necessariamente de mapas, para tentar ver o quão generalizáveis são os resultados.

AGRADECIMENTOS

Clarisse Sieckenius de Souza agradece o apoio do CNPq a suas pesquisas. Maria Lúcia Arraes Seixas agradece o apoio do CNPq e do Tecgraf/PUC-Rio a suas pesquisas. As autoras agradecem aos participantes da pesquisa e aos colegas do SERG, por sua contribuição para este projeto.

REFERÊNCIAS

1. BEST-GIS. Guidelines for best practice in user interface for SIG. The European Commission, DGIII – Industry. *ESPRIT Programme*. Geneva: Geographical Information Systems International Group, 1998.
2. Câmara, G., Casanova, M. A.; Hemerly, A., Magalhães, G. C., Medeiros, C. M. B. Anatomia de Sistemas de Informação Geográfica. *10a. Escola de Computação*, UNICAMP, 1996.
3. Cartwright, W.E. and Hunter, G. Towards a Methodology for The Evaluation of Multimedia Geographical Information Products, *Geoinformatica*, Vol.5, No.3, pp. 291-315, Kluwer Academic Publishers, 2001.
4. da Silva, E. J., de Souza, C. S., Prates, R. O., and Nicolaci-da-Costa, A. M. What they want and what they get: A study of light-weight technologies for online communities. *Proc. of the Latin American Conference on Human-Computer Interaction (CLIHC'2003)*, pp. 135-146.

5. de Souza, C. S. The Semiotic Engineering of User Interface Languages. *International Journal of Man-Machine Studies (1993)*, v. 39, pp. 753-773.
6. de Souza, C. S., Barbosa, S. D. J., and Prates, R. O. A Semiotic Engineering Approach to User Interface Design. *Knowledge Based Systems (2001)*. Amsterdam, v. 14, n. 8, pp. 461-465.
7. de Souza, C. S., Barbosa, S. D. J., and da Silva, S. R. P. Semiotic Engineering Principles for Evaluating End-user Programming Environments. *Interacting with Computers (2001)*, 13 pp 467-495.
8. de Souza, C. S., Nicolaci-da-Costa, A. M., da Silva, E. J., Prates, R. O. Compulsory institutionalization: the paradox of computer-supported informal social processes. *Interacting with Computers (2004)*, in press.
9. de Souza, C. S. *The Semiotic Engineering of Human-Computer Interaction (in press)*. Acting with Technology series. MIT Press.
10. Fairbairn, D., Adrienko, G., Adrienko, N., Buziek, G., and Dykes J. Representation with Cartographic Visualization. *Cartography and Geographic Information Science (2001)*, 28 (1): 13-28.
11. Frank, A. U. and Timpf, S. Multiple Representations for Cartographic Objects in a Multi-Scale Tree - An Intelligent Graphical Zoom. *Computers and Graphics: Special Issue on Modelling and Visualization of Spatial Data in SIG (1994)*, 18(6): 823-829.
12. Freitas, C.M.D.S., Luzzardi, P.R.G., Cava, R.A., Winkler, M.A.A., Pimenta, M.S., Nedel, L.P. Evaluating Usability of Information Visualization Techniques. In *Proc. of IHC 2002 – V Symposium on Human Factors in Computer Systems*, pp. 40-51.
13. MacEachren, A. M. *How Maps Work: Representation, Visualization, and Design*. Guilford Press, New York, 1995.
14. MacEachren, A. M., and Kraak, M.J. Research Challenges in Geovisualisation. *Cartography and Geographic Information Science (2001)*, 28:1, 1-11.
15. Montello, D. R., and Golledge, R. G. Scale and Detail in the Cognition of Geographic Information. *Varenius Project Report Specialist Meeting*, 1999.
16. Norman, D. A. Cognitive engineering. *User centered system design* by Norman, D. A. and Draper, S. W. (Eds), Lawrence Erlbaum Associates, Inc., pp. 31-61, 1986.
17. Peirce, C.S. Semiótica. Tradução de “*The Collected Papers of Charles Sanders Peirce*” (1990). Editora Perspectiva, 3ª. Edição, 2000.
18. Prado, A.B. e Baranauskas, M.C.C. Avaliando a meta-comunicação designer-usuário de interface. *Anais do IHC'2000 – III Workshop sobre Fatores Humanos em Sistemas Computacionais*, pp. 33-42.
19. Prado, A.B., Baranauskas, M.C.C. e Medeiros, C.M.B. Cartography and Geographic Information Systems as Semiotic Systems. In *Proc. 8th ACM GIS International Symposium (2000)*, pp. 161-166.

20. Prates, R. O., Barbosa, S. D. J., and de Souza, C. S. A Case Study for Evaluating Interface Design through Communicability. *DIS'00 (2000)*, pp. 308-316.
21. Raper, J., Dykes, J., Wood, J., Mountain, D., Krause, A., and Rhind, D. A framework for evaluating geographical information. *Journal of Information Science (2002)*, 28 (1).
22. Schimiguel, J., Baranauskas, M.C.C., Medeiros, C.M.B. Modelando a Interface de Aplicações em Sistemas de Informação Geográfica como Espaços de Comunicação. *Anais do IHC'2001 – IV Workshop sobre Fatores Humanos em Sistemas Computacionais*, pp. 157-168.
23. Silveira, M. S., Dias, M. C., Quental, V. Helping the Composition of Help Contents. PorTAL - Portugal for Natural Language Processing. *Lecture Notes in Artificial Intelligence (LNCS/LNAI) Series (2002)*. Springer-Verlag.
24. Slocum, T. A., Blok, C., Jiang, B., Koussoulakou, A., Montello, D. R., Fuhrmann, S. and Hedley, N. R. Cognitive and usability issues in geovisualization: a research agenda. *Cartography and Geographic Information Science (2001)*, 28(1), pp61-76.
25. Timpf, S. Abstraction, levels of detail, and hierarchies in map series. *Spatial Information Theory & Cognitive and computational foundations of geographic information science (International Conference COSIT'99)*, edited by Freksa, C. and Mark, D. M. *Lecture Notes in Computer Science 1661(1999)*. Berlin-Heidelberg: Springer, pp. 125-140.
26. Tobón, C. Usability Testing for Improving Interactive GeoVisualization Techniques. *CASA – Centre for Advanced Spatial Analysis. Working Paper Series, 2002*.
27. Traynor, C. and Williams, M. G. "Why are Geographic Information Systems Hard to Use?" In Conference Companion of ACM SIGCHI. *Conference on Human Factors in Computing Systems, CHI '95*, pp. 288-289.
28. Traynor, C and Williams, M. G. Usability of Textual and Visual Query representations for Geographic Information Systems. *GIScience 2000*.
29. Winter, S. Communication about Space. *Transactions in GIS Vol. 8 Issue 3*, pp. 291-296, June 2004. <http://www.sli.unimelb.edu.au/winter/pub.htm>

Using an Interaction Model to Support Communication among HCI Design Team Members from Multidisciplinary Backgrounds

Maíra Greco de Paula

Simone Diniz Junqueira Barbosa

Departamento de Informática, PUC-Rio
R. Marquês de São Vicente, 225
Rio de Janeiro, RJ, Brazil – 22453-900
{mgreco, simone}@inf.puc-rio.br

RESUMO

Diversas representações de design têm sido propostas na área de Interação Humano-Computador (IHC) para apoiar o design centrado no usuário. Modelos de tarefa em um alto nível de abstração e linguagens de especificação detalhada da interface de usuário estão entre os mais utilizados. Já cenários e storyboards são representações comumente utilizadas para promover a comunicação numa equipe de design. Entretanto, estas representações não provêem uma visão global da solução, i.e., o comportamento aparente da aplicação do ponto de vista do usuário. Este artigo argumenta que tal visão global é importante e funciona como um *blueprint* da aplicação, que permite que profissionais formados em diferentes disciplinas compartilhem a mesma compreensão da essência da aplicação. Argumenta ainda que a adoção da metáfora de interação como uma conversa pode fazer face a este desafio. Para possibilitar isto, apresenta uma linguagem de modelagem de interação que permite que os designers construam um *blueprint* do conjunto de interações que poderão ocorrer durante o uso da aplicação.

ABSTRACT

A number of design models and representations have been proposed in the area of Human-Computer Interaction (HCI) to support user-centered system design. High-level, abstract task models and detailed interface specification languages are among the most widely used. Scenarios and storyboards are among the most common representations used for communication in multidisciplinary design teams. However, they don't give a global view of the solution, i.e., the application's apparent behavior from the users' point of view. We argue that such a global view is important and serves as a blueprint of the application that allows professionals from multidisciplinary backgrounds to share the same understanding of the essence of the application. In this paper, we argue that an interaction-as-conversation metaphor may face this challenge, and present an interaction modeling language that allows designers to

build a blueprint of the range of interactions that will possibly take place when the application is used.

Keywords

design representations, interaction modeling, multidisciplinary design teams

INTRODUCTION

The design and development of computer applications require knowledge that goes beyond Informatics. Multidisciplinary teams are necessary to broaden the understanding of the problem, to devise more adequate and innovative solutions, and to evaluate their impact on the users' work practices and lives. In various stages of the design process, we need professionals from multidisciplinary backgrounds. During user analysis, for instance, psychologists can make invaluable contributions for the elicitation and analysis of user-related information, as well as for devising user testing. At the user interface design stage, graphics designers help defining the presentation aspects of the user interface.

In [22], a case study is presented in which researchers from various areas involved in the design of information technology were interviewed about the advantages and disadvantages of multidisciplinary work. Among many of the problems they found, we may cite: the lack of an integrated view as a starting point for a project and loss of focus.

A number of design models have been proposed in the area of Human-Computer Interaction (HCI) to support user-centered system design [16]. Among these models, we find scenarios, task models and prototypes are the most widely used. Many attempts to foster communication among HCI team members (graphics designers, psychologists, sociologists, etc) are based on scenarios and prototypes.

Scenarios are a very useful representation to depict and envisage possible usage situations of the application [8, 20]. Because they are written in natural language, scenarios are valuable for validating the designers' assumptions and decisions with users, facilitating their understanding of the proposed solution. However, to completely represent an application, it would be necessary to create a large number of scenarios. This makes it difficult to achieve a global view of the

VI Simpósio sobre Fatores Humanos em Sistemas Computacionais — Mediando e Transformando o Cotidiano. Hotel Bourbon, Curitiba, 17 a 20 de outubro de 2004. UFPR, CEIHC—SBC.

application and the consequences of the designers' decisions, which need to be shared by all HCI team members.

Storyboards and prototypes are also extensively used by HCI designers. They allow designers to grasp the envisaged behavior of the application. However, in the initial design stages, when the HCI design team still focuses on *what* the users need, and not *how* they will achieve their goals using the system, the construction of storyboards or prototypes may be viewed as premature for representing the essence of the application. Moreover, because building prototypes may consume a lot of time and effort, designers may be reluctant to experiment new ideas or revise minor issues.

In this paper, we argue that we need a representation devised to facilitate the communication among the HCI design team members. It is supposed to be a reference material to which all members of the design team can refer during the design process. This representation should contain a minimal set of elements which are required for building a shared understanding of what needs to be done (the “essence” of the application), motivating the reflection of each professional in his own line of work. It is important to note that we do not aim at formal specification or automatic user interface generation, such as the models and approaches proposed in [14, 24]. Instead, we focus on supporting designers in the reflective process of conceiving (i. e. designing) a solution [21].

To face this challenge of representing the essence of the application, we propose the use of an interaction model that provides a global view of the apparent behavior, from the users' point of view. Also, we describe how the interaction model can be used to specify design patterns for well-known solutions.

The next section argues for the importance of communication-centered design to promote communication among design team members, and why an interaction model may be used for this purpose. In the third section, we present MoLIC, the proposed interaction model to play this role. In section four, we present a possible structure for HCI design patterns. In section five, we describe how MoLIC can be used to specify design patterns. We conclude the paper with a brief discussion about the usage of MoLIC by a multidisciplinary team dedicated to both HCI design and software development.

COMMUNICATION-CENTERED DESIGN

Interaction design is concerned with “designing interactive products to support people in their everyday and working lives” [19, p.6]. From a visual designer's perspective, HCI is like graphics design, industrial design, and architecture, in that it is “concerned most directly with the user's experience of a *form in the context of a specific task or problem*, as opposed to its functional or aesthetic qualities in isolation” [15, p.1].

Within HCI, Semiotic Engineering [9] has emerged as a semiotics-based theory that describes and explains human-computer interaction phenomena, adopting primarily a *media* perspective on the use of computer artifacts [13]. Semiotic Engineering focuses on the engineering of signs that convey what HCI designers and users have in mind and what effect they want to cause in the world of things, practices, ideas and experiences. The interface signs constitute a message sent from designers to users, representing the designers' solution to what they believe is the users' problems, what they have interpreted as being the users' needs and preferences, what the answer for these needs is and how they implemented their vision as an interactive system.

Our work builds on semiotic engineering in the sense that, in order to increase the chances of engineering adequate signs at the user interface to convey the designers' vision, we propose to engineer the designer's message to be conveyed first to other designers. If designers are unable to convey their vision to each other and to every stakeholder, they will hardly succeed in conveying it to users. If, on the other hand, we succeed in promoting designer-designer communication via a representation as a communication artifact, we will be better equipped to communicate with users through the user interface. Our envisaged representation is not only a carrier of information. Instead, we want to emphasize its social role during the design process [26], to promote understanding of and communication among professionals from multiple disciplinary fields. This way, we aim to achieve a communication-centered approach to interactive software design and development.

So, we need to elaborate a representation or a modeling language that fosters multidisciplinary discussion, represents the designers' vision of the essence of the system being designed and thus serves as a reference of *what* needs to be done (the “essence” of the system), to which every professional must commit. The way each professional works when designing his own portion of the system (the “accidental” aspects of the system) may be done *however* he wishes, as long as the common vision is respected. Of course there will be breakdowns along the way, and the design unfoldings in one or more disciplines may require that changes in the common vision are made.

We have argued for the need to elaborate a common representation to describe the designers' vision. But what exactly do we choose to represent? How should it be related to the software or to the user interface? We have made an attempt to represent the designers' vision as an *interaction model* [4, 17, 18].

The underlying assumption is that the designers' vision of *what* needs to be developed (and not *how*) should reflect the application's apparent behavior, i.e., the interactive behavior from the users' point-of-view. This assumption is in line with Frederick Brooks' suggestion that “the separation of architectural effort from implementation is a very powerful way of getting

conceptual integrity on very large projects” [7]. Although this statement will be considered obvious now by any software engineer, Brooks’ elaboration of what an architecture is may not be as widely known: “by the architecture of a system, I mean the complete and detailed specification of the user interface” [7, p.45 (1995 edition)]. In Requirements Engineering, a similar consideration may be found in [5], which describes the advantage of employing a user’s manual as a requirements specification.

One question to be answered is: who may create this common representation? HCI designers, software engineers, graphical designers, or other professional? We believe that the first draft representation of the common vision will emerge during the initial design sessions. We assume it will be collectively built, and not assigned to a single professional. As the designers’ vision mature, each professional will then create or specify the representations concerning his own field.

In the next section, we describe a possible representation language for modeling interaction. It is called MoLIC, which stands for “Modeling Language for Interaction as Conversation” [4, 17, 18].

INTERACTION MODELING WITH MOLIC

Our goal when creating MoLIC was to give designers a global view of the apparent behavior of the system, including alternative and breakdown interaction paths. Based on Semiotic Engineering [9], MoLIC represents interaction as threads of conversation users may have with the system¹, without yet specifying in detail the storyboards or the user interface itself [18].

Each thread of conversation represents courses of action to accomplish a certain goal or to take remedial action when a communicative breakdown occurs. MoLIC was devised to explicitly represent all possible interactions, allowing designers to inspect the model for inconsistencies.

A MoLIC diagram may be seen as a graphical view of the whole set of scenarios, from the user’s perspective. We do not aim to substitute scenarios, but to organize the relevant information they contain. As with scenarios, we emphasize the need to describe breakdown situations that may occur during interaction. In scenarios, however, there are often too many decisions to be made in moving to software specification. And, unfortunately, these decisions are seldom recorded for future reference or verification of the final product. We propose to use MoLIC to structure scenarios in such a way as to

¹ In fact, the conversation takes place between the user and a part of the system’s user interface called “designer’s deputy”, as defined in the Semiotic Engineering theory of HCI [9]. The semiotic engineering concepts used on the theoretical foundation for MoLIC and the complete MoLIC notation may be found at [18].

overcome the problems found in scenarios for defining the interaction.

MoLIC’s Notation

The interaction model basically comprises scenes, system processes, and user/system’s utterances —called transitions— that connect scenes and processes to form conversation threads. A scene represents a user–system conversation about a certain matter or topic, in which it is the user’s “turn” to make a decision about where the conversation is going. This conversation may comprise one or more dialogues, and each dialogue is composed of one or more user/system utterances. In other words, a scene represents a certain stage during execution where user–system interaction may take place.

In MoLIC’s diagrammatic representation, a scene is represented by a rounded rectangle, whose text describes the topics of the dialogues that may occur in it, from the users’ point-of-view. A system process is represented by a black rectangle, representing something users do not perceive directly. By doing so, we encouraged the careful representation of the system’s utterances about the result of system processing, as the only means to inform users about what occurs during interaction. Transitions represent changes in the conversation topic or focus. This may happen due to the user’s choice or to a result in system processing, as represented in the arrow label.

Some scenes may be accessed from any point in the application, i.e., from any other scene. The access to these scenes, named ubiquitous access, is represented by a transition from a grayed scene which contains a number following the letter U, for “ubiquitous”.

Figure 1 illustrates a MoLIC diagram for a banking application which achieves the following goals: check balance, check account statement, and transfer between accounts.

Interaction breakdowns in MoLIC

Error prevention and handling are an inherent part of the conversation between users and system, and not viewed as an *exception*-handling mechanism. The designer should convey to users not only how to perform their tasks under normal conditions, but also how to avoid or deal with mistaken or unsuccessful situations. Some of these situations may be detected or predicted during interaction modeling. When this is the case, we extend the diagrammatic representation with breakdown tags, classifying the interaction mechanisms for dealing with potential or actual breakdowns in one of the following categories:

Passive prevention (PP): errors that are prevented by documentation or online instructions. For instance, about which users have access to the system, what is the nature of the information expected (and not just “format” of information).

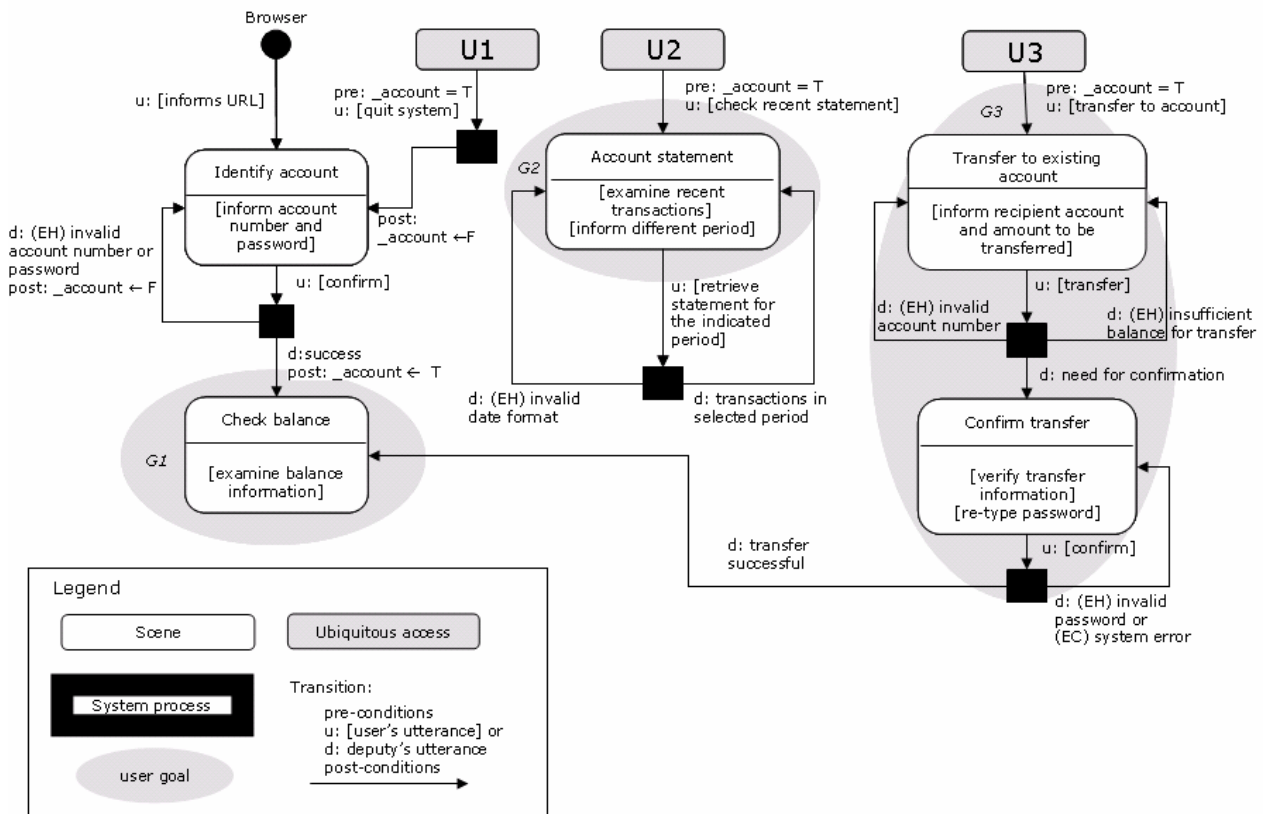


Figure 1: An abbreviated MoLIC diagram.

Active prevention (AP): errors that will be actively prevented by the system. For instance, tasks that will be unavailable in certain situations. In the interface specification, this may be mapped to making widgets disabled depending on the application status or preventing the user to type in letters or symbols in numerical fields, and so on.

Supported prevention (SP): situations which the system detects as being potential errors, but whose decision is left to the user. For instance, in the user interface, they may be realized as confirmation messages, such as “File already exists. Overwrite?”)

Error capture (EC): errors that are identified by the system and that should be notified to users, but for which there is no possible remedial action. For instance, when a file is corrupted.

Supported error handling (EH): errors that should be corrected by the user, with system support. For instance, presenting an error message and an opportunity for the user to correct the error.

MoLIC’s Textual Specification

MoLIC has both an abbreviated and an extended diagrammatic representation. The extended MoLIC diagram represents not only dialogues. This is represented by the signs at the user interface that either the user or the system manipulates at each moment (represented by the marks ? and !, respectively). Here, we use the term *sign* to denote any given element at the user

interface to which a user may attribute meaning with respect to his/her goal or task, or to the application itself.

The extended representation may also include additional information about each sign, such as: whether the user must provide a value for the sign, whether the system provides a default value for the sign (and how this default value is calculated), the abstract user interface widget associated to the sign (simple choice, free text, etc.), the degree of knowledge the user has about the sign, additional information to be conveyed to users to help them understand and/or manipulate the sign, and any additional annotations designers may want to represent. Figure 2 shows the extended representation for the “Identify account” scene.

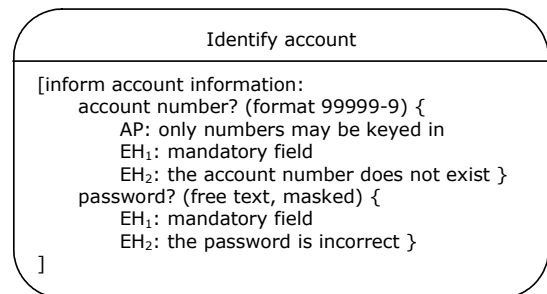


Figure 2: Extended representation of a scene.

We propose to use MoLIC to represent the designers’ vision, i.e., *what* needs to be the apparent behavior of the system. From this vision, each professional would then derive his own representations and models, to specify and refine *how* the system will be put together as envisioned.

DESIGN PATTERNS AS COMMUNICATIVE ARTIFACTS

Design patterns are, by nature, communicative artifacts, in which “problem and solution of each pattern [is presented] in such a way that you can judge it for yourself, and modify it, without losing the essence that is central to it” [2, p. xi]. Architectural patterns were originally defined in a narrative style, enriched by relevant contextual information [1]. They described the motivation and rationale from the point of view of the persons who would use the “architectural product” described by the pattern. Software design patterns, on the other hand, are defined in a more structured format, and the motivation is typically described from the point of view of system components [11]. HCI design patterns have tried to bring together the advantages of a more structured description, which eases reading and scanning, and the richness of contextual information [6, 25]. However, they fall short of directly supporting design, in that they provide information about good design solutions, but not a tool to model the resulting application using them.

Following both Alexander and Gamma et al., Borschers proposes that HCI design patterns should have the following structure:

- the *name* of the pattern, (to convey the idea of the pattern in one or a few words, to make it easy to remember and refer to when thinking about or discussing design solutions),
- a *ranking* of its validity (zero, one, or two asterisks, which indicate the degree of confidence that the authors had in the pattern — two asterisks are believed to describe a true invariant),
- a *picture* as an example of its application,
- the *context* in which it is to be used (which larger-scale patterns this specific pattern helps to implement),
- a short *problem statement* (summary of the general situation that the pattern addresses),
- a more detailed *problem description* with empirical background (problem description using the concept of competing “forces”, which the patterns aim to resolve or balance),
- the central *solution* of the pattern (clear but generic set of instructions that can be applied in varying situations),
- a *diagram* illustrating the solution (graphical sketch of the solution and its major constituents),
- and finally *references* to smaller patterns (author recommendations to implement and further “unfold” the solution of the current pattern).

In the next section, we describe how MoLIC can be used as a diagram to illustrate the solution described in a pattern, serving as a concrete resource for the interaction specification.

USING MOLIC TO DESCRIBE DESIGN PATTERNS

One of the major advantages of software design patterns as defined in [11] is that each pattern includes one or more diagrams that may be put together into a blueprint

of the application, i.e., a software specification. In this paper, we propose to follow a similar approach and use a notation to represent HCI patterns that may be used to *specify* user-system interaction.

Some HCI patterns include diagrams that look like storyboards, representing the concrete user interface without irrelevant details. However, many interesting patterns are related to *interaction paths* that span beyond individual interface elements. For these patterns, we need a representation language at a higher level of abstraction, and which focuses on the possible interactions users may have with the system. Differently from the diagrams in software design patterns, the interactions should be represented from the user’s point-of-view, i.e., without unnecessary details about the system’s internal functionality. Moreover, the language should support the representation of patterns that aren’t specific to a single user interface style or environment, so the pattern will not be unnecessarily limited.

We propose the use of MoLIC for interaction specification for representing HCI design patterns. A partial interactive design solution would fill the “diagram” slot proposed by Alexander and his colleagues.

Some application-specific interactions cannot be captured in patterns. An interaction representation language should be able to organize and compose both specific and pattern-derived interaction specifications into an application “blueprint”. This will help designers not only to build applications faster, but also to develop a more coherent whole.

As suggested by [12], it may also be interesting to create task patterns. However, such patterns are usually domain-specific, and do not point directly to interactive solutions.

CONCLUDING REMARKS

In this paper, we have argued that to support communication among HCI designers from multidisciplinary backgrounds, we need a representation that provides a global view of the apparent behavior of the application, from a users’ point-of-view, i.e., its essence. This representation is to be as “neutral” as possible regarding the different disciplines, i.e., it should not inherit the assumptions and beliefs of any one discipline involved in HCI design. This will allow for a shared understanding of the representation by professionals from all the involved disciplines, and its usage as a resource for each professional to proceed specifying or detailing the application in his field of work.

We have presented MoLIC, an interaction modeling language that aims to play the role of the common representation to describe the designers’ vision. We have been conducting studies of this approach involving HCI designers, graphics designers, psychologists, and software developers. In our preliminary findings, the usage of MoLIC as a specification tool for the surfacing

behavior of the application was found to be effective by all team members.

In a small case study, a model sketched using MoLIC was presented to a psychologist, a graphics designer, and two software developers. They were interviewed to give their impressions about the usefulness of the model and the quality of the notation itself.

According to the graphics designer, the idea of using an interaction model in the process of user interface design leverages their work and makes sure that relevant HCI decisions are not lost. In his view, MoLIC can be used as a resource to support the activities involved in graphics design, because it provides a global view of the user interface and interaction processes. This representation allows them to focus on the design of user interfaces that better align with the shared vision of the solution. There were a few doubts regarding the notation for the signs definition, but they were quickly resolved by using the notation guide, by consulting other HCI professionals involved in the project and with practice. Moreover, the graphics designer expressed that he would like to receive from the HCI team, together with the MoLIC representation (interaction diagram and signs), a definition of a choice of widgets for which sign, which they could resort to when creating the visual structure of the user interface.

The interview with the psychologist proved to be a pleasant surprise. Earlier we thought he would feel uncomfortable dealing with a graphical notation, because psychologists don't have a tradition of working with this kind of notation. According to him, the notation can be easily understood and his work can generate many resources for the construction of an interaction model in MoLIC, such as the users' expectations, wishes and will regarding the presence or absence of certain signs in the user interface. From his point of view, the greatest benefit MoLIC brings to psychologists is related to the role of elements represented in the model in supporting the development of interview scripts to be used in the interface evaluation stages. In his opinion, the prime resources to build the scripts are represented in the global interaction map. The representation of the "essence" of the application in MoLIC is thought to expedite the psychologists' work and increase the scope of their interviews, which become more comprehensive and complete.

The system developers who were interviewed told us that the greatest benefit brought about by MoLIC is the interaction specification, which they usually lack when using traditional software engineering models and representation languages. They typically receive from HCI designers storyboards as a resource to implement the user interface, but the interaction itself is not represented in them. However, they felt they had a lot of "avoidable" work in mapping or translating the signs represented in the interaction model into the system's data model.

MoLIC helps designers acquire a global view of the application they are conceiving, from a user's standpoint. The diagrammatic version of MoLIC does not include user interface details that belong to any specific scene; this important issue has been addressed by other HCI-related studies, such as storyboarding, which is already used in certain HCI pattern languages.

HCI patterns should attempt to bring together the advantages of both architectural and software design patterns: they should include not only rich, contextual information and design instructions, but also actual HCI design specifications that, together with application-specific specifications, compose the entire application's HCI design.

The design solutions represented graphically in current HCI patterns are mostly based on storyboards. We have argued for the need of a language in a higher level of abstraction that is able to represent interaction (and not just pieces of the user interface). MoLIC can be used as an interaction modeling language for diagrammatically representing higher-level HCI design patterns.

We are currently evaluating the use of MoLIC for representing HCI design patterns. Another interesting issue is to assess the usefulness of MoLIC and HCI design patterns for evaluating existing products.

Besides using MoLIC in multidisciplinary HCI design teams, we are now investigating the use of MoLIC as a concrete resource for software design as carried out by software engineers. We have described elsewhere some simple strategies for deriving partial UML class and sequence diagrams from a MoLIC representation [17].

We are also elaborating a more extensive study to investigate the benefits of using MoLIC from the standpoint of stakeholders and design team members from various disciplinary fields. From this study, some extensions and revisions to the model are being proposed, such as temporal dependencies, users' roles, and interaction breakdowns that are characteristic of different interaction styles. In addition, we want to explore what kind of design decisions are facilitated by our approach, and what are the impacts (good and bad) in the design processes from the standpoint of the involved disciplines.

ACKNOWLEDGEMENTS

Simone Barbosa and Maira de Paula would like to thank CNPq for ongoing support to their research. The authors are also grateful for their colleagues at the Semiotic Engineering Research Group at PUC-Rio for interesting discussions about many of the issues described in this paper.

REFERENCES

- [1] Alexander, C. *The Timeless Way of Building*. Oxford University Press. New York, NY. 1979.
- [2] Alexander, C., Ishikawa, S., Silverstein, M. *A Pattern Language*. Oxford University Press. New York, NY. 1977.

- [3] Barbosa, S.D.J., de Souza, C.S., de Paula, M.G., Silveira, M.S. Modelo de Interação como Ponte entre o Modelo de Tarefas e a Especificação da Interface (in Portuguese). Proceedings of V Simpósio Brasileiro sobre Fatores Humanos em Sistemas Computacionais, IHC 2002. Fortaleza, CE. 2002.
- [4] Barbosa, S.D.J.; Paula, M.G. “Designing and Evaluating Interaction as Conversation: a Modeling Language based on Semiotic Engineering” In J.Jorge; N. J. Nunes; J. Falcão e Cunha (eds.) *Interactive Systems Design, Specification, and Verification – 10th International Workshop, DSV-IS 2003*, Funchal, Madeira Island, Portugal, Junho 11-13, 2003, Revised Papers Series: Lecture Notes in Computer Science, Vol. 2844 2003, XIII, 429 p. ISBN: 3-540-20159-9. pp. 16–33.
- [5] Berry, D.M., Daudjee, K., Dong, J., Nelson, M.A., and Nelson, T., “User’s Manual as a Requirements Specification”, Technical Report, CS 2001-17, University of Waterloo, Waterloo, ON, Canada, May, 2001. Available online at http://se.uwaterloo.ca/~dberry/FTP_SITE/tech.report/users.man.pdf.
- [6] Borchers, J. *A Pattern Approach to Interaction Design*. West Sussex, England: John Wiley & Sons.2001.
- [7] Brooks, F. P., *The Mythical Man-Month*. Addison-Wesley, Reading, MA, 1975 (anniversary edition, 1995).
- [8] Carroll, J. M. (ed) *Scenario-based design: envisioning work and technology in system development*, New York, Wiley. 1995.
- [9] de Souza, C. S., *The Semiotic Engineering of Human-Computer Interaction*, The MIT Press, in press.
- [10] de Souza, C.S. The Semiotic Engineering of User Interface Languages. *International Journal of Man-Machine Studies*, 39, 753-773. 1993.
- [11] Gamma, E., Helm, R., Johnson, R., Vlissides, J. *Design Patterns*. Addison-Wesley. 1995.
- [12] Granlund, A., Lafrenière, D., Carr, D.A. A Pattern-Supported Approach to the User Interface Design Process. Proceedings of HCI International 2001. New Orleans, USA. 2001.
- [13] Kammersgaard, J., “Four different perspectives on Human-Computer Interaction”, *International Journal of Man-Machine Studies* 28:343-362, Academic Press, 1988.
- [14] Luyten, K.; Clerckx, T.; Coninx, K.; Vanderdonck, J. “Derivation of a Dialog Model from a Task Model by Activity Chain Extraction”. In J.Jorge; N. J. Nunes; J. Falcão e Cunha (eds.) *Interactive Systems Design, Specification, and Verification – 10th International Workshop, DSV-IS 2003*, Funchal, Madeira Island, Portugal, Junho 11-13, 2003, Revised Papers Series: Lecture Notes in Computer Science, Vol. 2844 2003, XIII, 429 p. ISBN: 3-540-20159-9. pp. 203–217.
- [15] Mullet, K., and Sano, D., *Designing Visual Interfaces: Communication-Oriented Techniques*, SunSoft Press, Mountain View, CA, 1995.
- [16] Norman, D. e Draper, S. (eds., 1986) *User Centered System Design*. Hillsdale, NJ. Lawrence Erlbaum.
- [17] Paula, M.G., Barbosa, S.D.J., and Lucena, C.J.P. “Relating Human-Computer Interaction and Software Engineering Concerns: Towards Extending UML Through an Interaction Modeling Language”, in *Closing the Gaps: Software Engineering and Human-Computer Interaction*, Workshop at Interact 2003, Zurich, Switzerland, 2003, pp.40-46.
- [18] Paula, M.G., *Designing the Human-Computer Interaction Based on Semiotic Engineering Models: Building an Interaction Model* (in Portuguese), Master dissertation, Informatics Department, Pontifícia Universidade Católica do Rio de Janeiro, 2003.
- [19] Preece, J., Rogers, Y., and Sharp, H., *Interaction design: beyond human-computer interaction*, John Wiley & Sons, New York, NY, 2002.
- [20] Rosson, M.B., and Carroll, J. M. *Usability Engineering: Scenario-Based Development of Human Computer Interaction*, Morgan Kaufmann, 2001.
- [21] Schön, D. *The Reflective Practitioner: How Professionals Think in Action*. New York: Basic Books. 1983.
- [22] Silva, E.J. Grupos Interdisciplinares de Pesquisa: Benefícios, Problemas e Caminhos de Solução. Workshop sobre Interdisciplinaridade, CLIHC 2003. Rio de Janeiro, agosto de 2003. Available online at <http://www.dimap.ufrn.br/~jair/wiihc/atas.pdf>, last accessed in May 2004.
- [23] Silva, P.P. “Object Modelling of Interactive Systems: The UMLi Approach”. PhD’s thesis, Department of Computer Science, University of Manchester, United Kingdom. 2002.
- [24] Traetteberg, H. “Dialog Modelling with Interactors and UML Statecharts – A Hybrid Approach”. In J.Jorge; N. J. Nunes; J. Falcão e Cunha (eds.) *Interactive Systems Design, Specification, and Verification – 10th International Workshop, DSV-IS 2003*, Funchal, Madeira Island, Portugal, Junho 11-13, 2003, Revised Papers Series: Lecture Notes in Computer Science, Vol. 2844 2003, XIII, 429 p. ISBN: 3-540-20159-9. pp. 346–361.
- [25] Van Welie, M. Patterns for Designers? Position Paper in CHI 2002 Workshop on “Patterns in Practice”. Available online at <http://www.welie.com/patterns/chi2002-workshop/index.html>
- [26] Winograd, T. and Flores, F., *Understanding Computers and Cognition: A New Foundation for Design*, Addison-Wesley, Reading, MA. 1986.

Artigos Resumidos

Avaliação de Acessibilidade na *Web* com a Participação do Usuário – um Estudo de Caso

Amanda Meincke Melo¹

M. Cecília C. Baranauskas¹

Fabiana Fator Gouvêa Bonilha²

1. Instituto de Computação (IC), Universidade Estadual de Campinas (Unicamp)
Caixa Postal 6176, CEP: 13083-970, Campinas, SP, Brasil
+55 19 3788-5870
{amanda.melo, cecilia}@ic.unicamp.br

2. Programa de Pós-Graduação em Música, Instituto de Artes (IA), Unicamp
Caixa Postal 6159, CEP: 13083-970, Campinas, SP, Brasil
+55 19 3788-7195
fbonilha@iar.unicamp.br

RESUMO

A inclusão social e educacional de pessoas com deficiência demanda a realização de pesquisas que a torne viável e efetiva. O presente artigo é fruto da convergência entre o referencial teórico/metodológico de uma pesquisa em Interação Humano-Computador (IHC) e a experiência concreta de uma mestrandia, deficiente visual, enquanto usuária de sistemas *Web*. A partir da realização de tarefas, que envolveram o acesso a páginas hospedadas na rede da Unicamp, buscou-se identificar as estratégias adotadas por essa usuária, bem como as dificuldades enfrentadas. A análise preliminar de acessibilidade dessas páginas aponta para a importância do reconhecimento das necessidades e especificidades de seus diferentes usuários, entre os quais pessoas com deficiência visual.

Palavras-chave

Inclusão na Universidade, Acessibilidade na *Web*, Avaliação com o Usuário.

INTRODUÇÃO

O tema inclusão social e educacional de pessoas com deficiência tem gerado inúmeras discussões, das quais emerge uma pluralidade de concepções e abordagens. Nesse contexto, as pessoas com deficiência têm o papel de se posicionarem diante desse tema, de maneira crítica e pró-ativa.

Assim, enquanto agentes e protagonistas de tal processo, devem propor alternativas e apontar caminhos para que essa inclusão seja possível e para que ela de fato ocorra em uma dimensão que transcenda o nível do discurso e alcance a prática. Além disso, também é importante o desenvolvimento de pesquisas que abordem temas referentes às demandas das pessoas com deficiências.

Nessa perspectiva, está em desenvolvimento, na Unicamp, um projeto apoiado pelo PROESP/CAPES, que tem por objetivo investigar e implementar mecanismos facilitadores da inclusão das pessoas com deficiência no Ensino Superior. Uma de suas linhas de ação está relacionada à acessibilidade na *Web*, o que inclui também a investigação da acessibilidade às pessoas com deficiência visual. Assim, são investigadas as estratégias adotadas pelos cegos no uso de sistemas *Web*, bem como as dificuldades encontradas na interação com esses sistemas.

O presente artigo tem por objetivo discutir resultados de uma análise preliminar sobre acessibilidade de páginas hospedadas na rede da Unicamp, a partir de uma avaliação realizada com uma usuária cega, aluna de mestrado dessa instituição. Na seção a seguir apresentamos o referencial teórico e metodológico do trabalho. Então, relatamos uma experiência de avaliação de acessibilidade com a participação de uma usuária. Finalmente, apresentamos as considerações finais deste trabalho.

REFERENCIAL TEÓRICO E METODOLÓGICO

A acessibilidade da Internet caracteriza-se pela flexibilidade da informação e interação relativa ao respectivo suporte de apresentação. Essa flexibilidade deve permitir a sua utilização por pessoas com necessidades especiais, bem como a utilização em diferentes ambientes e situações, e através de vários equipamentos ou navegadores [1].

A avaliação da usabilidade de um sistema computacional passa pela verificação de sua acessibilidade relacionada ao contexto de uso, às tarefas que apóia, bem como às necessidades e às preferências dos diferentes usuários prospectivos. Um *software* ou página *Web* que não é acessível a uma determinada pessoa tão pouco pode ser considerado eficaz, eficiente ou mesmo agradável a essa pessoa [2].

No contexto da *Web*, métodos para avaliação de acessibilidade têm sido propostos [2,5] e a complementaridade desses métodos possibilita a

VI Simpósio sobre Fatores Humanos em Sistemas Computacionais — Mediando e Transformando o Cotidiano. Hotel Bourbon, Curitiba, 17 a 20 de outubro de 2004. UFPR, CEIHC—SBC.

verificação da acessibilidade *Web* sob diferentes perspectivas. Alguns desses métodos são mencionados a seguir:

- **Uso de navegadores gráficos e textuais:** permite investigar questões relacionadas à interpretação de páginas *Web* por meio de diversas configurações de acesso.
- **Validação automática da linguagem de marcação:** identifica problemas relacionados à sintaxe HTML e CSS, por exemplo; atividade difícil de ser realizada depois que as páginas já estão codificadas.
- **Verificação de acessibilidade por ferramentas semi-automáticas:** evidencia erros no design de páginas HTML que prejudicam sua acessibilidade, além de lembrarem verificações importantes que devem ser realizadas manualmente.
- **Avaliação com usuários com diferentes habilidades e/ou deficiências:** possibilita a observação das estratégias de interação construídas pelos diferentes usuários na realização de tarefas típicas, em contextos diversificados e com o uso de tecnologias assistivas — ex. leitores de tela, ampliadores de tela, etc —, bem como a identificação das dificuldades que enfrentam.

A Iniciativa de Acessibilidade da *Web* (WAI) recomenda a realização de testes de usabilidade, que considerem características relacionadas à acessibilidade, como método de avaliação de acessibilidade com usuários (W3C, 2002). No entanto, a literatura não apresenta métodos específicos que contemplem usuários com deficiências. Neste artigo, o estudo de caso realizado baseia-se no método de observação participativa, que possibilita uma interação social intensa entre pesquisador e sujeitos [4].

AVALIAÇÃO DE ACESSIBILIDADE WEB: UM ESTUDO DE CASO COM A PARTICIPAÇÃO DO USUÁRIO

Nesta seção, relatamos uma experiência de avaliação de acessibilidade de páginas hospedadas na rede da Unicamp, realizada com a participação de uma de suas usuárias, em 05 de maio de 2004, no Laboratório de Acessibilidade (LAB) da Biblioteca Central (BC) da Unicamp.

Essa experiência foi motivada pela convergência entre os problemas de acessibilidade enfrentados por uma aluna do Programa de Mestrado em Música, no acesso a páginas hospedadas no portal da Unicamp que a impossibilitam de realizar tarefas por meio de sistemas de informação *Web* — ex. uso do *Webmail*, realização de matrícula e acesso ao histórico escolar por meio do *site* da Diretoria Acadêmica (DAC) — e uma pesquisa de doutorado realizada no Programa de Pós-graduação do Instituto de Computação, que envolve questões relacionadas ao *Design Universal* e Acessibilidade de sistemas *Web*.

Assim, essa avaliação tem como propósitos complementar uma verificação preliminar de acessibilidade do portal da Unicamp, identificar estratégias de acesso adotadas pela usuária colaboradora,

bem como suas dificuldades no uso de páginas da Unicamp.

Sujeito e Método

A usuária em questão é aluna regular de pós-graduação da Unicamp, co-autora deste artigo. Apresenta deficiência visual total e congênita. Considera o uso do computador imprescindível, uma vez que os avanços tecnológicos lhe possibilitaram um acesso à informação que antes era impensável. É usuária experiente de sistemas computacionais, bem como usuária habitual do sistema *Web* da Unicamp.

Utiliza o computador em casa e no LAB/BC, cerca de seis horas/dia. Segundo ela, o LAB/BC apresenta melhores condições de ergonomia por oferecer um espaço mais amplo onde os equipamentos de informática estão dispostos em espaços mais adequados.

O método adotado para avaliação trata-se da observação participativa, no qual foram definidas — pela examinadora — quatro tarefas que lembram atividades corriqueiras para os alunos da Unicamp, realizadas pela usuária. Essas tarefas pressupunham o acesso ao portal da Unicamp, bem como o uso dos sistemas de informação *Web* voltados à comunidade em geral e aos alunos dessa instituição de ensino superior.

Durante a realização das tarefas, as intervenções foram realizadas mediante a solicitação da usuária e/ou para o esclarecimento de dúvidas da examinadora sobre as estratégias de interação adotadas. Ao final de cada tarefa, estabeleceu-se um diálogo a respeito da interação com o sistema. Essa atividade foi registrada em fita K7, por meio de um gravador de áudio.

Estado Inicial do Sistema no Início da Avaliação

O Sistema Operacional Windows XP já estava ligado e a usuária teve acesso ao navegador Microsoft Internet Explorer 6.0 (IE), por meio do “Menu Iniciar → Programas”, com o auxílio do leitor de telas *Jaws* para Windows [3]. Ao ser ativado, o navegador apresentou a página principal do Portal da Unicamp como página de início.

Tarefa 1

Quantos cursos de pós-graduação a Unicamp oferece?

A resposta para essa tarefa está a um clique da página principal do Portal da Unicamp (<http://www.unicamp.br>), a partir do *link* “Pós-Graduação”: 63 cursos. Foi rapidamente completada pela usuária, devido a sua familiaridade com a estrutura da página principal do portal da Unicamp. Estratégia adotada: procura pela palavra-chave “graduação”, na página principal da Unicamp, com a ferramenta Localizar (Ctrl+F) do navegador. Nesse caso, a usuária sabia de antemão que o *link* “Pós-Graduação” seria a próxima informação alcançada, após a localização da palavra-chave. A resposta para a tarefa foi encontrada de forma exploratória na página com informações sobre a pós-graduação da Unicamp. Duração: 1 min.

Tarefa 2

Qual o último dia para Abandono Justificado de Matrícula em disciplinas oferecidas na primeira metade do primeiro período letivo de 2004?

A resposta a essa tarefa exige o acesso ao calendário dos cursos de pós-graduação/2004 e está a dois cliques da página principal do *site* da DAC (<http://www.unicamp.br/prg/dac>): 22 a 28/04. Foi realizada com bastante dificuldade devido à falta de texto alternativo que ajudasse a contextualizar as opções oferecidas aos alunos de graduação, pós-graduação e extensão. Várias estratégias foram adotadas para sua realização: busca pelas palavras-chave “calend”, “pós”, “matrícula”, “aband”, “catálogos de” na página da DAC, acesso a *links* por meio da tecla TAB, uso do mecanismo de busca oferecido no portal da Unicamp. Duração: 13 min.

Tarefa 3

Qual o número de créditos integralizados em seu histórico escolar?

O desenrolar dessa tarefa foi influenciado pelo diálogo estabelecido antes de sua realização, no qual a usuária investigou e questionou sobre as opções de acesso aos serviços oferecidos no *site* da DAC (<http://www.unicamp.br/prg/dac/daconline.php>). Essas opções são oferecidas por meio de um mapa de imagens (Figura 1), no qual faltam textos alternativos a elas.



Figura 1. Acesso aos Serviços no *site* da DAC

A resposta a essa tarefa requeria o acesso ao histórico escolar, a partir do acesso aos serviços da DAC oferecido aos alunos (<http://www.unicamp.br/prg/dac/alunos.php>), disponibilizado também por meio de um mapa de imagens sem texto alternativo para suas opções (Figura 2).



Figura 2. Acesso aos Serviços, oferecidos aos alunos, no *site* da DAC

A resolução da tarefa, então, teve início após a identificação dessas barreiras de acesso oferecidas pelos mapas de imagens, que não ofereciam textos alternativos as suas opções.

O acesso, propriamente dito, aos serviços oferecidos aos alunos exige autenticação pela informação de *username* e senha. Na primeira tentativa de autenticação, a usuária informou errado seu *username*. Apesar de o sistema oferecer *feedback* sobre esse erro, ele foi realizado de forma inadequado à usuária, uma vez que para ter acesso à informação de erro ela teria que percorrer quase toda a interface — com o leitor de telas. A usuária conseguiu inferir que algo havia ocorrido de errado, pois o sistema a remeteu para o formulário de autenticação novamente. Finalmente, após realizada a autenticação, a usuária explorou algumas opções de serviços oferecidas aos alunos e completou a tarefa, pelo acesso e leitura de seu histórico escolar. Duração: 6 minutos.

Tarefa 4

Quais os serviços de apoio oferecidos aos alunos da Unicamp?

A resposta para essa tarefa está a um clique da página principal do Portal da Unicamp (<http://www.unicamp.br>), a partir do *link* “Alunos”: matrícula de graduação e pós-graduação, Webmail da DAC, bolsas, cardápio do RU, etc. No entanto, ela foi realizada a partir do acesso à página do Serviço de Apoio ao Estudante (SAE) (<http://www.sae.unicamp.br/>). Estratégia inicial: procura pela palavra-chave “apoio”, na página principal da Unicamp, com a ferramenta Localizar do navegador. Com essa busca, a usuária teve acesso ao *link* “Apoio Acadêmico”, o qual acessou, mas não obteve êxito em sua busca. Estratégia 2: acessar o *site* do SAE e explorá-lo. Duração: 5 min.

Resultados Preliminares

Ilustramos, nesta subseção, como a interação humano-computador pode tornar-se excessivamente complicada quando não há preocupação com a acessibilidade. Por limite de espaço, destacamos neste artigo um dos resultados observados das tarefas.

Na tarefa 2, a usuária adotou diferentes estratégias para conseguir realizá-la. Em um primeiro momento, a partir da página inicial do *site* da DAC, buscou pela palavra-chave “calend”, sendo remetida ao *link* “Calendário”

referente aos calendários da graduação, o qual explorou logo em seguida. Ao perceber que a página acessada não estava relacionada à pós-graduação, retornou ao *site* da DAC e procurou pela palavra-chave “pós”, sem obter sucesso. Como o conteúdo lido a seguir, com o leitor de telas, foi o *link* “Calendário”, o acessou, inferindo que seria referente à pós-graduação. No entanto, logo percebeu que se tratava novamente do *link* referente aos calendários da graduação. Outras buscas com o recurso Localizar do IE foram realizadas, sem êxito.

Uma outra estratégia adotada foi o uso do mecanismo de busca oferecido no portal da Unicamp. A palavra-chave de busca, nesse caso, foi “abandono justificado”, que retornou vários resultados. A primeira tentativa foi a de explorar os *links*, da primeira página de resultados da busca, seqüencialmente, de cima para baixo. Ao perceber que os resultados, que remetiam aos calendários, não estavam na ordem cronológica mais recente ao mais antigo, passou a explorá-los seqüencialmente, mas de baixo para cima. Não havia, no resultado da busca, informações que ajudassem a usuária identificar o ano ao qual os calendários estavam associados, o que ela conseguia perceber apenas após acessá-los.

Por não obter sucesso na realização da tarefa com o uso desse mecanismo de busca, a usuária acessou a página da DAC mais uma vez, entrou no *link* “Calendário” referente à graduação, retornou à página da DAC, leu os *links* subsequentes e, ao perceber, um outro *link* “Calendário”, também o acessou. A partir de então, uma vez que o *link* era relativo aos calendários da pós-graduação, deu prosseguimento à realização da tarefa.

Todo esse percurso para acessar o calendário oferecido aos alunos de pós-graduação evidenciou que uma boa estrutura para alguém que faz uso da referência visual não implica em uma boa estrutura para quem está desprovido dela. O *site* da DAC, explorado nessa tarefa, oferece um menu com as opções referentes à graduação, pós-graduação e extensão. Enquanto é possível identificar facilmente a que se refere cada opção desse menu quando as imagens são exibidas (Figura 3.a), o mesmo não acontece com a exibição de imagens desabilitadas no navegador (Figura 3.b), pois não há texto alternativo que identifique as imagens utilizadas para separar/categorizar essas opções. Na ausência do texto alternativo, a usuária ficou sem as referências necessárias para identificar o *link* “Calendário” da pós-graduação.

Nesse caso, o mais adequado teria sido oferecer a informação textual veiculada em imagem, por meio de texto, o que facilitaria a busca por palavra-chave. Oferecer teclas de atalho aos diferentes menus de opções também melhoraria o acesso a eles.

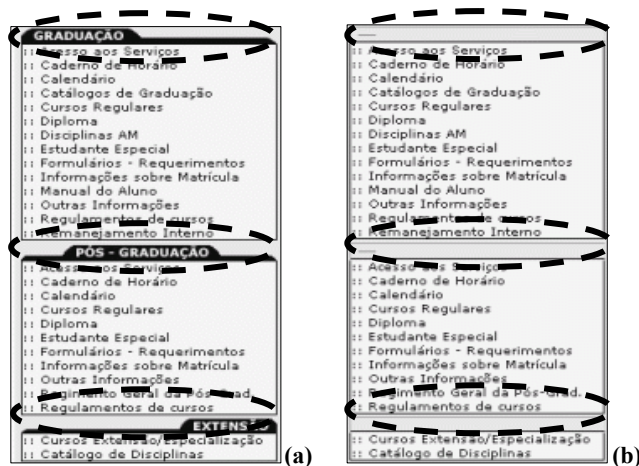


Figura 3. Menu de opções do *site* da DAC (a) e o mesmo menu apresentado com a exibição de imagens desativada (b)

AGRADECIMENTOS

Ao PROESP/CAPEs, ao CNPq e à FAPESP.

CONCLUSÕES

Neste artigo apresentamos subsídios para entender as estratégias de interação utilizadas por uma usuária deficiente visual e identificamos dificuldades na interação com o sistema *Web* da instituição de ensino que frequenta. Resultados dessa avaliação complementam uma verificação preliminar de acessibilidade de páginas da Unicamp, em curso. A partir de resultados desse tipo de avaliação, será possível propor soluções relacionadas à acessibilidade, como parte da construção da inclusão da pessoa com deficiência na Unicamp. Este trabalho contribui na direção do envolvimento de pessoas com deficiência em práticas participativas no *design* e avaliação de sistemas.

REFERÊNCIAS

1. Centro de Engenharia de Reabilitação em Tecnologias de Informação e Comunicação.
<http://www.acessibilidade.net/web/>
2. Dias, C. Usabilidade na Web: Criando Portais mais Acessíveis”, AltaBooks, Rio de Janeiro, Brasil, 2003.
3. Freedom Scientific Home Page
<http://www.freedomscientific.com/>
4. Simoni, C. A. C; Baranauskas, M. C. C. Pesquisa Qualitativa em Sistemas de Informação, Relatório Técnico, IC/Unicamp, Campinas, Brasil, 2003.
5. Evaluating Web Sites for Accessibility.
<http://www.w3.org/WAI/eval>

Desafios de Avaliação de Interfaces de Ambientes Educacionais – Um Estudo de Caso

Raquel Oliveira Prates

Departamento de Informática
IME – UERJ
R. São Francisco Xavier, 524 –
6o. andar, Maracanã, RJ
raquel@ime.uerj.br

Carla Faria Leitão

Departamento de Psicologia,
SERG/DI - PUC-Rio
R. Marquês de São Vicente, 225
Gávea, RJ
cfaria@inf.puc-rio.br

Rosa Maria V. de Figueiredo

Departamento de Matemática
Aplicada – IME – UERJ
R. São Francisco Xavier, 524 –
6o. andar, Maracanã, RJ
rosa@ime.uerj.br

RESUMO

Ambientes educacionais trazem novos desafios para o design e avaliação de interfaces [2, 10]. Neste artigo levantamos questões sobre a avaliação de interfaces educacionais, a partir de um estudo de caso. Este estudo compreendeu a avaliação formativa da interface de um sistema de apoio ao aprendizado utilizando o método de avaliação de comunicabilidade [4, 5].

Palavras Chave

Avaliação, comunicabilidade, interfaces educacionais.

INTRODUÇÃO

Algumas teorias, métodos e técnicas já estão bem consolidados para a área de IHC [7]. No entanto, alguns domínios têm especificidades que podem tornar necessária a adaptação ou mesmo a revisão destes métodos e técnicas e apresentam novos desafios para o design e avaliação de suas interfaces. Este parece ser o caso do domínio educacional, no qual, por exemplo, uma interface pode deliberadamente permitir (ou até mesmo induzir) o erro do usuário para propiciar ou consolidar seu aprendizado [2]. Neste contexto a máxima de IHC de que a interface deve sempre prevenir erros de seus usuários [3, 8] é conscientemente violada.

À medida que ambientes educacionais vêm sendo, cada vez mais, integrados às práticas de ensino e aprendizado justifica-se investir na pesquisa sobre como tratar as particularidades do domínio educacional. Tal investimento já começa a se fazer presente na área de IHC por meio, por exemplo, de propostas de métodos de avaliação preditiva capazes de auxiliar o professor na escolha de software adequados a seus objetivos didáticos [10].

Neste artigo, adotamos um foco distinto ao discutir a avaliação formativa da interface de ambientes educacionais com o objetivo de melhorar sua qualidade. Para isto, relatamos a nossa experiência com a avaliação formativa de um ambiente de apoio ao aprendizado utilizando o método de avaliação de comunicabilidade [4, 5], e levantamos algumas questões específicas encontradas nesta avaliação. Estas questões são relevantes, uma vez que ilustram os desafios de design de interfaces no contexto educacional e criam a demanda

para uma investigação mais profunda sobre a necessidade de adaptação de métodos de avaliação existentes para este domínio ou da proposta de novos métodos específicos.

Na próxima seção apresentamos o sistema Ampliar [6] utilizado neste estudo de caso. A seguir apresentamos o método de avaliação de comunicabilidade e o teste da interface do Ampliar feito usando este método. Na seção seguinte descrevemos trechos dos testes e levantamos algumas das questões que surgiram durante este estudo em relação ao método utilizado. Finalmente, nas conclusões discutimos a relevância dos resultados para a pesquisa de IHC voltada ao domínio educacional.

SISTEMA AMPLIAR

O sistema Ampliar [6] é um sistema de apoio ao aprendizado desenvolvido no Instituto de Matemática e Estatística da Universidade do Estado do Rio de Janeiro (UERJ) para apoiar o ensino do método simplex no curso de Programação Linear. Na UERJ, o método simplex normalmente é ensinado aos alunos em 2 ou 3 aulas, nas quais o professor explica o método, exemplifica o seu uso e passa exercícios para os alunos. No entanto, a resolução de problemas pelo simplex implica a resolução de um grande número de operações algébricas. Estas operações consomem tempo, e requerem uma grande atenção, pois erros podem ser propagados e ter um alto custo para serem corrigidos. Assim, o aluno muitas vezes se sente desmotivado a fazer exercícios para fixar o conteúdo aprendido ou pode priorizar as operações em detrimento do entendimento do método. Além disso, os exercícios usualmente são feitos em horário extra-classe pelos alunos, o que dificulta o esclarecimento de dúvidas com o professor.

Com o objetivo de auxiliar os alunos na resolução de problemas pelo simplex e possibilitar um melhor entendimento do método é que o sistema Ampliar foi proposto e desenvolvido. Para que se conseguisse atingir no Ampliar os objetivos educacionais e de design desejados, ele foi desenvolvido dentro da abordagem da teoria da Engenharia Semiótica [1].

No design do Ampliar buscou-se dar espaço para os alunos tomarem as decisões sobre o passo a ser executado a cada momento. A cada passo, o aluno é quem deve solicitar ao sistema os cálculos das operações necessárias, fazendo assim com que ele tenha que saber os cálculos a serem executados, mas não cometa erros ao executá-los. O sistema apresenta sempre todas as opções ao aluno, mesmo aquelas que não estão corretas, para que o aluno decida e pense sobre o método. Quando o aluno comete um erro o sistema não lhe oferece de imediato uma

VI Simpósio sobre Fatores Humanos em Sistemas Computacionais — Mediando e Transformando o Cotidiano. Hotel Bourbon, Curitiba, 17 a 20 de outubro de 2004. UFPR, CEIHC—SBC.

explicação completa sobre o que ele deveria fazer, mas vai fornecendo gradativamente mais informações sobre o erro que está sendo cometido.

O sistema pretende não apenas permitir ao aluno automatizar a execução do método, mas também possibilitar seu entendimento. Assim, o sistema permite ao aluno tirar suas dúvidas sobre o método durante a execução do problema. Para isso, a cada passo são oferecidas ao aluno diversas perguntas e respostas organizadas nas categorias: “O que é?”, “Como?”, “E agora?”, “Quando?” e “Por que?”. Para cada categoria um conjunto de perguntas relevantes à etapa do método que ele se encontra são apresentadas, permitindo assim ao aluno um melhor entendimento do que significam os elementos envolvidos no método, como executar cada etapa ou sub-etapa, o que fazer a cada momento, quando executar cada cálculo e por que uma etapa ou cálculo é necessário ou relevante.

DESCRIÇÃO DO TESTE COM USUÁRIOS

Os testes de avaliação foram feitos no laboratório do SERG [9], utilizando o método de avaliação de comunicabilidade e contou com a presença de 7 alunos do curso de Programação Linear da UERJ.

Método de Avaliação de Comunicabilidade

O método de avaliação de comunicabilidade [4, 5] tem como objetivo avaliar uma interface com relação à qualidade da comunicação do designer para os usuários. Para isto, este método simula a comunicação do usuário para o designer sobre os problemas de interação que o usuário teve ao utilizar a interface. Isto é feito através de um pequeno conjunto de expressões que o usuário potencialmente pode usar para se exprimir em uma situação onde acontece uma ruptura na sua comunicação com o sistema. A aplicação do método pode ser dividida em duas etapas: a coleta de dados e a sua análise.

A coleta de dados é feita através de um teste em laboratório, durante o qual se grava um filme da interação, usando para isto um software de captura das ações do usuário. No caso do Ampliar, o filme da interação foi complementado com anotações dos observadores durante o teste, e uma entrevista pós-teste com cada aluno.

Analisa-se então o filme da interação gerado, utilizando as anotações para facilitar o entendimento do contexto e possíveis linhas de raciocínio dos participantes do teste. A análise é feita em 3 passos: (1) etiquetagem na qual se atribui a cada momento de ruptura da interação observado no filme a expressão apropriada (a partir do conjunto proposto). Por exemplo, quando o usuário percorre opções do menu à procura de uma determinada função, então se associaria a esta porção da interação a expressão “Cadê?”; (2) a interpretação das expressões associadas através da sua associação a ontologias de problemas de interação ou diretrizes de design; e (3) um perfil semiótico, na qual, a partir dos passos anteriores e com base no quadro teórico da Engenharia Semiótica, tenta-se reconstruir a meta-mensagem do designer para o usuário através da interface.

Perfil dos Alunos

Participaram da avaliação do Sistema Ampliar 5 homens

e 2 mulheres, alunos do 5º ao 8º período dos Cursos de Matemática, de Informática ou de Engenharia Mecânica da UERJ. Em sua maioria, os alunos participantes utilizam o computador diariamente, em média por 30 a 40 horas semanais, acessando-o tanto do trabalho como de casa.

Todos os participantes pertenciam a uma mesma turma da disciplina Programação Linear. Por ocasião da avaliação, os alunos haviam assistido às aulas sobre o método simplex e – idealmente – deveriam estar executando uma lista de exercícios para consolidar seu aprendizado. Dado que ainda estariam na etapa de consolidação de conhecimentos sobre o simplex, os alunos desta turma mostraram-se especialmente representativos do perfil de usuários-alvo para o qual o sistema Ampliar foi desenvolvido. Os alunos recrutados voluntariamente se interessaram em participar da avaliação de comunicabilidade do Ampliar.

Teste Aplicado

Durante a aplicação do teste foi apresentado a cada aluno uma breve descrição do sistema Ampliar. Nesta descrição foi explicado o objetivo do Ampliar, ressaltando a possibilidade de se fazer perguntas sobre o conteúdo ao próprio sistema e a apresentação gradativa da explicação completa no caso de erro. Além disso, foram apresentadas algumas particularidades da interface, como a necessidade de se dar *scroll* para se visualizar os cálculos efetuados e a representação da notação de alguns elementos na interface.

Após o aluno ter confirmado o entendimento do texto explicativo, ele recebeu então um exercício para resolver utilizando o Ampliar. O exercício passado era pequeno, de solução simples e semelhante a um exercício apresentado em sala de aula.

Os testes foram aplicados por uma especialista em IHC, e observados por uma professora de Programação Linear.

DESAFIOS NA AVALIAÇÃO

A aplicação do teste com os alunos e a coleta dos dados pôde ser considerada habitual, no sentido de que não exigiu cuidados diferentes daqueles que se têm na avaliação da comunicabilidade da interface de qualquer outro software. No entanto, ao iniciarmos a análise, nos deparamos com algumas situações de ruptura de interação que deveriam ter um tratamento distinto do proposto no método original. Vale ressaltar que a análise foi feita por 2 especialistas no método de avaliação de comunicabilidade (EC) e 1 especialista no domínio (ED).

A partir da análise, classificamos estas situações em 3 categorias de acordo com o tratamento diferenciado no método que consideramos necessário: (1) ambigüidade de sintoma – a expressão associada aos sintomas observados não descreve a ruptura ocorrida; (2) novos sintomas – ações do usuário que normalmente não eram um sintoma de ruptura, deveriam neste contexto ser encaradas como tal; (3) alterações na interpretação da ruptura observada. Nesta seção apresentamos para cada uma das categorias identificadas as descrições do que foi observado no teste, as questões envolvidas na análise e discutimos os pontos sobre a avaliação da interfaces educacionais que elas levantam.

Ambigüidade de Sintoma

Observação no Teste

Durante o teste do Participante 7 (P7) verificamos que em um determinado momento, após executar alguns cálculos, seu cursor é colocado sobre a área em que estes cálculos são mostrados e fica por um tempo parado ou se alternando entre dois trechos dos cálculos apresentados.

Questão

Normalmente no método de avaliação de comunicabilidade este comportamento durante a interação seria etiquetado com um “E agora?” indicando que o usuário não sabia como prosseguir. No entanto, neste ambiente como o cursor estava parado sobre uma a área de cálculos as avaliadoras EC não souberam identificar se o aluno não sabia como prosseguir, o que seria uma ruptura de interação, ou simplesmente se estava raciocinando sobre o que o sistema acabara de lhe apresentar. Na avaliação da ED fica claro que era o 2o. caso, uma vez que ela identificou que os trechos de cálculo sobre os quais o cursor se alternava eram justamente aqueles necessários naquele momento para se decidir o próximo passo na resolução do problema.

Discussão

Na avaliação do Ampliar ficou claro que eram necessárias a presença de um especialista do método de avaliação de comunicabilidade e outro do domínio para fazer uma análise precisa da interação. Assim, o ponto levantado é qual o conhecimento necessário para se fazer a avaliação de um teste com usuários de um ambiente educacional. São necessários especialistas distintos ou poderia ser dado a um deles subsídios para tirar este tipo de ambigüidades (ou algumas delas)? Que subsídios seriam estes? Qual seria o custo x benefício de se ter todos os especialistas necessários ou de se abrir mão de algum deles?

Novos Sintomas

Observação 1 no Teste

O Participante 3 (P3) em um determinado ponto da resolução do problema não sabe como continuar. Apesar de fazer algumas perguntas ao sistema, ele continua sem entender o que deve fazer para tomar as decisões necessárias naquele momento. Após um tempo, o usuário simplesmente desiste de entender e continua a resolução do problema na base da tentativa e erro.

Questão 1

Neste caso, o que foi observado não caracteriza uma ruptura da comunicação sobre a interface, mas sim uma ruptura sobre a comunicação sobre o método simplex. No entanto, o método de avaliação de comunicabilidade não permite aos avaliadores o registro deste tipo de ruptura. Vale ressaltar que a indicação desta ruptura não é baseada em uma ação do usuário, mas em um conjunto de ações que incluem muitas vezes outras rupturas.

Observação 2 no Teste

Nos testes dos participantes P3 e P4, em alguns momentos em que tinham que tomar alguma decisão, sua primeira iniciativa era sempre fazer todos os cálculos disponíveis, mesmo aqueles que não eram necessários para sua decisão.

Questão 2

Neste caso as ações não representam uma ruptura na

comunicação sobre a interface. O aluno pode sempre que desejar (e achar necessário) fazer os cálculos. Até porque um dos cálculos disponibilizados não é necessário nunca para a resolução do problema, ele só é útil na verificação da melhoria na solução a cada iteração do método. No entanto, aqui existe uma falha da interface à medida que ela perde a oportunidade de deixar claro para o aluno os cálculos necessários ou sua utilidade na solução do problema. Além disso, ela ainda pode estar encorajando um entendimento errôneo de que todos os cálculos são sempre necessários.

Discussão

Estes dois exemplos apontam para a necessidade de se identificar outros tipos de quebras da comunicação designer-usuário que são referentes não às opções de interação, mas ao aprendizado associado a elas. Uma investigação mais profunda seria necessária para se tentar identificar se existe um conjunto adicional de expressões que descrevam estas rupturas na comunicação referente ao processo de ensino e aprendizado, ou se novas expressões surgem no contexto específico do objetivo educacional de uma determinada aplicação.

O primeiro exemplo levanta ainda uma outra consideração a ser feita sobre a aplicação do teste. Sabe-se que, muitas vezes, participantes de teste, embora avisados de que o sistema é quem está sendo testado, se sentem também avaliados. Em um contexto educacional, principalmente o do Ampliar, no qual o aluno-usuário deve ter um conhecimento mínimo para conseguir usá-lo, o teste pode explicitar a falta do conhecimento esperado do aluno. Mesmo que esta exposição não tenha nenhuma consequência prática para o aluno em termos do seu curso, ela pode ser bastante desconfortável para ele, principalmente se o professor do tópico estiver presente durante o teste. Assim, formas de lidar com isso devem ser uma preocupação permanente na preparação e execução de um teste de um sistema educacional com alunos.

Novas Interpretações

Observação no Teste

O Participante 5 (P5) recorre às perguntas a todo momento, para entender o que são os elementos do problema e o que fazer. Fica claro que o aluno tem tantas dúvidas, que de fato sua tentativa era de aprender o método, e não consolidá-lo. Isto fica ainda mais patente no momento em que o aluno interrompe o teste e se diz incapaz de resolver o problema.

Questão

No método de avaliação de comunicabilidade a falha em executar a tarefa e desistência do uso de um sistema é marcado com a expressão “Não dá.” Normalmente este é um problema bastante grave uma vez que indica que a solução do designer para o problema do usuário não o satisfaz. No caso de ambientes educacionais, um “Não dá” pode não ser um ponto negativo na sua avaliação, e pode ser até positivo. No caso do Ampliar, ao desistir o aluno disse: “Descobri que estou totalmente perdido.” Esta descoberta pode ter permitido ao aluno avaliar o seu nível de conhecimento sobre o Simplex contra o esperado, ou mesmo o de seus colegas e ter sido um estímulo para que ele o estudasse. Assim, pode-se dizer que o Ampliar atingiu seu objetivo ao comunicar

claramente ao aluno qual era o conhecimento esperado que ele tivesse.

Discussão

Este exemplo indica que pode ser necessário associar as expressões do método de avaliação de comunicabilidade a ontologias propostas ou adaptadas ao domínio educacional.

CONCLUSÃO

Neste trabalho apresentamos uma avaliação formativa de um sistema educacional utilizando-se o método de avaliação de comunicabilidade. A partir deste estudo de caso levantamos questões específicas ao uso deste método de avaliação para ambientes educacionais. Algumas destas questões parecem extrapolar o método de avaliação de comunicabilidade, e devem ser consideradas mesmo na utilização de outros métodos de avaliação.

Apesar dos desafios encontrados, a avaliação feita forneceu um feedback bastante produtivo para o design do Ampliar. Assim, a discussão apresentada tem por objetivo também fornecer a equipes de desenvolvimento de software educacional alguns parâmetros que lhes permita avaliar o custo x benefício de uma avaliação formativa utilizando o método de comunicabilidade, ou até mesmo tentar definir uma adaptação do método para seu contexto específico.

Este trabalho contribui também para a pesquisa do método de avaliação de comunicabilidade à medida que mostra que o método proposto pode demandar adaptações em domínios específicos, como no caso do educacional. Apenas com uma investigação mais detalhada poder-se-á discutir se uma proposta de adaptação para todo o domínio educacional seria possível e qual seria ela. Além disso, neste estudo de caso identificamos rupturas tanto na comunicação sobre as opções de interação, quanto sobre o conteúdo apresentado pela interface, e ainda sobre o conhecimento presumido do aluno. Para a avaliação formativa, em particular, seria interessante se poder identificar em qual destes aspectos desta comunicação ocorreu a quebra. Assim, seria interessante se refletir sobre se, e como, estes aspectos da comunicabilidade da mensagem do designer para o

usuário neste domínio têm repercussões na prática (e.g. no método) e até mesmo na teoria.

AGRADECIMENTOS

Agradecemos aos alunos da UERJ que participaram do teste, à UERJ pelo seu apoio à pesquisa do Ampliar e ao SERG pelo seu apoio à avaliação com usuários.

REFERÊNCIAS

1. de Souza, C.S. *The Semiotic Engineering of Human-Computer Interaction*. The MIT Press, Cambridge, MA, in press, 2004.
2. Lewis, C., Brand, C., Cherry, G., Rader, C. Adapting User Interface Design Methods to the Design of Educational Activities. *Proc. CHI 98*, USA, pp 18-23.
3. Nielsen, J. Heuristic Evaluation, *Usability Inspection Methods*. NY: John Wiley & Sons, 1994, 25-62.
4. Prates, R.O.; de Souza, C.S.; Barbosa, S.D.J.; “A Method for Evaluating the Communicability of User Interfaces.”. *Interactions* 7, 1. NY: ACM, 31-38, 2000.
5. Prates, R. O., Barbosa, S. D. J. Avaliação de Interfaces de Usuário - Conceitos e Métodos. JAI, SBC, 2003.
6. Prates, R. O.; Figueiredo, R. M. V. de; Bach, C. F. Um Modelo de Apoio ao Projeto de Design de Interfaces de Ambientes de Aprendizado. *Anais do WIE*, SBC, 2003. V.541- 552.
7. Preece, J.; Rogers, Y.; Sharp, E. *Interaction Design: Beyond HCI*, NY, John Wiley & Sons. 2002.
8. Shneiderman, B. *Designing the User Interface*, 3rd. Edition. Reading, MA: Addison Wesley, 1998.
9. SERG – Semiotic Engineering Research Group, <http://www.serg.inf.puc-rio.br/>
10. Squires, D.; Preece, J. “Predicting quality in educational software: Evaluating for learning, usability and the synergy between them”. *Interacting with Computers*, 11, 1999, 467–483.

Estratégias Cognitivas para Aumento da Qualidade do Hiperdocumento que Contém o Material Instrucional para EAD

Vânia Paula de Almeida

Júnia Coutinho A. Silva

Universidade Federal de São Carlos
Caixa Postal 676 CEP 13565-905 São Carlos/SP
{vânia,junia}@dc.ufscar.br

RESUMO

Acredita-se que para um processo de ensino e aprendizagem efetivo em Educação a Distância (EAD) se faz necessário apoiar o professor durante o desenvolvimento do material instrucional, auxiliando-o na estruturação desse material, estimulando-o a utilizar recursos de edição de hiperdocumentos, apropriados ao contexto computacional, considerando aspectos pedagógicos e computacionais para esse fim, que facilitem a construção do conhecimento do aluno usuário de tal material. O objetivo deste trabalho¹ é avaliar o uso de um conjunto de estratégias cognitivas no apoio à geração e estruturação de material instrucional para EAD, resultando em um hiperdocumento, através da avaliação de usabilidade da interface computacional que contém o material.

PALAVRAS-CHAVE

Estratégias cognitivas, EAD, qualidade, material instrucional.

INTRODUÇÃO

Qualidade é um fator determinante para o sucesso ou fracasso dos produtos no mercado extremamente competitivo e globalizado. No contexto computacional, a qualidade de software vem sendo cada vez mais estudada e discutida entre aqueles que estão preocupados e conscientes da necessidade de softwares bem projetados e bem construídos. Particularmente, pesquisadores interessados no uso dos computadores como ferramentas de apoio a educação têm apresentado trabalhos considerando aspectos de qualidade em softwares educacionais [10], [7], [13].

No contexto de EAD, essa preocupação deve ser ainda maior. As dificuldades inerentes da distância física entre professor e aluno aumentam ainda mais a necessidade de professores preparados para a edição de cursos bem planejados, ambientes computacionais com recursos adequados e fáceis de serem utilizados e alunos motivados a adquirirem conhecimento à distância. Considera-se aqui o curso como resultado da junção de um conjunto de ferramentas computacionais oferecidas pelo ambiente e escolhidas pelo professor (chat, e-mail, links interessantes, controles de acesso, fórum, etc) com o material em si.

Os professores devem entender as características intrínsecas a esse tipo de material instrucional, representado por um hiperdocumento, visto que se o desenvolvedor, já nos primeiros passos do desenvolvimento, percebe a maneira como o sistema é entendido pelos usuários, então seu projeto e implementação serão desenvolvidos criando um sistema mais intuitivo [14]. Analogamente, se o professor entende as necessidades de seu aluno e a forma como esse entende o material instrucional, então o professor estará desenvolvendo um material mais intuitivo, que pode facilitar o processo de ensino e aprendizagem.

O objetivo deste trabalho é avaliar o uso de um conjunto de estratégias cognitivas no apoio à geração e estruturação de material instrucional para EAD, resultando em um hiperdocumento, através da avaliação de usabilidade da interface computacional que contém o material, levando em consideração esse domínio de aplicação e as características de hiperdocumento.

Para avaliar do hiperdocumento que contém o material instrucional e validar o uso das estratégias cognitivas no contexto de EAD, é proposta a realização de estudos comparativos nos quais serão aplicadas avaliações de usabilidade no hiperdocumento, em versões distintas: sem o uso das estratégias cognitivas e com o uso das mesmas. As avaliações têm por objetivo verificar se a organização e estruturação de conteúdo, resultante do uso das estratégias cognitivas na edição do material instrucional, aumentam a usabilidade do hiperdocumento e da interface computacional gerada.

Neste trabalho, o termo material instrucional é utilizado para todo material didático que possa ser disponibilizado via rede de computadores, com o objetivo de ensinar. Também os atores: professor e aluno representam aqueles que desenvolvem e usam o material respectivamente.

USABILIDADE COMO REQUISITO DE QUALIDADE DE SOFTWARE

Segundo o modelo de qualidade da norma ISO/IEC 9126-1 [9] os atributos de qualidade de software são classificados em seis características:

1. Funcionalidade: capacidade que o software tem de prover funções que vão ao encontro de necessidades estabelecidas, quando o software está sob as condições especificadas;
2. Usabilidade: capacidade que o software tem de ser entendido, usado e aprendido, e também sua capacidade de agradar ao usuário, quando utilizado sob condições específicas;

VI Simpósio sobre Fatores Humanos em Sistemas Computacionais — Mediando e Transformando o Cotidiano. Hotel Bourbon, Curitiba, 17 a 20 de outubro de 2004. UFPR, CEIHC—SBC.

3. Confiabilidade: capacidade que o software tem de manter seu nível de desempenho, quando utilizado sob condições específicas;
4. Eficiência: a capacidade que o software tem de prover os requisitos de performance, relativo a quantidade de recursos utilizados, sob condições definidas;
5. Manutenibilidade: capacidade do software de ser modificado. Modificações podem incluir correções, melhorias ou adaptações do software com relação às mudanças no ambiente ou nos requisitos;
6. Portabilidade: capacidade do software de ser transferido de um ambiente computacional para outro.

Das seis características citadas anteriormente, a principal para este trabalho é a usabilidade, como será visto nos capítulos seguintes.

Cada uma das seis características da qualidade de software pode ainda ser desdobrada em sub-características. As sub-características de usabilidade são:

- Inteligibilidade: medida da facilidade do usuário para reconhecer a lógica de funcionamento do software e sua aplicação;
- Apreensibilidade: medida da facilidade encontrada pelo usuário para aprender a utilizar o software;
- Operacionalidade: medida da facilidade para operar o software.

Dessa forma, tem-se que usabilidade é uma característica de qualidade de software e então aumentando a usabilidade do hiperdocumento que contém o material instrucional para EAD, estar-se-á aumentando a qualidade do material em questão no seu aspecto computacional.

[2] relatam a importância de se preocupar com critérios de usabilidade, principalmente aqueles ligados com a organização e estruturação de conteúdo pelas interfaces que são utilizadas em EAD: “Interfaces devem ser focadas nas necessidades e objetivos dos usuários, provendo uma clara idéia da organização do conteúdo e funcionalidades do sistema, navegação simples, personalização avançada de caminhos e processos”.

[8] também citam como problemas de usabilidade para hiperdocumentos que contenham material instrucional para EAD:

- Navegação: os usuários tem dificuldades para encontrar a informação desejada ou não sabem como retornar a páginas anteriormente visitadas;
- Recursos multimídia: uso de maneira inadequada (uso abusivo de cores, *frames* e textos em destaque);
- Fatores culturais: as referências culturais utilizadas não são as mesmas para o mundo todo;
- Tecnologia: incompatibilidade entre *browsers* e plataformas de hardware, por exemplo.

Frente a esses problemas, se faz necessário auxiliar o professor na edição do hiperdocumento que contém o

material instrucional, com o intuito de garantir usabilidade ao hiperdocumento. Nesse contexto, as estratégias cognitivas, descritas a seguir, além de serem um recurso pedagógico comprovado no ensino presencial, vêm se mostrando um recurso para a estruturação e organização do conteúdo pela interface do hiperdocumento para EAD.

ESTRATÉGIAS COGNITIVAS PARA APOIO AO ENSINO

Gagné [6] aborda os processos internos de aprendizagem através de itens que foram denominados domínios. Um desses domínios é constituído pelas estratégias cognitivas, que segundo ele são capacidades internamente organizadas que o aluno usa para guiar seus próprios processos de atenção, aprendizagem, memória e pensamento. O aluno usa uma estratégia cognitiva, por exemplo, ao prestar atenção nas diversas características daquilo que está lendo. O leitor usa certas estratégias cognitivas para selecionar e codificar o que aprende, valendo-se de outras estratégias para recuperar posteriormente essas informações [1].

As estratégias cognitivas são, portanto, os meios que o aluno dispõe para administrar seus próprios processos de aprendizagem. Gagné relaciona tais estratégias com os conceitos de "aprender a aprender" e "aprender a pensar".

[3] define as estratégias cognitivas como “uma estratégia ou um grupo de estratégias ou procedimentos que os alunos usam para cumprir tarefas acadêmicas ou melhorar habilidades sociais. Normalmente, mais do que uma estratégia cognitiva é utilizada, dependendo do esquema de aprendizado do aluno”. As estratégias citadas em [3] são: Visualização; Verbalização; Associações; Particionamento; Questionamento; Inspeção; Grifar; Sinalização; Uso de mnemônicos; Reforço oral de algumas palavras; Auto-verificação; Monitoramento.

[4] apud [12] sugerem o uso das seguintes estratégias cognitivas: Ensaio; Mecanismo de mnemônicos; Auto-questionamento; Anotações.

[5] defendem o uso de estratégias cognitivas no ensino da língua inglesa e literatura e reforça em seu trabalho o uso de: Esquemas; Organizadores gráficos.

[17], que estudou o uso de estratégias cognitivas por quase 20 anos no ensino presencial, reforça que a melhor maneira de saber que estratégia utilizar é observar como os alunos mais experientes resolvem os problemas e que estratégias utilizam. Algumas das estratégias citadas em seu trabalho são: Quebra de tarefas; Guiar o aluno; Suporte; Feedback; Mapas de conceito.

[19] sugerem o uso de mais algumas estratégias cognitivas. As atividades apresentadas por [19] e utilizadas, com sucesso, por [11] no ensino presencial são listadas a seguir: Organização; Estruturação; Mapas de conceito; Metáforas e analogias; Ensaio; Organizadores de avanço.

Os exemplos citados acima representam apenas uma parte do universo de estratégias cognitivas que podem ser aplicadas no contexto educacional. A seguir, são apresentados alguns exemplos de utilização prática das estratégias citadas anteriormente:

- Em uma aula de Engenharia de Software, listar as semelhanças e diferenças entre Análise Estruturada e Análise Essencial [18];
- Em uma aula de geografia, explicar as alterações climáticas dos últimos anos indicando como uma possível causa o derretimento das calotas polares ou os efeitos do El Niño;
- Explicar que os benefícios do aleitamento materno têm uma relação direta com a queda da mortalidade infantil;
- O professor pode desenvolver um mapa hierárquico representando os paradigmas de linguagens de programação em uma aula sobre Linguagens de Programação [18];
- O professor pode desenvolver um diagrama com os tipos de advérbios em uma aula de Língua Portuguesa ou Literatura.
- Apresentar exercícios de fixação para os estudantes sobre o conteúdo visto na aula, resolvendo-os em seguida;
- O professor pode fazer perguntas no começo da aula a respeito do que será visto, ministrar a aula e ao final retomar essas perguntas e verificar se as respostas dos estudantes sofreram alterações.

DECIDE COMO SUPORTE À AVALIAÇÃO DE USABILIDADE DO MATERIAL INSTRUCIONAL

Para avaliar a usabilidade da interface computacional com o aluno, e validar o uso das estratégias cognitivas no contexto de EAD, é proposta a realização de estudos comparativos nos quais serão aplicadas avaliações de usabilidade no hiperdocumento que contém o material instrucional, em versões distintas: sem o uso das estratégias cognitivas e com o uso das mesmas. As avaliações têm por objetivo verificar se a organização e estruturação de conteúdo, resultante do uso das estratégias cognitivas na edição do material instrucional, aumentam a usabilidade da interface computacional gerada.

Independente do método ou modelo de avaliação que se deseja aplicar, [15] propõem a utilização de um *framework* que visa auxiliar avaliadores inexperientes no planejamento e na realização de uma avaliação – o DECIDE. Neste trabalho, as avaliações a serem realizadas serão apoiadas pelo *framework* DECIDE, da seguinte maneira:

1. Determinar os objetivos gerais que a avaliação deve tratar – Neste trabalho, a avaliação proposta tem como principal objetivo detectar problemas de usabilidade, principalmente aqueles relacionados à estruturação e organização do hiperdocumento, que afetam o entendimento do conteúdo instrucional;
2. Explorar perguntas específicas que devem ser respondidas – O aluno localiza pontos de interesse com facilidade no hiperdocumento? O aluno entende a relação entre os tópicos apresentados? O aluno demonstra interesse em aprender utilizando o hiperdocumento apresentado? Quais as impressões gerais do aluno quanto ao documento apresentado?

3. Escolher o paradigma e as técnicas de avaliação que poderão responder às perguntas elaboradas no item anterior – Pretende-se utilizar um método empírico e um método analítico de avaliação de usabilidade, visando obter diferentes perspectivas. Pelo levantamento bibliográfico realizado, percebeu-se que métodos de avaliação de usabilidade convencionais vêm sendo utilizados com sucesso para avaliação de softwares educacionais. No entanto, acredita-se que o contexto de EAD tem requisitos de usabilidade específicos e que possam ser necessárias algumas extensões aos métodos convencionais para avaliação do hiperdocumento que contém o material para EAD. Nesse sentido, entende-se que este trabalho também pode levantar esses requisitos de usabilidade específicos e propor possíveis extensões aos métodos utilizados;
4. Identificar questões práticas que precisam ser tratadas – O trabalho será realizado em parceria que está sendo estabelecida com o Departamento de Enfermagem da UFSCar e também com o auxílio do projeto TIDIA da FAPESP visando possibilitar a distribuição do material para análise, e também o recrutamento de alunos para os testes.
5. Decidir como lidar com questões éticas – Neste trabalho, serão utilizados os princípios sugeridos por [15] e também por [16].
6. Avaliar, interpretar e apresentar os dados – No intuito de levantar mais informações sobre os usuários do hiperdocumento que será avaliado, um questionário será editado visando minimizar suposições durante a aplicação do método analítico.

CONCLUSÃO

As dificuldades inerentes da distância física entre professor e aluno aumentam ainda mais a necessidade de professores preparados para a edição de cursos bem planejados, ambientes computacionais com recursos adequados e fáceis de serem utilizados e alunos motivados a adquirirem conhecimento à distância.

Neste trabalho defende-se a necessidade de considerar aspectos pedagógicos na edição do material instrucional para EAD, no intuito de facilitar a organização e estruturação do conteúdo pela interface, aumentando a usabilidade do material e conseqüentemente sua qualidade. Acredita-se que um conjunto de estratégias cognitivas, já validadas no ensino presencial, e apresentado aqui, pode ser o apoio pedagógico em questão, que se espera validar para o contexto de EAD através de avaliações de usabilidade no hiperdocumento que contém o material instrucional.

REFERENCIAS

1. Almeida, V. P., Silva, J. C. A. Planejamento Estratégico para Educação Apoiada por Computador Visando a Produção de Material Instrucional para EAD Baseado em Estratégias Cognitivas. Anais do XXIV SBC 2004.
2. Ardito C., Marsico, M., Lazilotti, R., Levialedi, S., Roselli, T., Rossano, V., Tersigni, M. Usability of E-learning Tools. Proc. AVI 2004. ACM Press (2004), 80 – 84.

3. Beckman, P. Strategy Instruction. ERIC Clearinghouse on Disabilities and Gifted Education Arlington. Educational Resources Information Center.
http://www.ericfacility.net/databases/ERIC_Digests/ed474302.html
4. Biehler; Snowman. Psychology Applied to Teaching. 8th. Edition. Houghton Mifflin Co., 1997.
5. Crandall, J.; Jaramillo, A.; Olsen, L.; Peyton, J. K. Using Cognitive Strategies to Develop English Language and Literacy.
<http://www.ericdigests.org/2003-3/using.htm>
6. Gagné, R. M. “The Conditions of Learning”. 3rd edition. Holt, Rinehart e Winston, 1974.
7. Gladcheff, A.P; Sanches R.; Silva, D.M.; Um Instrumento de Avaliação de Qualidade de Software Educacional: Como elaborá-lo. VII Workshop de Qualidade de Software. Simpósio Brasileiro de Engenharia de Software, 2001.
8. Gonçalves, L. L., Pimenta, M. S. EditWeb: Auxiliando Professores na Autoria de Páginas Web que Respeitem Critérios de Usabilidade. Novas Tecnologias na Educação V1 No.2.
<http://www.cinted.ufrgs.br/ciclo2/resumoleila.pdf>
9. ISO/IEC 9126-1. International Organization for Standardization. Information Technology – Software Product Quality. Part 1: Quality Model. 1999.
10. Lewis, C.; Brand C.; Cherry G.; Rader C. Adapting User Interface Design Methods to the Design of Educational Activities. CHI 1998 ACM Press (1998). 619-626.
11. Liebman, J. Teaching Operations Research: Lessons from Cognitive Psychology. Interfaces, 28 (2), 1998. 104-110.
12. Project Learning Based Space. Site da Universidade de Houghton Mifflin.
<http://college.hmco.com/education/pbl/tc/cog.html>
13. Prates, R.O.; Figueiredo R.M.V.; Bach, C.F. Um Modelo de Apoio ao Projeto de Interfaces e Ambientes de Aprendizado. IX WIE – Workshop de Informática na Escola. Anais do XXIII SBC, 2003.
14. Preece, J.; Rogers, Y.; Sharp, E. Human-Computer Interaction. Addison-Wesley, England, 1994.
15. Preece, J.; Rogers, Y.; Sharp, E. Interaction Design: Beyond Human-Computer Interaction. John Wiley a& Sons, New York, 2002.
16. Rocha, H. V.; Baranauskas, M. C.C. Design e Avaliação de interfaces humano-computador. 242p. IME-USP, São Paulo, 2000.
17. Rosenshine, B. The Case for Explicit, Teacher-led, Cognitive Strategy Instruction. Annual Meeting of the American Educational Research Association. Chicago, 1997.
<http://www.epaa.asu.edu/barak/barak1.html>
18. Silva, J. C. A. O Processo de Aprendizagem na Sala de Aula: Experiências e Desafios na Era do Computador. 24p. São Carlos, 1999. RT-DC 004/99, DC - UFSCar.
19. West, C. K.; Farmer, J. A.; and Wolff, P. M. Instructional Design: Implications from Cognitive Science. Allyn and Bacon. Boston, Massachusetts, 1991.

Integrando a Visão da ES e da IHC através da Aplicação de Padrões sobre o Modelo de Prototipação

André Constantino da Silva^{1*}

Júnia C. Anacleto Silva¹

Rosângela A. Dellosso Penteado¹

Sérgio Roberto P. da Silva²

1. Universidade Federal de São Carlos
Rod. Washington Luís, Km 235, São Carlos, SP
{andrecons,junia,rosangel}@dc.ufscar.br
+ 55 – 16 – 33608232

2. Universidade Estadual de Maringá
Av. Colombo, 5790, Maringá, PR
srsilva@din.uem.br
+ 55 – 44 – 261-4324

RESUMO

O modelo Prototipação é empregado pelas duas áreas, Engenharia de Software (ES) e Interação Humano-Computador (IHC), com diferentes visões. Considerar essas duas visões no processo de desenvolvimento é uma tarefa árdua, pois há uma divergência de foco. Nesse contexto, padrões, que estão sendo identificados e aplicados por ambas as áreas, podem auxiliar a amenizar os problemas resultantes da integração, e para que aspectos distintos de cada área no processo de desenvolvimento sejam considerados. Neste artigo é apresentada uma discussão sobre como padrões podem ser aplicados para integrar a visão da ES com a visão da IHC no modelo de processo Prototipação.

Palavras-chaves do Autor

Padrões, Integração da IHC com ES, Desenvolvimento de Sistemas Interativos, Modelo de Processo Prototipação.

INTRODUÇÃO

A IHC e a ES são subáreas da Ciência da Computação que tratam o desenvolvimento de sistemas interativos de forma sistemática. Entretanto, divergem quanto ao foco durante a realização do desenvolvimento de tais sistemas. Enquanto o foco principal da ES é o produto de software e seu processo, a IHC foca mais nos aspectos de interação do homem com a máquina [3].

Essas duas subáreas não são excludentes, mas complementares, sendo possível desenvolver sistemas interativos considerando práticas de ambas as áreas, objetivando o desenvolvimento de sistemas de uma forma mais abrangente, tratando tanto os aspectos considerados relevantes para a ES quanto os aspectos tratados pela IHC.

A integração da ES com a IHC não é uma tarefa fácil,

devido à divergência de focos e o fato dos aspectos de interação e tecnologia se influenciarem mutuamente. Outro fator que dificulta a integração das áreas é que, muitas vezes, os profissionais podem ter formações diferentes, dificultando a comunicação. Essa falta de comunicação impacta no produto construído, como por exemplo, a disponibilização de funcionalidades não necessárias e um código de difícil compreensão e manutenção.

Nesse contexto, padrões (*patterns*) podem ser utilizados para melhorar a comunicação entre os especialistas das diferentes áreas, além de expressar conhecimentos obtidos durante o desenvolvimento de sistemas, de tal forma que possam ser reaproveitados em outros projetos. Padrões são entendidos aqui como uma forma de expressar conhecimento através de textos e esboços em um formato estruturado, cuja solução é de sucesso para um problema que ocorre frequentemente em um determinado contexto [1].

Na próxima seção são apresentados o modelo de processo Prototipação e a visão da ES e da IHC sobre esse modelo de processo. Em seguida são apresentadas as características e vantagens dos padrões e a divisão dos padrões em categorias, que foram utilizadas para discutir a aplicabilidade dos padrões na Prototipação. Por fim, é discutido como padrões podem auxiliar a integrar essas duas visões nesse modelo de processo, e são apresentadas as conclusões parciais.

O MODELO DE PROCESSO PROTOTIPAÇÃO

O modelo Prototipação objetiva a realização de diversas iterações em que um protótipo é desenvolvido a partir de um conjunto de requisitos, e em seguida é avaliado pelo cliente [7]. Através dos resultados obtidos da avaliação, um novo protótipo, ou o sistema final, é construído.

Preece [6] e Sommerville [7] apresentam alguns modelos de Prototipação, entre eles podem-se citar a: a) **Prototipação Descartável**, que enfatiza o

VI Simpósio sobre Fatores Humanos em Sistemas Computacionais — Mediando e Transformando o Cotidiano. Hotel Bourbon, Curitiba, 17 a 20 de outubro de 2004. UFPR, CEIHC—SBC.

* Bolsista financiado pela CAPES.

desenvolvimento do protótipo e a sua avaliação, descartando-o quando os requisitos se tornam mais claros para então desenvolver o sistema final; b) **Prototipação Evolucionária**, cujo protótipo é desenvolvido em partes com as funcionalidades, modificado-o e acrescentando requisitos até satisfazer o cliente. A última versão do protótipo é o sistema final; e c) **Prototipação Incremental**, em que o protótipo é construído incrementalmente, sendo elaborado um projeto que considere todos os aspectos sem que as funcionalidades estejam completas.

Outro fator que difere o modelo Prototipação Descartável dos outros dois modelos está relacionado aos requisitos que são explorados no início do processo. Na Prototipação Descartável, são implementadas no protótipo os requisitos que estão menos claros, enquanto que nos outros dois modelos de Prototipação normalmente o processo se inicia implementando os requisitos que são mais bem compreendidos e com maior prioridade.

O foco da ES e da IHC na Prototipação também se diverge. O engenheiro de software, ao desenvolver protótipos, está preocupado com a compreensão e modelagem dos componentes internos do sistema, enquanto que o especialista de IHC está preocupado com aspectos da interação entre usuário e máquina. Pode-se concluir que, na Prototipação, a visão da ES é focada inicialmente na estrutura interna do sistema a ser desenvolvido, e a visão da IHC é focada nas tarefas a serem realizadas com o apoio do sistema, conforme é apresentado na Figura 1. As práticas propostas em cada uma das áreas, como métodos, técnicas e ferramentas, seguem a tendência do foco principal da área.

Considerar as duas visões na Prototipação não é uma tarefa fácil, pois algumas ocasiões o foco do protótipo será mais intenso nos aspectos externos, como ocorre na Prototipação Descartável, e em outras será na estrutura interna, para possibilitar flexibilidade ao sistema que futuramente pode receber novos requisitos, como o que ocorre na Prototipação Evolucionária. Outra questão é a comunicação entre os profissionais, que deve ser ampla para garantir que o protótipo desenvolvido permita explorar os aspectos de ambas as áreas. O processo também deve possibilitar o aprendizado dos profissionais

na área em que não é especialista, pois pode facilitar comunicação em projetos futuros. Esse aprendizado objetiva a compreensão dos aspectos relevantes da outra área, e não o domínio do especialista. Outro problema é realizar a conexão da estrutura externa, explorada pelo especialista em IHC, com a estrutura interna, explorada pelo engenheiro de software.

O modelo de processo Prototipação foi escolhido como o primeiro modelo a ser estudado para integrar as visões da ES com a da IHC através da aplicação de padrões devido aos fatos anteriormente apresentados e por ser parte integral do projeto interativo quanto por ser uma abordagem altamente participativa [6].

PADRÕES

Diversos padrões foram identificados tanto pela ES quanto pela IHC, e cuja aplicação vem sendo mais explorada. Essas áreas são motivadas pelas características dos padrões e os benefícios que sua aplicação traz ao processo de desenvolvimento. As características dos padrões são: Formato de Apresentação; Captura da Prática; Abstração; Princípio de Organização; Sistema de Valores; Facilidade de Compreensão.

Através dessas características, padrões possuem dois benefícios: a) fornecimento de um vocabulário comum, melhorando a comunicação entre a equipe de desenvolvimento e usuários leigos; b) captura da experiência, pois padrões podem ser utilizados para a transferência de conhecimentos entre pessoas de níveis de experiência diferentes.

Diversos são os padrões propostos pela ES, assim como pela IHC. Para facilitar a apresentação da sua aplicabilidade, os padrões foram agrupados em categorias. Para os padrões de ES, estende-se aqui o grupo de categorias proposto por Buschmann *et al.* [4], incluindo as categorias padrões de análise e padrões de testes, resultando no seguinte grupo: Padrões de Análise; Padrões Arquiteturais; Padrões de Projeto; Padrões de Implementação ou Idiomas; Padrões de Testes.

Não se encontra na literatura uma classificação tão clara dos padrões de IHC quanto as encontradas para os padrões de ES, como a descrita anteriormente. Por isso, foram utilizadas as categorias propostas por Welie e van der Veer [9], as apresentadas em Alpert [2], e padrões para avaliação de usabilidade.

Welie e van der Veer dividiram seus padrões nas seguintes categorias: Metas de Negócio; Padrões de Postura; Padrões de Experiência; Padrões de Tarefas; Padrões de Ação. O outro grupo de categorias considerado divide os padrões da seguinte forma: Padrões de Interação Humano-Computador; Padrões de Interface com o Usuário.

Na próxima seção é apresentado como os padrões de ES e de IHC existentes, considerando as categorias descritas, podem ser aplicados no modelo Prototipação.

PADRÕES NA PROTOTIPACÃO

Na Tabela 1 é apresentado em qual etapa do modelo de processo Prototipação é possível aplicar padrões,

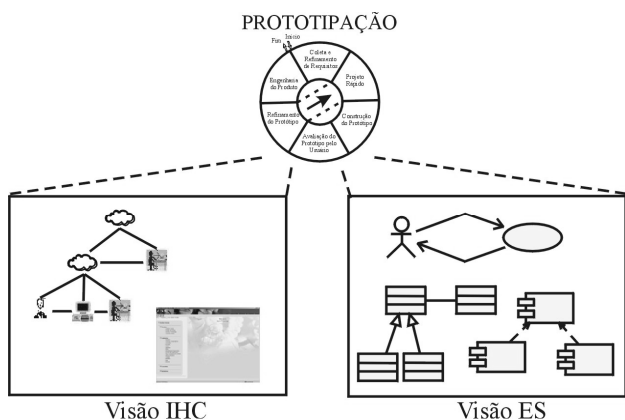


Figura 1 – A visão da IHC e visão da ES sobre o modelo de processo Prototipação.

Modelo de Prototipação	Descartável	Evolucionária ou Incremental
Coleta e Refinamento de Requisitos	Análise (ES), Metas de Negócio (IHC), Postura (IHC), Experiência (IHC), Tarefas (IHC)	Análise (ES), Metas de Negócio (IHC), Postura (IHC), Experiência (IHC), Tarefas (IHC)
Projeto Rápido	Ação (IHC), Interação (IHC), Interface com o Usuário (IHC)	Ação (IHC), Arquitetural (ES), Interação (IHC), Interface com o Usuário (IHC), Projeto (ES)
Construção do Protótipo	Testes (ES)	Idiomas (ES), Testes (ES)
Avaliação do Protótipo pelo Usuário	Avaliação de Usabilidade (IHC)	Avaliação de Usabilidade (IHC)
Refinamento do Protótipo	Análise (ES), Metas de Negócio (IHC), Postura (IHC), Experiência (IHC), Tarefas (IHC)	Análise (ES), Metas de Negócio (IHC), Postura (IHC), Experiência (IHC), Tarefas (IHC)
Engenharia do Produto	Todos as Categorias (ES, IHC)	

Tabela 1. Categorias de padrões nas etapas do modelo de processo Prototipação, considerando modelos existentes.

considerando as categorias descritas anteriormente, nos modelos de Prototipação Descartável, Evolucionária e Incremental.

Durante a etapa de coleta e refinamento de requisitos e na etapa de refinamento do protótipo, o engenheiro de software e o especialista em IHC estão preocupados em compreender melhor o domínio do problema, enquanto que o usuário está tentando se expressar da melhor forma possível. Padrões que possuem informações relevantes ao domínio, como por exemplo, alguns padrões de Análise, os padrões de Metas de Negócio, Postura, Experiência e Tarefas podem ser aplicados.

Na etapa de projeto rápido, os profissionais estão elaborando um projeto que considere um conjunto de requisitos. Esse conjunto de requisitos, como dito anteriormente, dependerá do modelo utilizado. Se for utilizado a Prototipação Descartável, pouco tempo deve ser despendido elaborando um projeto, com pouca preocupação da estrutura interna do protótipo. Logo, padrões de Ação, Interação e Interface com o Usuário podem ser aplicados nessa etapa. Esses padrões também podem ser aplicados quando considerado o modelo Prototipação Evolucionária ou o Incremental, podendo

também aplicar os padrões arquiteturais e de projeto, pois nesses dois modelos existe uma preocupação com a estrutura interna.

Durante a etapa de construção do protótipo, podem ser aplicados padrões de testes, seja qual for o modelo de Prototipação adotado, sendo diferente a quantidade de padrões adotados. Na Prototipação Descartável, aplicam-se testes no protótipo apenas para obter um certo grau de confiabilidade, antes de realizar a avaliação pelo cliente na próxima etapa. Nos outros modelos de Prototipação, aplica-se um maior número de padrões de testes, pois, sendo o último protótipo o sistema final, é exigido um grau de confiabilidade maior para evitar a propagação de erros. Quando o modelo for Evolucionário ou Incremental pode-se também aplicar um maior número de Idiomas na construção do protótipo. Caso contrário, não se sugere aplicar padrões de projeto ou idiomas por ser entendido ser um esforço desnecessário para um protótipo descartável.

Durante a etapa de avaliação do protótipo pelo cliente, em qualquer modelo de Prototipação utilizado, padrões para avaliação de usabilidade podem ser aplicados para obter o *feedback* do usuário.

No caso da Prototipação Descartável, após refinar os requisitos, o protótipo é descartado e então o sistema é construído, seguindo critérios de qualidade e boas práticas definidas pela ES e pela IHC. Nessa etapa, padrões de todas as categorias listadas podem ser aplicados, principalmente os que se mostraram com bons resultados durante a elaboração do protótipo. Quando o modelo é Evolucionário ou Incremental, essa etapa não é executada, pois o último protótipo é o sistema final.

Somando os padrões de ES com os padrões de IHC, é possível facilitar a Prototipação através do apoio que eles fornecem nas etapas que são realizadas ali. Essa constatação pode ser visualizada quando se considera o diagrama do modelo Prototipação apoiado com a aplicação de padrões, conforme apresentando na Figura 2.

Através dessa discussão, percebe-se que os padrões de IHC também representam o foco principal da área, ou seja, a estrutura externa do sistema, enquanto os padrões da ES representam a visão inicial que a área tem, ou seja, a estrutura interna.

INTEGRANDO AS VISÕES DA PROTOTIPAÇÃO

Padrões podem ser aplicados no processo de desenvolvimento para amenizar ou resolver os problemas de integração entre IHC e ES.

O problema da comunicação entre os profissionais pode ser amenizado através da aplicação de padrões, pois eles possuem como vantagem a melhoria da comunicação entre os profissionais, pois fornecem um vocabulário comum entre eles. O padrão também representa valores relevantes da área que o propõe e é escrito para ser de fácil leitura, colaborando assim com o aprendizado desses profissionais, bem como facilitando o diálogo entre os profissionais e os usuários.

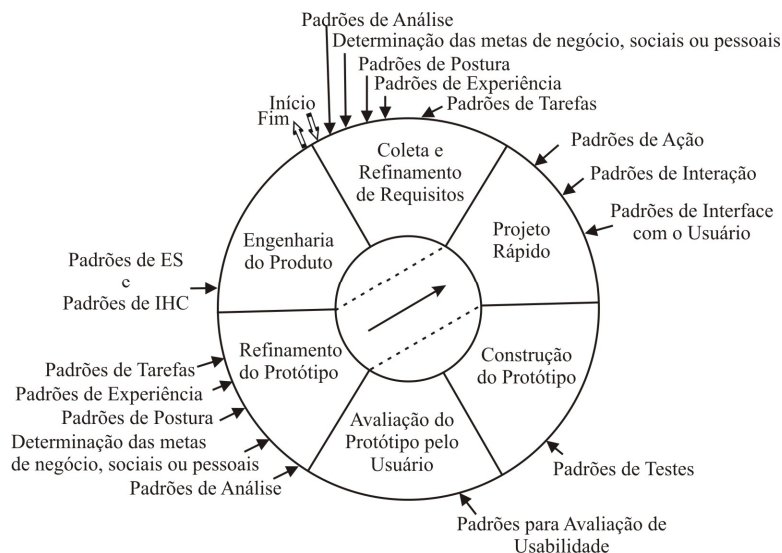


Figura 2 – Relacionamento das categorias de padrões de IHC e de ES com as etapas do modelo Prototipação Descartável.

Na Prototipação, a diferença do foco entre IHC e ES pode ser minimizada através da aplicação conjunta dos padrões de ambas as áreas, considerando o relacionamento entre eles. Por exemplo, o especialista em IHC, vendo a necessidade de disponibilizar comandos como desfazer e refazer, pretende aplicar o padrão *Multi-Level Undo* [8]. Entretanto, como o engenheiro de software está focado na representação interna do sistema, ele pretende aplicar o padrão *Command* [5]. Percebe-se que estes padrões estão relacionados, pois tratam da disponibilização dos comandos desfazer e refazer. Através do relacionamento dos padrões é possível pensar nas diretivas para a estrutura externa (fornecida pelo padrão *Multi-Level Undo*) e também na estrutura interna (fornecida pelo padrão *Command*).

CONCLUSÕES PARCIAIS

Diversas são as abordagens propostas de integração da ES com a IHC, entretanto poucas delas consideram a aplicação de padrões no processo de desenvolvimento, e tratam tal aplicação de forma aleatória e limitando-se a um certo número de padrões. Este artigo apresenta uma discussão da aplicabilidade de padrões da ES e da IHC no modelo de processo Prototipação, amplamente utilizado pelas duas áreas, de forma sistemática e considerando um grande conjunto de padrões e linguagens de padrões, na ordem de 300 padrões.

Padrões podem auxiliar na integração das visões da ES e da IHC amenizando problemas da integração. Além de apresentar boas soluções para problemas recorrentes das duas áreas através da captura da prática, o que permite o reuso de informações, apresentam também valores considerados relevantes para a área que o propõe e são de fácil compreensão, podendo ser utilizados para melhorar a comunicação entre os profissionais e usuários envolvidos. Como trabalhos futuros, pretende-se considerar padrões de processo e organizacionais da ES,

assim como relacionar os padrões das duas áreas, montando uma coleção para desenvolvimento de sistemas interativos, e realizar validações da proposta através de estudos de caso.

REFERÊNCIAS

- Alexander, C., Ishikawa, S., Silverstein, M., Jacobson, M., Fiksdahl-king, I., Angel, S. *A Pattern Language: Towns, Buildings, Construction*. Oxford University Press, Nova Iorque, NY, EUA, 1977.
- Alpert, S. R.. Getting Organized: Some Outstanding Questions and Issues Regarding Interaction Design Patterns. *Workshop on Perspectives on HCI Patterns*, CHI'2003.
- Brown, J. Methodologies for the Creation of Interactive Software. Tech. Rep. CS-TR-96/1, Victoria University of Wellington, Wellington, Nova Zelândia, mai., 1996.
- Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., Stal, E. *Pattern-Oriented Software Architecture. Volume 1: A System of Patterns*. John Wiley & Sons Ltd., Chichester, UK, 1996.
- Gamma, E., Helm, R., Johnson, R., Vlissides, J. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Boston, EUA, 1995.
- Preece, J. *A Guide to Usability: Human factors in computing*. Addison-Wesley Pub. Co., Reading, MA, EUA, 1993.
- Sommerville, I. *Engenharia de Software*, 6.ed. Addison-Wesley Pub. Co., São Paulo, SP, Brasil, 2003.
- Tidwell, J. UI Patterns and Techniques. <http://time-tripper.com/uipatterns/index.php>.
- Welie, M. Van, van der Veer, G. C., Eliëns, A. Patterns as Tools for User Interface Design. *Proc. Int. Workshop on Tools for Working with Guidelines*, 2000, 313-324.

Modelando a interação do NiTA: um estudo de caso e extensões ao MoLIC

Bruno Santana da Silva

SERG, Departamento de Informática, PUC-Rio
Rua Marquês de São Vicente, 225
Rio de Janeiro, RJ, Brasil – 22453-900
brunosantana@inf.puc-rio.br

Simone Diniz Junqueira Barbosa

SERG, Departamento de Informática, PUC-Rio
Rua Marquês de São Vicente, 225
Rio de Janeiro, RJ, Brasil – 22453-900
simone@inf.puc-rio.br

RESUMO

Em linha com esforços recentes de projeto da interação para múltiplos dispositivos, este trabalho apresenta um estudo de caso do MoLIC, linguagem de modelagem de interação como uma conversa, no projeto do NiTA (Notes in the Air). Trata-se de uma aplicação de comunicação assíncrona para dispositivos móveis. São apresentadas limitações do MoLIC para a representação de sistemas interativos, e propostas extensões para superar estas limitações, tais como: estruturação de diálogos, pré-condições de falas de transição, e relações entre modelos de interação projetados para usuários distintos.

Palavras-chave

projeto da interação humano-computador baseado em modelos, modelo de interação (MoLIC), interação como uma conversa

INTRODUÇÃO

O desenvolvimento das interfaces de usuário para dispositivos móveis trouxe novos desafios para área de IHC, já que as formas de interação nestes dispositivos são restritas e as situações de uso são bem diferentes se comparadas às aplicações em plataformas estacionárias. Por exemplo, muitos dispositivos móveis possuem tamanho de tela e entrada de dados restritos, não podendo exigir o emprego de duas mãos em certas situações de uso [4]. Além disso, eles permitem que a interação possa ocorrer onde o usuário estiver (andando, subindo escadas, no ônibus, etc.), o que requer que a aplicação se adapte ao contexto de uso. Estas novas situações de uso limitam a capacidade de atenção e o tempo de interação do usuário [5]. Esses desafios motivaram o estudo de caso aqui apresentado utilizando modelos de IHC, e em particular o MoLIC, linguagem que visa permitir a modelagem da interação de forma independente do dispositivo a ser utilizado [1,6].

Este artigo descreve algumas soluções de interação para o sistema NiTA (*Notes In The Air*), desenvolvido em parceria com o Laboratório de Colaboração Avançada (LAC, <http://www.lac.inf.puc-rio.br>) da PUC-Rio. NiTA é um sistema de comunicação assíncrona para dispositivos móveis. Ele permite a troca de mensagens

entre os usuários considerando o local em que eles se encontram. Isto significa que um usuário poderá enviar uma mensagem para um determinado local, e os destinatários que passarem ou estiverem naquele local receberão a mensagem enviada.

Durante o projeto da interação do usuário com o NiTA, construímos alguns cenários de uso [2], modelos hierárquicos de tarefas e o modelo de interação, conhecido como MoLIC – do inglês “Modeling Language for Interaction as Conversation” [1,6].

Esbarramos com algumas limitações do MoLIC, principalmente pelo fato de o modelo ser proposto apenas para a representação da interação de um único usuário com o sistema. Esta limitação já havia sido prevista pela própria autora. Além disto, foram detectadas outras limitações do MoLIC para representar as soluções sendo projetadas. Existem características de conversas que têm um paralelo interessante com fenômenos de IHC que podem ser modelados segundo a mesma metáfora de interação como uma conversa. Em particular, como se trata de uma aplicação voltada à comunicação entre pessoas, devemos deixar claras as relações ou influências das falas de um usuário sobre outro.

Este trabalho estende o MoLIC definindo a semântica e a notação gráfica de alguns novos elementos do modelo. Partindo de uma tarefa do usuário, apresentamos como o MoLIC original pôde representar a interação dele com o sistema. Em seguida, mostraremos em que pontos o MoLIC original falhou na representação da interação esperada para a realização da tarefa proposta. Por fim, mostraremos a definição de alguns novos elementos do modelo de interação propostos e como eles foram usados para melhor representar o modelo de interação do NiTA. Outros elementos estão sendo propostos para representar fenômenos de interação específicos, mas não serão descritos neste artigo por limitações de espaço.

MODELO HIERÁRQUICO DE TAREFAS

A Figura 1 mostra o modelo hierárquico de tarefas para a meta *Enviar Mensagem*.

Para um usuário enviar uma mensagem utilizando o NiTA, ele obrigatoriamente deverá informar o destinatário e fornecer o conteúdo da mensagem. Opcionalmente, ele poderá informar o local de destino da mensagem. Caso ele não informe, a mensagem será enviada para o local de onde foi postada. Ele ainda pode optar por definir o tempo de vida da mensagem, indicando de forma explícita uma data e hora para o

limite de entrega da mesma, ou, de forma alternativa, vinculando o seu tempo de vida ao tempo de permanência dele no local de envio.

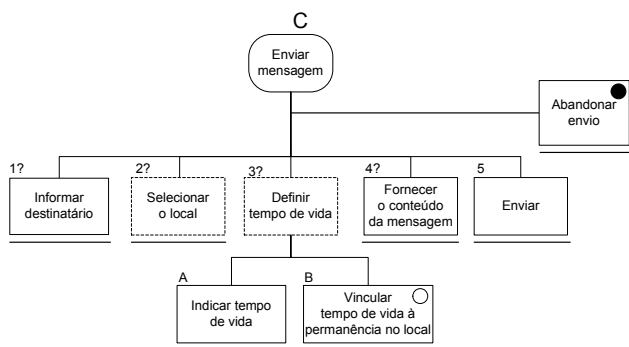


Figura 1. Modelo hierárquico de tarefas da meta Enviar Mensagem.

Após o usuário efetuar o envio, o receptor da mensagem só poderá lê-la quando ele estiver ou entrar no local que recebeu a mensagem antes do fim do seu tempo de vida. Sendo assim, o resultado da execução de uma tarefa para um usuário (enviar mensagem) poderá determinar a execução de outra tarefa por outro usuário do mesmo sistema (ler mensagem), em outro dispositivo.

MODELO DE INTERAÇÃO

Vejamos como a interação para a meta Enviar Mensagem pode ser modelada no MoLIC original (Figura 2).

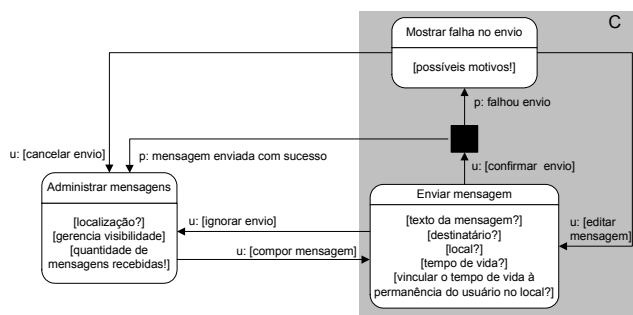


Figura 2. Modelo de interação da meta Enviar Mensagem segundo o MoLIC original.

Limitações do MoLIC Original

Neste modelo não foi possível representar a influência da interação do usuário sobre a interação de outro usuário (externo ao contexto de interação representado). A interação com o usuário externo passa despercebida pela transição de fala *p:mensagem enviada com sucesso*. Também não foi possível representar a indicação de obrigatoriedade da ocorrência dos diálogos *[texto da mensagem?]* e *[destinatário?]* como pré-condição para a transição de fala *u:[confirmar envio]*. Além disso, não pudemos indicar que os diálogos *[tempo de vida?]* e *[vincular o tempo de vida à permanência do usuário no local?]* são mutuamente exclusivos, ou seja, não faz sentido falar sobre ambos os (sub)tópicos.

Diante destas limitações, passamos a estudar as situações que ocorrem durante a interação com dispositivos móveis, procurando entender melhor os elementos e formas de interação que ainda não são suportados pelo MoLIC.

Neste estudo, conseguimos definir algumas extensões para superar as limitações encontradas, no âmbito da Engenharia Semiótica [3] e da metáfora de interação como uma conversa.

Na próxima seção são definidos alguns dos elementos de interação que compõem nossa proposta de extensão do MoLIC.

Novos Elementos de Interação

Diálogos Mutuamente Exclusivos

Em uma conversa, existem tópicos alternativos dentre os quais somente um é escolhido para ser travado entre os falantes. Analogamente, na interação em uma cena pode haver um conjunto de diálogos potenciais dentre os quais somente um pode ser escolhido para ser travado.

Na Figura 3(a) podemos observar como os diálogos mutuamente exclusivos são representados. O conjunto deste tipo de diálogos também é representado entre colchetes, só que precedidos pelo indicador de seleção *ou* e separados internamente por vírgulas. No lado (b) da Figura 3, observamos como as tarefas alternativas *indicar tempo de vida* (signo *tempo de vida?*) e *vincular tempo de vida à permanência no local* (signo *vincular o tempo de vida à permanência do usuário no local?*) foram mapeadas para o modelo de interação como diálogos mutuamente exclusivos.

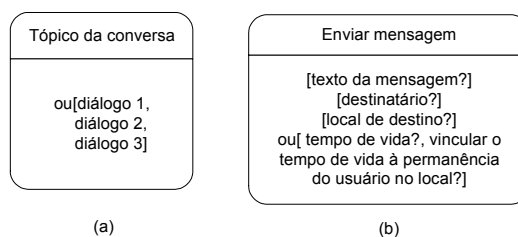


Figura 3. Representação gráfica dos diálogos mutuamente exclusivos.

Diálogos Obrigatórios

Para certas transições entre tópicos ocorrerem, pode ser necessário que anteriormente tenha(m) sido travado(s) algum(ns) diálogo(s) na cena de origem. Isto torna necessário representar os diálogos correspondentes como sendo obrigatórios para estas transições. Como pode haver uma combinação não-trivial de diálogos relacionados a cada transição, não é suficiente apenas dizer que um diálogo é obrigatório numa cena. A obrigatoriedade de um ou mais diálogos deve estar vinculada à transição.

Não necessariamente uma transição exige a ocorrência de todos os diálogos obrigatórios. Por exemplo, se estivermos modelando a interação de um formulário de busca dos materiais de uma biblioteca, poderíamos informar o nome do autor, ou o nome da obra, ou ainda algumas palavras-chave do assunto antes de efetivar a busca. Deste modo, percebemos que existem dois tipos de relações entre as pré-condições de ocorrência de uma fala de transição: a relação de conjunção (*e*) e a relação de disjunção (*ou*). Alguém ainda poderia levantar a possibilidade de haver uma relação de disjunção exclusiva, mas acreditamos não ser necessária pois esta

relação poderia ser obtida pelos diálogos mutuamente exclusivos definidos acima.

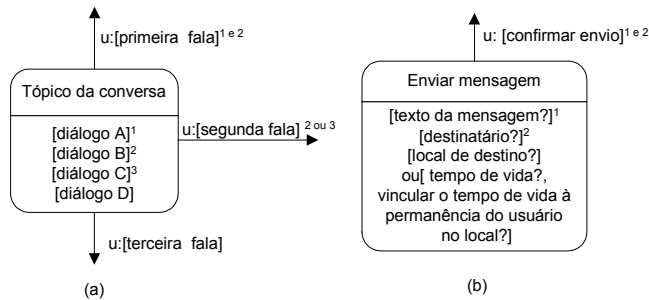


Figura 4. Representação gráfica dos diálogos obrigatórios para certas transições.

A Figura 4(a) mostra um exemplo de representação dos diálogos obrigatórios para determinadas transições. Os números sobre-escritos ao lado direito da descrição dos diálogos são seus identificadores. Podemos observar o diálogo [diálogo A] identificado por ¹, o diálogo [diálogo B] identificado por ² e o diálogo [diálogo C] identificado por ³. Não é necessário identificar todos os diálogos da cena, apenas os diálogos obrigatórios para alguma transição. No exemplo acima, podemos perceber que o diálogo [diálogo D] não está identificado porque nenhuma transição o tem como pré-requisito.

Com este identificador, é possível referenciar o diálogo em qualquer fala de transição, indicando que aquela transição exige que tenham sido travados previamente os referidos diálogos. Observando o exemplo da Figura 4 (a), podemos perceber que a transição de fala *u:[primeira fala]* ^{1 e 2} exige que, segundo a relação de conjunção, ambos os diálogos [diálogo A] e [diálogo B] ocorram antes dela. Já a transição de fala *u:[segunda fala]* ^{2 ou 3}, segundo a relação de disjunção, tem como pré-requisito a ocorrência de pelo menos um dos diálogos referidos. Todavia, a disjunção não impede que mais de um ou todos os diálogos ocorram antes da execução. No caso da fala de transição *u:[terceira fala]*, podemos perceber que, como não há referência a diálogos, ela pode ocorrer sem que qualquer diálogo da cena tenha sido travado, ou seja, todos os diálogos serão opcionais para ela. No lado (b) da Figura 4, podemos observar como a obrigatoriedade dos diálogos [texto da mensagem?] e [destinatário?] para a ocorrência do envio da mensagem está representado graficamente no modelo de interação.

Interação entre usuários

Alguns diálogos travados durante a interação usuário-sistema têm como interlocutor, direto ou indireto, um ator externo ao contexto imediato de interação. No MoLIC, isto não era considerado.

Neste trabalho, propomos que seja elaborado um modelo em MoLIC para cada ator ou papel de usuário envolvido na interação (considerando um modelo de usuário conforme proposto por [7]). Em cada modelo será projetada a interação usuário-sistema sob o respectivo ponto de vista, levando o designer a refletir sobre a interação com o sistema de cada (papel de) usuário separadamente. Como grande parte das aplicações atuais são aplicações multi-usuário, torna-se necessário incluir

no modelo uma representação para quando a interação de um ator P_1 com o sistema influenciar a interação de um outro ator P_2 . Optamos por representar esta situação através de pontos de contato no MoLIC.

Cada ponto de contato é representado graficamente por um círculo com um rótulo de onde saem falas do par usuário-sistema para algum(ns) ator(es) externo(s), ou vice-versa (Figura 5). Para facilitar a leitura dos modelos, sugerimos ainda indicar os (papéis de) usuários envolvidos nos pontos de contato, como uma referência ao MoLIC que conterá o ponto de contato correspondente (de origem ou de destino da fala).

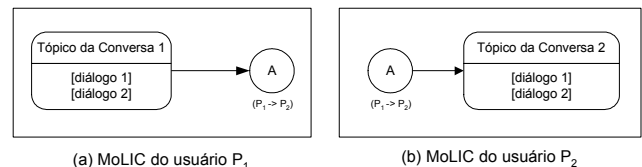


Figura 5. Representação gráfica de falas com atores externos.

Na Figura 5 podemos ver a representação gráfica equivalente ao usuário P_1 emitir uma fala “para” o usuário P_2 , isto é, uma fala que afeta a interação de P_2 com o sistema. No lado (a) desta figura, vemos o MoLIC do ponto de vista do usuário P_1 . Neste caso, a emissão da fala de P_1 está representada pela seta em direção ao ponto de contato A. Já no lado (b) desta figura, vemos o MoLIC do ponto de vista do usuário P_2 . Aqui, P_2 está recebendo a fala emitida por P_1 proveniente do ponto de contato A. Vale notar que junto ao ponto de contato A, estão indicados os usuários envolvidos: $(P_1 \rightarrow P_2)$.

As falas associadas a pontos de contato podem ser originárias de, ou destinadas a, qualquer ponto da interação usuário-sistema, seja um processo ou uma cena.

Modelagem da interação segundo o MoLIC estendido

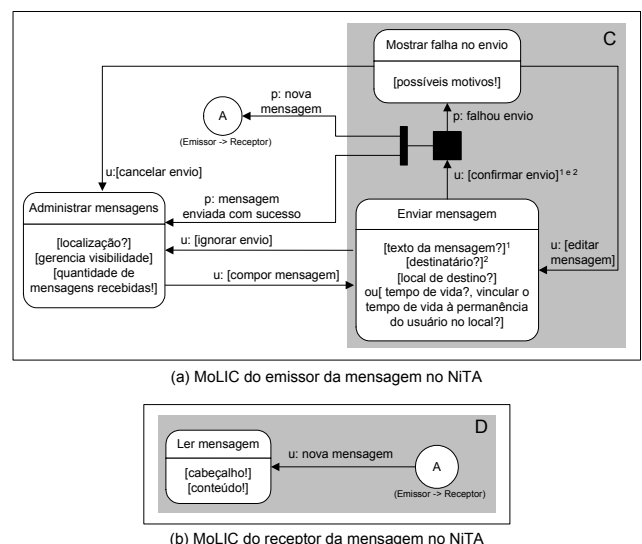


Figura 6. Modelo de interação das metas **Enviar Mensagem (a)** e **Receber Mensagem (b)** segundo o MoLIC estendido.

A Figura 6 mostra o modelo de interação da meta *Enviar Mensagem* conforme a extensão do MoLIC aqui proposta. Podemos ver, neste exemplo, o quanto melhorou a representação da interação segundo o MoLIC estendido.

As tarefas alternativas foram mapeadas para diálogos mutuamente exclusivos, as tarefas opcionais foram mais bem expressas quando se informou explicitamente quais eram os diálogos obrigatórios. Por fim, pôde-se representar a interação do ponto de vista do emissor (a) e do receptor (b), permitindo ao designer modelar e refletir sobre quais são as influências na interação do receptor, quando o emissor emite alguma fala.

Observe que uma barra vertical também foi introduzida ao modelo como um elemento de bifurcação para quando a conversa extrapola o contexto usuário-sistema e envolve atores externos. Este elemento também compõe a extensão ao MoLIC, que por limitações de espaço não pôde ser completamente descrita neste artigo.

CONSIDERAÇÕES FINAIS

Os elementos acrescentados ao MoLIC permitiram representar melhor o mapeamento do modelo de tarefas para o modelo de interação. Se voltarmos ao modelo de tarefas da Figura 1, poderemos verificar que existiam tarefas opcionais e alternativas. Quando mapeamos estas tarefas em um modelo de interação usando o MoLIC original, não ficava clara a diferença na forma de interação entre tarefas opcionais, tarefas alternativas e as outras tarefas. Todas eram tratadas igualmente. Acreditamos que o MoLIC acrescido dos diálogos obrigatórios possa dar conta do mapeamento de tarefas opcionais. E se acrescido dos diálogos alternativos, ele também possa dar conta do mapeamento das tarefas alternativas.

A versão original do MoLIC foi definida para abranger apenas a interação usuário-sistema. Este nível de interação foi definido por [7] como sendo um nível individual, pois o usuário interage exclusivamente como o sistema. Quando, neste trabalho, definimos diálogos entre usuários externos no MoLIC, ele passa agora a ser capaz de representar a interação no nível interpessoal, através da aplicação. O próximo nível de interação necessário para o desenvolvimento de aplicações multi-usuário seria o nível de contexto. Desta forma, a inclusão dos diálogos externos ao MoLIC deu mais um passo em direção à modelagem da interação de aplicações de grupos.

Vale notar que as extensões ao MoLIC aqui propostas não se limitam a aplicações para dispositivos móveis, ou seja, podem ser utilizadas na modelagem da interação de aplicações em geral. O MoLIC tem sido usado durante o

trabalho com a equipe de engenheiros de software do LAC desde a fase de análise de requisitos até a fase de especificação de software para modelar e promover a reflexão sobre a interação do usuário com o NiTA.

AGRADECIMENTOS

Simone D. J. Barbosa agradece ao CNPQ pelo suporte financeiro para este trabalho. Bruno Santana da Silva agradece a CAPES pela bolsa de mestrado recebida. Ambos os autores agradecem aos seus colegas do Grupo de Pesquisa de Engenharia Semiótica e do LAC, ambos da PUC-Rio, pelos seus valiosos comentários sobre as idéias apresentadas neste artigo. Agradecemos especialmente à Maira Greco de Paula pelo valioso trabalho inicial.

REFERÊNCIAS

1. Barbosa, S.D.J.; Paula, M.G. “Designing and Evaluating Interaction as Conversation: a Modeling Language based on Semiotic Engineering” In J.Jorge; N. J. Nunes; J. Falcão e Cunha (eds.) *Proceedings of the 10th International Workshop, DSV-IS 2003*, Funchal, Madeira Island, Portugal, LNCS 2844, 2003, pp. 16–33.
2. Carroll, J. M. (ed) *Scenario-based Design: Envisioning Work and Technology in System Development*. New York, NY. John Wiley and Sons, 1995.
3. de Souza, C. S. *The Semiotic Engineering of Human-Computer Interaction* (in press). The MIT Press.
4. Kawachiya, K., Ishikawa, H., “NaviPoint: An Input Device for Mobile Information Browsing”, *Proceedings of CHI 98*, Los Angeles, CA, 1998, pages 18-23.
5. Pascoe, J., Ryan, N., Morse, D., “Using While Moving: HCI Issues in Fieldwork Environment”, *ACM Transactions on Computer-Human Interaction*, Vol. 7, nº 3, September 2000, pages 417-437.
6. Paula, M.G., *Projeto da interação humano-computador baseado em modelos fundamentados na engenharia semiótica: construção de um modelo de interação*, Dissertação de Mestrado, DI, PUC-Rio, 2003.
7. Prates, R. O., *A Engenharia Semiótica de Linguagens de Interfaces Multi-Usuário*, Tese de Doutorado, DI, PUC-Rio, 1998.

Range-sliders alternativos para representar grandes quantidades de valores

Celmar Guimarães da Silva

Heloísa Vieira da Rocha

Instituto de Computação – Universidade Estadual de Campinas
Campinas – São Paulo – Brasil
{celmar, heloisa}@ic.unicamp.br
+55 19 3788-5866

RESUMO

Range-sliders são controles que possibilitam aos usuários selecionar via manipulação direta um intervalo de valores dentro de um determinado domínio. Embora de fácil utilização quando o intervalo contém poucos valores, esses controles não são adequados para representar faixas com grande quantidade de valores. Desenvolver alternativas para trabalhar nessas situações possibilita ao mesmo tempo representar grandes faixas de tempo (anos ou meses) em *range-sliders* e selecionar intervalos de tempo de diferentes grandezas (meses, dias, horas ou minutos). Este artigo apresenta duas alternativas que visam superar a limitação citada: empregar técnicas de distorção, ou utilizar uma combinação de dois ou mais *range-sliders*.

ABSTRACT

Range-sliders are widgets that enable users to select through direct manipulation an interval of values into a defined domain. Although range-sliders are easy to use when its interval has few values, these widgets are not suitable to represent intervals having a large amount of values. Developing alternatives to work on these situations makes possible both representing big time intervals (years or months) on range-sliders, and enabling users to select time intervals of different order of magnitude (months, days, hours or minutes). This paper presents two possible alternatives to overcome the presented limitation: applying distortion techniques, or using a combination of two or more range-sliders.

Palavras-chaves

Range-sliders, distorção, foco + contexto.

INTRODUÇÃO

Para efetuar consultas sobre um conjunto de dados de um sistema, o usuário precisa informar os parâmetros que compõem essas consultas, para que então o sistema procure e apresente os resultados a eles relacionados. Uma maneira de possibilitar isso é pedir aos usuários que preencham um ou mais campos de um formulário e pressionem um botão do tipo Buscar ou algo semelhante, para que então o resultado da busca apareça na tela. Contudo, a não tolerância a erros do usuário e um atraso significativo entre a formulação da consulta e a exibição do seu resultado [8] são sérios contratempos que muitas vezes desestimulam o uso da ferramenta de consulta pelo usuário.

Uma abordagem que procura evitar esses contratempos é a de consultas dinâmicas [3]. Essa abordagem é caracterizada pelos seguintes princípios de manipulação direta: apresentação visual dos parâmetros da consulta, bem como dos resultados; controle rápido, incremental e reversível da consulta, seleção de parâmetros por apontamento e não por digitação; e *feedback* contínuo e imediato (em menos de 100 ms) de cada modificação dos componentes da consulta [7]. Dentre as diversas vantagens de consultas dinâmicas, derivadas do modo de manipulação direta, destaca-se a facilidade de aprendizagem do usuário sobre como efetuar uma consulta, sem precisar entender complexas sintaxes de linhas de comando, por exemplo.

Segundo esses conceitos, cada tipo de situação exige controles adequados. Por exemplo, caixas de seleção e botões liga/desliga podem ser usados para ativar ou desativar um parâmetro booleano. Para selecionar um valor em uma lista de valores, podem ser usados *radiobuttons* ou listas de seleção. *Sliders* podem ser usados no caso de valores (numéricos ou alfanuméricos) ordenados, como em [1] e [2].

Há situações especiais em que existe uma seqüência ordenada de valores na qual o usuário deve selecionar um intervalo. Por exemplo, o usuário deseja consultar todos os filmes lançados entre 2002 e 2004 numa base de dados de uma locadora. Para esses casos, poderiam ser usados dois *sliders*, um para o limite inferior do intervalo e outro para o superior. Contudo, a existência de dois controles aparentaria uma independência entre eles, além de ocupar mais espaço na tela.

Prevenindo esses problemas, o *range-slider* (ou *range-selection slider*) [4] é um tipo de controle que, usando o mesmo espaço de um *slider* comum, permite ao usuário selecionar um intervalo de valores dentro de um limite estipulado. Como mostrado na Figura 1a, um *range-slider* apresenta dois seletores (normalmente em formato de setas) delimitando o intervalo selecionado. O usuário pode arrastar esses seletores, ajustando os valores mínimo e máximo do intervalo (Figura 1b). Em um *range-slider*, o próprio intervalo também pode ser arrastado pelo usuário (Figura 1c); dessa forma, novos valores para o início e fim do intervalo podem ser simultaneamente selecionados, mantendo entre esses valores a distância existente antes do arraste.

Range-sliders apresentam aplicação direta em várias situações, mas também apresentam algumas restrições. Existem sérias limitações relacionadas à quantidade de valores que esses controles representam e permitem selecionar. Este artigo tem por objetivo discutir essas limitações e apresentar algumas soluções que podem ser adotadas

para superá-las. Duas abordagens distintas são propostas: uma baseada em técnicas de distorção de espaço e outra baseada na combinação de *range-sliders*.

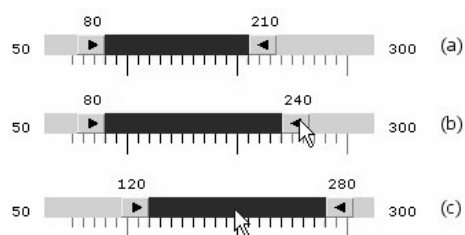


Figura 1 – Exemplo de *range-slider*: (a) parado; (b) seletor direito sendo arrastado; (c) intervalo sendo arrastado.

Este artigo está organizado da seguinte forma: na próxima seção são apresentadas algumas aplicações de *range-sliders*, bem como a limitação referida anteriormente. As duas seções subsequentes apresentam técnicas de distorção e de combinação de *range-sliders* que visam superar o problema citado. Por fim, a última seção apresenta algumas considerações finais.

APLICAÇÕES E LIMITAÇÕES

Várias aplicações da área de Visualização de Informação utilizam *range-sliders* para possibilitar que o usuário selecione intervalos de valores. Uma delas é o programa HomeFinder [1], que auxilia a escolha de casas para compra em um mapa de Washington. À medida que o usuário seleciona em um *range-slider* um intervalo de preços, somente as casas com preços nesse intervalo aparecem no mapa. Da mesma forma, outro *range-slider* permite escolher um intervalo contendo a quantidade de quartos desejada.

Range-sliders também podem ser utilizados para a seleção de intervalos de tempo. Isso é feito no LifeLines [5], um programa que apresenta graficamente o histórico da vida de uma pessoa sob algum aspecto específico (por exemplo, seus registros médicos contendo consultas, diagnósticos, remédios indicados, etc.). Nesse programa, o usuário pode usar um *range-slider* para informar ao sistema qual intervalo de tempo deve ser considerado na exibição do histórico.

Convém ressaltar, no entanto, que nesses exemplos os *range-sliders* representam seqüências com uma quantidade pequena de valores. No HomeFinder, o número de quartos varia de 1 a 7, enquanto o preço das casas varia de \$50 a \$500 em intervalos de \$10 (ou seja, 46 valores). Já no LifeLines, o *range-slider* permite escolher intervalos de dias em uma faixa de 10 meses, o que limita a quantidade de valores a 300. Assim, em nenhum desses casos existe a necessidade de se representar grandes quantidades de valores.

Essa necessidade surgiu ao tentar aplicar técnicas de Visualização de Informação ao ambiente de educação a distância TelEduc [6]. Esse ambiente apresenta uma ferramenta chamada Acessos, responsável por registrar e exibir o momento em que cada participante de um curso do ambiente acessou cada ferramenta que compõe o sistema. Para facilitar a tarefa de selecionar parâmetros de consulta para essa ferramenta, estudam-se formas para o usuário entrar esses parâmetros via apontamento. Utilizando ma-

nipulação direta para definir participantes, ferramentas e intervalo de tempo a serem considerados, o usuário pode analisar mais rapidamente os acessos feitos no ambiente e, então, tomar decisões relacionadas a eles e ao curso em si.

Uma das formas escolhidas para possibilitar essa seleção foi permitir que o usuário utilizasse um *range-slider* para escolher o intervalo de tempo cujos acessos devem ser exibidos. Em Acessos, assim como no LifeLines, é importante que o usuário tenha visões tanto genéricas quanto específicas. Visões genéricas (como a análise de meses de um curso) permitem fazer uma análise global, durante ou após o curso, detectando momentos críticos ou falhas pedagógicas relacionadas à ausência de acessos em uma ou mais ferramentas. Por sua vez, analisar os acessos de alguns dias ou das últimas horas possibilita acompanhar de perto os passos dos participantes e detectar problemas mais específicos.

Para isso, contudo, o *range-slider* de Acessos precisa representar uma quantidade de valores muito maior que o LifeLines; afinal, um período de apenas 3 meses contém aproximadamente 90 dias, o que equivale a 2160 horas ou 129600 minutos. Nesse caso, o *range-slider* se depara com uma situação para a qual não foi planejado inicialmente: existem mais valores para representar do que pontos para representá-los na largura do *slider* (ou altura, se o *slider* for vertical). Supondo, por exemplo, que a largura de um *range-slider* seja de 1024 pixels (a largura máxima em uma tela de resolução 1024 por 768), e que se deseje representar as 2160 horas referenciadas anteriormente. Nessa situação, cada pixel do controle passa a representar um intervalo de aproximadamente 2 horas.

Com isso, dois problemas principais são levantados. Primeiramente, torna-se impossível selecionar intervalos menores que 2 horas. Nessa configuração do *range-slider*, é impossível selecionar para análise uma única hora ou um intervalo de 10 minutos, por exemplo. Em segundo lugar, é necessário ter um controle muito preciso do mouse para selecionar o intervalo exato desejado, o que pode não ser fácil para muitos usuários, em especial para aqueles com alguma dificuldade motora ou visual. Essa precisão não era necessária quando se tinha uma pequena quantidade de valores representada no *range-slider*, porque cada valor possível era representado por uma razoável quantidade de pixels. Agora que a situação é exatamente o oposto, o usuário é obrigado a arrastar os seletores com precisão, já que cada pixel a mais ou a menos representa uma grande diferença no intervalo escolhido.

Nas próximas seções, são apresentadas duas formas alternativas de se contornar os problemas apresentados. A primeira delas se relaciona ao uso de técnicas de distorção que modificam a relação (atualmente linear) entre pixels e valores no *range-slider*, de acordo com o intervalo selecionado. A segunda consiste em utilizar dois ou mais *range-sliders* combinados, permitindo ajustes finos do intervalo escolhido.

DISTORÇÃO EM RANGE-SLIDERS

Conforme explicado anteriormente, a existência de vários pixels representando um único valor no *range-slider* auxilia a seleção desse valor, à medida que elimina do usuá-

rio a obrigação de escolher o pixel correto ao arrastar um seletor. Assim, quanto maior for a relação pixels/valor, mais fácil se torna escolher um valor do *slider*. Para permitir a seleção de quaisquer valores representados, essa relação deve ser maior ou igual a 1.

Porém, se esse controle tiver que representar mais valores do que a quantidade de pixels de seu tamanho, torna-se impossível manter constante e ao mesmo tempo maior ou igual a 1 a relação pixels/valor em todo o *slider*. Supondo que essa relação fosse constante, a limitação de tamanho do controle obrigaria a divisão pixels/valor a ser fracionária; ou seja, um único pixel representaria mais de um valor, o que impediria a seleção de determinados valores. Por outro lado, se a relação fosse maior ou igual a 1 ao longo de todo o *slider*, o tamanho limite do controle seria ultrapassado, o que também não se deseja.

Como forma de contornar a situação, propõe-se que a relação pixels/valor *varie* ao longo do *slider*, de acordo com o intervalo selecionado. Em especial, na região do intervalo selecionado essa relação deve ser constante e maior ou igual a 1. À medida que os valores se afastam desse intervalo, essa relação deve diminuir gradualmente, de modo a garantir a manutenção do tamanho limite do *range-slider* (Figura 2). Ao alterar essa relação, ocorre na verdade uma distorção do espaço utilizado pelo *slider* para representar valores. A região do intervalo selecionado e de suas proximidades aparece ampliada, como se uma lupa tivesse sido colocada sobre ela (Figura 3). Nessa região de foco, o *slider* detalha a escala representada, mostrando valores de ordens de grandeza menores que os mostrados fora dessa região. Na parte externa, é mantida uma escala menos detalhada, que serve de contexto para o usuário e mostra apenas valores de maiores ordens de grandeza.

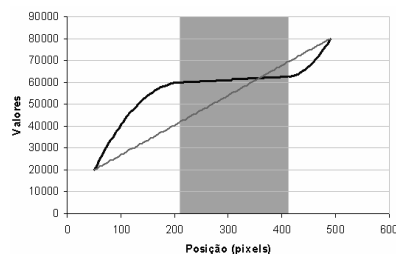


Figura 2 – Gráfico representando a relação entre pixels e valor. A linha reta representa a relação linear de um *range-slider* normal. A curva em S invertido representa a distorção proposta. A região destacada em cinza indica o intervalo selecionado.



Figura 3 – *Range-slider* apresentando a distorção mostrada na figura anterior.

A distorção aplicada apresenta algumas características importantes. Inicialmente, ela somente se aplica quando o intervalo selecionado tiver menos da metade da quantidade de valores do *slider*. Além disso, quando a distorção está aplicada, o tamanho do intervalo selecionado deve ser igual à metade do tamanho original do *slider* (como na Figura 3). Esse espaço fica reservado para o foco do usuário, enquanto a outra metade representa os valores fora do limite do intervalo (contexto).

Também foi importante definir onde a distorção deve posicionar o intervalo no *range-slider*, pois essa posição dá ao usuário uma idéia aproximada da quantidade de valores fora do intervalo selecionado. Basicamente, os espaços reservados à esquerda e à direita do intervalo no *range-slider* são proporcionais às quantidades de valores à esquerda e à direita do intervalo selecionado.

Como caso especial, quando o intervalo selecionado possui menos de 5 valores, seu tamanho em pixels passa a ficar menor, até que se torna nulo quando os dois seletores estão exatamente sobre o mesmo valor.

Interação

A interação do usuário com esse novo *range-slider* apresenta características importantes a serem discutidas, em especial sobre o comportamento da distorção durante o arraste de elementos do *slider*. Por exemplo, ao se arrastar um dos seus seletores, a distorção existente antes do arraste é mantida até que o seletor seja solto (*mouse-up*). A escolha desse comportamento justifica-se: caso a distorção não fosse mantida e acompanhasse os novos valores escolhidos pelo usuário durante o arraste, seria difícil conseguir selecionar um determinado valor, pois a escala que lhe serve de referência (sempre visível abaixo do *slider*) estaria sendo modificada a todo o momento.

Assim, após o usuário selecionar um novo valor, uma nova distorção é calculada e apresentada a ele. Contudo, uma distorção somente é efetiva quando o usuário consegue perceber como a estrutura original foi distorcida [3]. Assim, para mostrar a ele essa distorção e mantê-lo a par do que aconteceu com a escala, a escala antiga é gradualmente transformada na nova por uma animação de aproximadamente 0,8 s. Esse período de tempo foi escolhido para que o usuário possa acompanhar a animação, entendendo a distorção efetuada, e também para que sua interação com o sistema não se torne lenta. É preciso, contudo, verificar se pequenos ajustes nesse período são necessários, procurando não superar a barreira de 1 s (conhecida como "tempo de resposta não preparada" do usuário [4]).

Já no arraste do intervalo selecionado, não foi possível manter a distorção anterior ao arraste. Se ela fosse mantida, ao tentar arrastar o intervalo, o usuário veria esse intervalo ficar com menos pixels na tela, embora representasse a mesma quantidade de valores; isso poderia dar a falsa impressão de que a quantidade selecionada de valores teria diminuído. Para evitar esse problema, foi estabelecido que o intervalo deve manter a mesma quantidade de pixels e de valores de antes do arraste. Com isso, a distorção varia durante o arraste de acordo com a nova posição do intervalo no *slider*.

COMBINAÇÃO DE RANGE-SLIDERS

Outra alternativa para permitir que o usuário selecione um intervalo em uma grande quantidade de valores é o uso de uma combinação de *range-sliders* interdependentes. Como exemplo, suponha a existência de dois *range-sliders*, como na Figura 4. O primeiro deles permite escolher valores até uma ordem de grandeza mínima, na qual esbarra com o problema de representar mais de um valor por pixel. Tendo sido selecionado um intervalo nesse primeiro *slider*, os valores máximo e mínimo escolhidos se tornam instantaneamente os valores limites do segundo

slider. Dentro desses novos limites, o usuário pode refinar o intervalo inicial escolhido, contando para isso com a representação de valores de menor ordem de grandeza. Ou seja, o segundo *slider* funciona como uma ampliação do intervalo escolhido no primeiro, permitindo ajustes finos no intervalo previamente selecionado.

Comparando as duas alternativas apresentadas, verifica-se que a combinação de *sliders* usa mais espaço e obriga o foco do usuário a alternar entre os dois (ou mais) *sliders* que a compõem, o que pode causar uma sobrecarga cognitiva. A solução baseada em distorção, por sua vez, pode apresentar sobrecarga no entendimento da distorção (pela inconsistência do tamanho do intervalo com relação à quantidade de valores selecionados), mas isso tende a acontecer apenas nos usos iniciais do controle. Além disso, o controle com distorção ocupa o mesmo espaço de um *slider* comum e o foco de atenção permanece parado, o que leva a acreditar que essa seja a alternativa com melhor usabilidade dentre as apresentadas neste artigo.

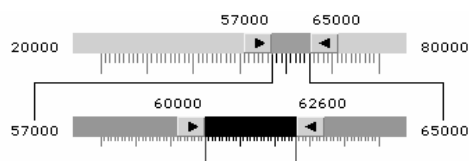


Figura 4 – Combinação de dois *range-sliders*.

CONSIDERAÇÕES FINAIS

Este artigo apresentou duas variações de *range-sliders* que podem ser utilizadas para a representação de grandes quantidades de valores a serem selecionados pelo usuário. Testes de usabilidade serão efetuados para avaliar essas alternativas e escolher qual delas será incorporada à ferramenta Acessos do ambiente TelEduc. Pretende-se estender os protótipos desses controles (Figura 3 e Figura 4) para trabalhar com datas e horas, possibilitando que eles sejam usados para selecionar via manipulação direta o intervalo de tempo cujos acessos devem ser exibidos.

Além disso, o estudo de como representar grandes faixas de valores em um espaço limitado pode ter utilidade para

sistemas de computadores de mão (*hand-helds*), nos quais o tamanho da tela é extremamente limitado e a interação acontece principalmente por apontamento.

REFERÊNCIAS

1. Ahlberg, C., Shneiderman, B. Visual Information Seeking: Tight Coupling of Dynamic Query Filters with Starfield Displays. *Proc. CHI'94, ACM Conference on Human Factors in Computing Systems*, 1994, 313-317.
2. Ahlberg, C., Shneiderman, B. Alphaslider: A Rapid and Compact Selector. *Proc. CHI'94, ACM Conference on Human Factors in Computing Systems*, 1994.
3. Card, S.K., Mackinlay, J.D. e Shneiderman, B. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufman Publishers, 1999.
4. Carr, D. A., Jog, N., Kumar, H. P., Teittinen, M., Ahlberg, C. *Using Interaction Object Graphs to Specify and Develop Graphical Widgets*. Technical Report CS-TR-3344, University of Maryland, 1994.
5. Plaisant, C., Mushlin, R., Snyder, A., Li, J., Heller, D., Shneiderman, B. LifeLines: Using Visualization to Enhance Navigation and Analysis of Patient Records. In: Bederson, B., Shneiderman, B. (eds.), *The Craft of Information Visualization – readings and reflections*, Morgan Kaufmann Publishers, 2003.
6. Rocha, H.V. da. O ambiente TelEduc para Educação a Distância Baseada na Web: Princípios, Funcionalidades e Perspectivas de Desenvolvimento. In: Moraes, M.C. (org.), *Educação a Distância: Fundamentos e Práticas* (11), 197-212, Nied/Unicamp, 2002. http://teleduc.nied.unicamp.br/pagina/publicacoes/8_Cap11_heloisa.pdf (23/Maio/2004).
7. Shneiderman, B. Dynamic Queries for Visual Information Seeking. *IEEE Software*, 11 (6), 70-77, 1994. Também em [3].
8. Spence, R. *Information Visualization*. ACM Press, 2001.

Suporte empírico e neurolingüístico em hipermídia adaptativa

Adriana Barbosa Santos

UNESP
São José do Rio Preto - SP
adriana@ibilce.unesp.br

Elcimar Alves Socorro

UNESP
São José do Rio Preto - SP
elcimarsocorro@yahoo.com.br

RESUMO

Um dos problemas mais criticados dos sistemas de hipermídia adaptativa é o fato de que o processo de adaptação necessita de um número razoável de interações dos usuários para que haja um nível adaptativo satisfatório. Ao mesmo tempo, as características para definição dos perfis de usuário nesse processo consideram fatores dessa interação, como documentos acessados, seqüência de acesso e tempo de permanência nos documentos. Esses elementos delimitadores de perfis não são compatíveis com os delimitadores tradicionais, como os usados em marketing ou psicologia, que são, por exemplo, faixa etária, sexo ou nível cultural. O objetivo desse artigo é mostrar um modelo de hipermídia adaptativa com suporte à informação neurolingüística no início do processo adaptativo, juntamente, com aspectos que permitem a definição dos perfis de usuário de acordo com os delimitadores de perfis tradicionais já utilizados em marketing e psicologia e bem difundidos nos estudos de operações comerciais na atualidade.

ABSTRACT

One of the most criticized problems in adaptative hypermedia systems is the necessity of a reasonable number of user interactions in adaptation process to get a satisfactory adaptative level. At the same time, the characteristics for definition of user profiles in this process consider factors of this interaction as accessed documents, sequence of access and connected time to documents. These profile delimiters elements are not compatible with the traditional delimiters used in marketing or psychology, such as, for example, age, gender or cultural level. The aim of this article is to present an adaptative hypermedia model with neurolinguistics information support at the beginning of the adaptative process. It is also presented aspects related with user profile definition, in accordance with traditional profile delimiters already used in marketing and psychology and very diffused in commercial transactions nowadays.

Palavras Chave

Hipermídia adaptativa, perfis de usuário, neurolingüística

INTRODUÇÃO

Com a crescente disponibilidade de informação nas diversas formas de mídia, a idéia de descobrir a identidade do usuário para melhor descrever suas preferências e interesses visando direcionar o conteúdo do documento apresentado a ele, aumentando sua satisfação, tornou-se um objetivo estratégico imperativo das empresas que atuam no comércio eletrônico.

Adaptar o conteúdo de documentos de hipermídia usando perfis de usuário deu origem a um novo conceito chamado de hipermídia adaptativa. Segundo Brusilovsky [1] um sistema de hipermídia adaptativa é todo sistema de hipermídia que reflita algumas características de seus diferentes usuários em modelos e aplique-os na adaptação de diversos aspectos visíveis do sistema às necessidades e desejos de cada usuário.

Existem poucas referências na literatura que tratam da adaptação de documentos hipermídia tendo como foco o comércio eletrônico (*e-Commerce*). O que é certo é que os modelos empregados atualmente na definição dos perfis de usuário nos sistemas adaptativos não são coerentes com os modelos de perfis utilizados em outras áreas como Marketing [2] e Psicologia, já que os sistemas adaptativos atuais têm seus perfis delimitados pela interação do usuário nos elementos dos documentos, que podem ser, por exemplo, os assuntos de interesse do usuário, o tempo que ele permanece em algum documento ou a seqüência de acesso a esses documentos [1][5][6]. Já os delimitadores de perfis utilizados nos estudos das áreas de Marketing e Psicologia consideram, entre outros aspectos, características demográficas e neurolingüísticas¹ [2][7]. Essa diferença acaba dificultando a aplicação dos modelos de hipermídia adaptativa atuais no comércio eletrônico, já que do ponto de vista de estratégias de marketing e de controle de mercados esses modelos usam estratégias arbitrárias para definir o perfil do consumidor *online*.

Esse artigo apresenta um modelo de adaptação de documentos de hipermídia denominado ERBAHM (Empiric-Random-Based Adaptative Hypermedia Model), cuja proposta é permitir a utilização de delimitadores de perfis de usuário tendo como base uma pesquisa empírica [8] que quantifique a relação entre as características de elementos dos documentos de hipermídia com os componentes desse perfil do usuário.

¹ Estão relacionadas às vivências pessoais e experiência no ramo a que se refere o perfil em questão

O processo de adaptação é feito usando geradores pseudo-aleatórios que representam bem o elemento incógnito sobre o perfil do usuário quando a crença em uma determinada característica não é completa; o que define regras de adaptação. A figura 1 ilustra a estrutura do modelo ERBAHM como uma adaptação do modelo sugerido por Palazzo[3] já com suporte à informação empírica e neurolinguística, tanto no conjunto de dados sobre a identidade do usuário, como nas regras de modelagem. Observe que as regras de modelagem do usuário representam o conjunto de informações relevantes em um perfil. No caso dos sistemas adaptativos convencionais essas regras se referem às características navegacionais. Em ERBAHM essas regras estão associadas também aos delimitadores de perfis de outras áreas.

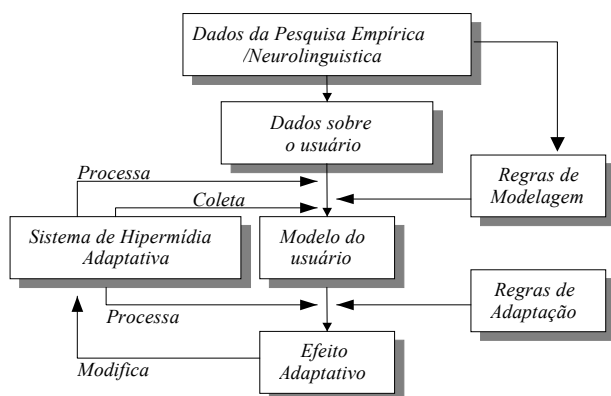


Figura 1. Estrutura do modelo adaptativo com suporte empírico/neurolinguístico

METODOLOGIA

As suposições a respeito da identidade do usuário no momento da adaptação de um documento são chamadas de crenças. O nível de relevância de uma crença é chamado de grau de crença. Quanto maior é a evidência de que uma crença seja verdadeira maior é o grau dessa crença.

A estratégia adaptativa básica no modelo ERBAHM considera o fato de que a identidade do usuário é sempre desconhecida, mas que pode haver indícios sobre probabilidades associadas às características do perfil do usuário, as quais são obtidas à medida que ele interage com o sistema. Considerando cada interação como indicio de satisfação, as ações do usuário ensinam o sistema quais são as melhores formas de exibir um documento de hiperídia para cada usuário. Assim, uma vez que o sistema se adapta às preferências do usuário, há um aumento de sua satisfação e os resultados são otimizados.

Suponha que fatores neurolinguísticos, como questões sociais e culturais, indicam que o maior número de pessoas que se interessa por carros é do sexo masculino. Esse indicio pode ser quantificado com uma pesquisa empírica que evidencie o fato de que 90% das pessoas que se interessa por carros são do sexo masculino. Se houver um clique em uma hiperligação cujo assunto seja carros, é possível redistribuir as crenças sobre o sexo do usuário, levando em conta esse dado probabilístico. Então, o sistema infere que há 90% de chance de que o usuário seja do sexo masculino e adapta a apresentação

do documento, segundo as informações da pesquisa empírica.

A arquitetura ERBAHM

Tendo como base os indícios neurolinguísticos quantificados por levantamentos empíricos e evidenciados pela ação do usuário em elementos do documento de hiperídia, a arquitetura da implementação do modelo ERBAHM é proposta de acordo com a figura 2.

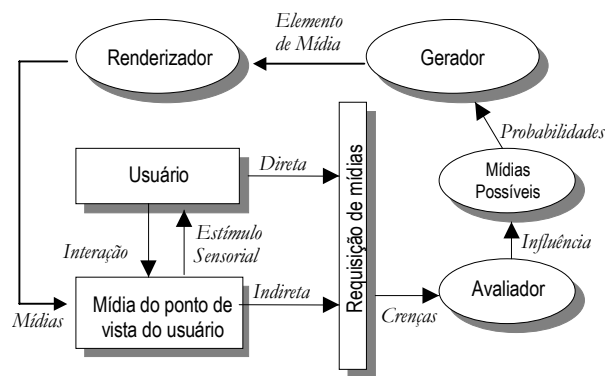


Figura 2. Arquitetura modular de ERBAHM

Modelagem do usuário

As características relevantes sobre o perfil do usuário variam de acordo com a situação e contexto onde o sistema deve estar inserido, tornando a abordagem mais genérica. Assim, podem-se introduzir informações sobre os perfis de usuário baseadas em outras fontes de dados, desde que estas reflitam os perfis no sistema de forma consistente.

Logo, se um sistema de hiperídia é representado por uma loja virtual de cosméticos, cujas características de interesse para adaptação de conteúdo dos documentos sejam cor da pele e tipo de cabelo, por exemplo, essas características devem existir no perfil do usuário. Todavia, outros critérios muito mais subjetivos podem ser utilizados, inclusive os já utilizados pela maioria dos sistemas adaptativos atuais.

A partir do modelo ERBAHM cada usuário é entendido como um conjunto de atributos, cada atributo possui um conjunto de valores possíveis, e para cada usuário diferente pode haver crenças diferentemente distribuídas dentro do conjunto de valores possíveis de um atributo. Sendo assim, cada usuário U possui n atributos A_i , e cada atributo possui um conjunto de q valores possíveis a_{ij} . A equação 1 define algebricamente o usuário.

$$U = \bigcup_{i=1}^n A_i, A_i = (a_{i1}, a_{i2} \dots, a_{iq}) \quad (1)$$

O grau de crença em um determinado valor possível do atributo é definido por uma função f que é denominada função de interferência empírica/neurolinguística, a qual associa a cada valor possível a_{ij} de um atributo A_i um valor real no intervalo $[0,1]$, sujeita à restrição dada em (2).

$$\sum_{j=1}^q f(a_{ij}) = 1 \quad (2)$$

Modelagem das mídias

O modelo dos elementos (mídias) adaptáveis do documento de hipermídia é semelhante ao modelo do usuário, isto é, cada elemento é definido como um conjunto de atributos que o descrevem de forma suficiente, só que nesse caso, com um elemento diferenciador a mais. Tipo de mídia é o elemento diferenciador necessário para classificar diferentes mídias segundo sua forma de exibição. Os atributos que definem cada tipo de mídia devem ser unicamente os atributos que se tem interesse em adaptar. Os demais atributos devem estar subentendidos segundo o tipo de mídia.

Assim, em um documento D onde existem k tipos de mídia T_i , onde cada uma delas é definida por um conjunto de p_i atributos do tipo M_{is} , sendo que cada M_{is} possui t_{is} valores possíveis do tipo c . Um elemento adaptável do documento é definido de acordo com a equação 3 e o documento completo pela equação 4.

$$T_i = \bigcup_{s=1}^{p_i} M_{is}; M_{is} = (m_{is1}, m_{is2} \dots m_{ist_{is}}) \quad (3)$$

$$D = \bigcup_{i=1}^k T_i, i = (1, 2 \dots k) \quad (4)$$

Aqui se considera que o sistema tem como ponto de partida algum conhecimento sobre as distribuições das preferências do usuário a respeito das características das mídias. Deste modo, pode-se associar a cada valor m_{isb} uma função g que leva a um valor real positivo, de forma que cada $g(m_{isb})$ seja proporcional à distribuição de preferências de cada m_{isb} . Por exemplo, se o levantamento empírico evidencia 25% de preferência dos usuários por imagens em tons de cinza, e 75% de preferência por imagens coloridas, $g(m_{is1})=50$ e $g(m_{is2})=150$ são dois valores possíveis. Vale ressaltar, que o dimensionamento de $g(m_{isb})$ é importante no momento da definição da velocidade de adaptação do sistema.

A mídia para o usuário

É possível definir o elemento do modelo que une usuários e mídias. Suponha que cada mídia é composta por um conjunto de atributos, e cada atributo possui um conjunto de valores possíveis. Considere ainda que a função $g(m_{isb})$ é usada como indicio empírico das preferências do usuário. Um gerador pseudo-aleatório é usado de forma que ele escolha algum dos valores possíveis para um atributo, mantendo as proporções das distribuições empíricas. No caso de não existir qualquer indicio empírico, é associado a $g(m_{isb})$ um valor proporcional correspondente ao valor obtido pela distribuição de probabilidades uniforme para cada valor possível do atributo. Sendo assim, se 75% das preferências é por imagens coloridas, então o gerador deve escolher aleatoriamente um desses valores de forma que tenha 75% de probabilidade de sortear o atributo para imagem colorida e 25% de probabilidade de sortear o atributo para

imagem cinza, cada vez que um usuário ou documento fizer uma requisição de mídia. Depois que cada atributo tem seu valor sorteado o conjunto completo de valores sofre uma mudança de coordenadas e é exibido para um espaço métrico que estimule o usuário sensorialmente. Essa mudança de coordenadas é de responsabilidade do módulo renderizador.

As ações do usuário

A principal hipótese sobre as ações do usuário considera que se o usuário interage com a mídia, então ela o agrada de alguma forma. No entanto, em situações mais complexas, o modelo de ações pode considerar descontentamento. Mas, nesse contexto, a hipótese mais trivial é a mais indicada.

Se usuários interagem com as mídias e isso é indicio de sua preferência e de sua identidade, duas suposições a respeito do usuário são possíveis quando ele atua no sistema:

1. Cada valor do atributo pode evidenciar crenças diferentes sobre a identidade do usuário.
2. O usuário prefere os valores de atributos da mídia onde ele interage.

Na primeira situação, uma função $f(a_{ij})$ associa a cada valor m_{isb} de um atributo de mídia uma nova distribuição de graus de crenças sobre a identidade do usuário.

Na segunda situação, um conjunto denominado de avaliador é usado para estabelecer a relação entre perfis e preferências. O conjunto P de todos os perfis possíveis é mostrado na equação 5 e a descrição formal do avaliador H está expressa na equação 6.

$$P = A_1 \times A_2 \dots \times A_q \quad (5)$$

$$H = \bigcup_{i=1}^k P \times T_i \quad (6)$$

O papel do avaliador é armazenar as experiências dos usuários no sistema, quantificar os indícios de identidade e alterar o sistema proporcionalmente a cada crença da identidade do usuário.

No caso do avaliador, a função $g(m_{isb})$ dos elementos da relação na forma m_{isb} é incrementado a cada interação do usuário. O tamanho do incremento é o produto dos graus de crença $f(a_{ij})$, associados ao valor m_{isb} por uma constante real qualquer, cujo tamanho também interfere na velocidade de adaptação do sistema. O redimensionamento de $g(m_{isb})$ só ocorre se o valor m_{isb} do atributo existe na mídia em que o usuário interagiu. O incremento é uma forma de alterar as probabilidades de exibição de valores de atributos de mídia onde houve interação. A figura 3 ilustra a proposta ERBAHM para a adaptação de conteúdo baseada nas escolhas do usuário. Observe que as crenças sobre a característica N do perfil do usuário, cujos valores possíveis são X, Y e Z, se alteram depois que o usuário interage no sistema. A escolha do elemento F é responsável pela redistribuição das crenças e está fundamentada na distribuição empírica das características da identidade do usuário de acordo

com as preferências pelos elementos do sistema. O documento inicial é adaptado de acordo com as distribuições empíricas de X, Y e Z na amostra populacional e representam o conhecimento inicial do sistema, sendo que a inferência neurolingüística indica a validade da relação entre a identidade do usuário e a interação.

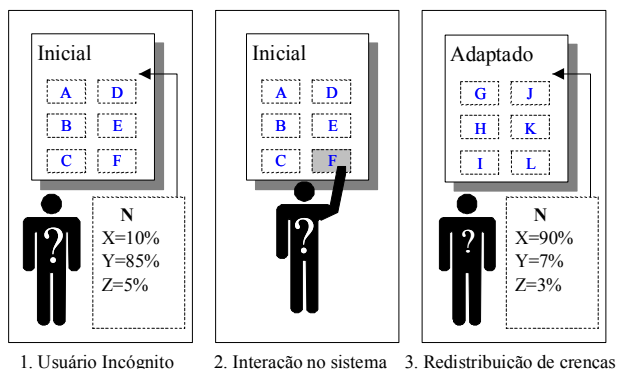


Figura 3. Alteração das crenças na identidade do usuário

São os valores dos $g(m_{isb})$ de H que são utilizados no sistema durante o processo de sorteio aleatório das características de mídias que serão exibidas para o usuário. Novamente, considerando os valores proporcionalmente às novas crenças evidenciadas pela escolha de m_{isb} .

Andamento do projeto

O modelo ERBAHM já possui uma implementação simplificada como exemplo de aplicação. É uma página de Internet criada com o propósito de fornecer mais detalhes sobre o modelo.

O desenvolvimento do modelo se deu nos últimos seis meses como uma necessidade de relacionar conceitos já difundidos nas áreas de Marketing, Neurolingüística e Estatística e associá-los com a adaptação de documentos de hipermídia, a fim de proporcionar um aumento da satisfação do cliente em lojas virtuais.

A continuidade do projeto está relacionada à definição dos delimitadores de perfis para relações empresa-empresa na Internet (*e-Business*) e ao uso de métricas que diferenciem as ações do usuário, para que possam ser usadas como critérios de controle de qualidade, de análise de desempenho e de usabilidade de sistemas virtuais.

A literatura recente destaca com maior ênfase os projetos de sistemas adaptativos para ensino e educação à distância [4]. As estratégias de adaptação são bem variadas, mas os perfis sempre são definidos por delimitadores vindos da interação do usuário com o sistema [5][6].

RESULTADOS

O modelo ERBAHM apresenta suporte nativo a informação neurolingüística devidamente quantificada por um levantamento empírico. A aplicabilidade do modelo em sistemas de comércio eletrônico parece ser bem viável, sobretudo pela possibilidade de uso dos perfis já utilizados nos estudos de marketing.

Da mesma forma, como os delimitadores de perfis de aluno usados em pedagogia são bem particulares, a idéia de utilizar o modelo ERBAHM em ferramentas de ensino parece interessante, diante da proposta de se incorporar informações quantificáveis, normalmente usadas no ensino real, para melhor caracterizar o perfil do aluno.

O suporte empírico dá ao sistema a capacidade de armazenar informação real como ferramenta da adaptabilidade, mesmo no processo inicial do sistema. Dessa forma se, desde a primeira interação de usuário, o nível de adaptabilidade já considere conhecimento agregado de outras áreas, é de se esperar que o processo adaptativo seja acelerado. Como o custo de uma pesquisa empírica costuma ser alto, a idéia de agregar esse tipo de informação em um sistema de hipermídia adaptativa pode ser inviável em alguns casos. Todavia, na sua concepção, o modelo ERBAHM considera informação existente. Desse modo, a estratégia de adaptação é recomendada em casos onde o levantamento empírico seja viável. Um outro ponto que vale ressaltar é que a redistribuição das crenças na identidade do usuário considera hipóteses neurolingüísticas para conectá-las às características das mídias, o que implica na necessidade de um estudo cuidadoso sobre tais hipóteses na entrada das informações no sistema.

REFERÊNCIAS

1. Brusilovsky, P. Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*(1998), 1-43.
2. Kotler, P. *Administração de Marketing*. São Paulo, Atlas, 1998.
3. Palazzo, L. A. *Sistemas de hipermídia adaptativa*. Grupo de Pesquisa em Inteligência Artificial, Universidade Católica de Pelotas, 2003.
4. Brusilovsky, P. *Adaptive and intelligent technologies for web-based education*, *Künstliche Intelligenz*(1999).
5. Kobsa, A. *Generic user modeling systems*. *User Modeling and User Adapted Interaction*(2001), 49-63.
6. Vassileva, J. *A task-centered approach for user modelling in a hypermedia office documentation system*, *User Modeling and User Adapted Interaction*(1996), 87-129.
7. Knight, S. *Introdução à Neurolingüística*, São Paulo, Nobel, 2001.
8. Dencker, A. F. M.; Via, S. C. *Pesquisa empírica em ciências humanas: com ênfase em comunicação*, São Paulo, Futura, 2001.

Aesthetics and perceived usability of VLEs: preliminary results

R. Parizotto-Ribeiro and Dr. N. Hammond

Department of Psychology, University of York,
York, YO10 5DD, UK
R.Ribeiro@, N.Hammond@psych.york.ac.uk
+44 (1904) 432869

J. Mansano and C. Cziulik, Ph.D.

Department of Mechanics, CEFET-PR
80.230-901, Curitiba, Brazil
mansano@, cziulik@cefetpr.br
+55 (41) 3104665

RESUMO

Este artigo apresenta os resultados preliminares de uma pesquisa sobre Ambientes Virtuais de Aprendizagem (AVAs), focando nos seus aspectos estéticos, com ênfase em cinco princípios de design e a usabilidade percebida destas interfaces gráficas. O estudo está baseada em experimentos realizados por [2] e [5] em Caixas Bancários Automáticos. Estes autores encontraram uma alta correlação entre estética e usabilidade percebida. O presente trabalho objetiva certificar se esta correlação também se aplica à diferentes tipos de interfaces computacionais como AVAs. Primeiramente, a correlação entre a avaliação da estética e da usabilidade percebida de um interface AVA foi calculada usando os cinco princípios de design em três diferentes conjuntos de layout de telas. Em seguida, a correlação entre as notas de estética e usabilidade percebida da interface foi calculada usando estes princípios em três diferentes conjuntos de layouts de tela.

ABSTRACT

This article presents the preliminary results of research on Virtual Learning Environments (VLEs), focusing on their aesthetic aspects, with emphasis on five design principles and the perceived usability of their graphical interfaces. The study is based on experiments performed by [2] and [5] on Automated Teller Machines (ATMs). These authors found a high correlation between aesthetics and perceived usability. The present work aims to ascertain whether this correlation also applies to different types of computer interfaces such as VLEs. Firstly, the influence of five design principles, based on [1] and [3], on screen layout design was evaluated. Next, a correlation between ratings of aesthetics and of perceived usability of a VLE computer interface was calculated using those principles on three different sets of layout screens.

Palavras Chaves

Princípios de design, estética/usabilidade percebida, AVA.

INTRODUCTION

The role of aesthetics in people's representations of artifacts has been widely documented. Probably, aesthetics is also related to appreciation of computer systems.

Nowadays, computer interfaces have much higher usability than previously (i.e. they enable the achievement of greater effectiveness and efficiency). However, the third element of usability, user satisfaction, has not been adequately taken into account.

A recent book 'Emotional Design – Why we love or (hate) everyday things' [4] gives a new perspective to products based on affective design, emphasizing their aesthetics and pleasure of use. The theory proposed by [4] presents three levels of emotional design: visceral, behavioural and reflective.

Visceral design is concerned with appearance and it doesn't depend on cultural aspects. Behavioural design deals with pleasure and effectiveness of use. Reflective design considers the message, meaning of the product and it is culturally dependent.

This theory can also be applied to computer interfaces, especially to VLEs, where the person may need additional motivation to be fully engaged on a distance learning course.

In the light of recent studies on perceived usability and aesthetics [6], use of design principles to achieve usability [3] and Norman's arguments and findings reported in [4], this research has identified strong support to its main hypotheses that aesthetically pleasing interfaces have a positive effect on user satisfaction.

This paper aims to further explore the role that interface aesthetics plays in VLE system usability and acceptability.

RESEARCH APPROACH

In order to achieve the research goals it has been necessary to identify:

- which design principles are relevant for computer interface aesthetics;
- how aesthetics impact the usability of the interface;
- what graphical aspects can influence user satisfaction, engagement and enjoyment.

A pilot experiment with five participants was performed in a controlled environment to check, improve and tune the main set of experiments.

Finally, a model that can assess the correlation between aesthetics and usability in the context of VLEs will be developed.

VI Simpósio sobre Fatores Humanos em Sistemas Computacionais — Mediando e Transformando o Cotidiano. Hotel Bourbon, Curitiba, 17 a 20 de outubro de 2004. UFPR, CEIHC—SBC.

RELATED WORK

Researches on aesthetics aspects

The literature reports the first study about aesthetics in the field of Human Computer Interface (HCI) conducted by [2]. This study showed evidence of the importance of aesthetics in screen layout and apparent usability of a computerized system. Later on, a similar study was performed by [5] in Israel, producing similar findings to [2], which was conducted in Japan. Tractinsky has run three experiments to replicate and validate Kurosu and Kashimura's study in a different cultural setting, concerning the relationship between aesthetics and apparent usability. Ratings of aesthetic quality were found to correlate strongly with the system's perceived usability, both before and after the interaction [5]. This study suggests that the user may be strongly affected by the aesthetic aspect of the interface, even when evaluating its functional aspects. He also found that aesthetics and apparent usability are related in both cultures, given stronger support to the hypotheses that aesthetics and usability are correlated in a positive way.

Aesthetics of layout composition

Since the early twentieth century, the principles of visual communication developed by Gestalt Psychologists have been discussed and re-emphasized [1]. Nowadays, their importance and relevance to user interface design is more widely accepted and understood. Based on these principles, [3] have developed fourteen aesthetic measures (balance, equilibrium, symmetry, sequence, cohesion, unity, proportion, simplicity, density, regularity, homogeneity, rhythm, order and complexity) for assessing graphic displays completeness. Their empirical studies have suggested that these measures may help gain attention and build confidence in using computer systems.

In the present work, these aesthetic measures have been combined with relevant design principles that are most accepted by designers' community and widely used for the development of their practical work. The investigation has resulted in five design principles (unity, proportion, homogeneity, balance and rhythm) that seem to suit the particular needs of a screen layout of a computer interface for Virtual Learning Environments.

Unity refers to the extent to which a group of visual elements are perceived as all part of one single piece. Proportion is defined as the comparative relationship between the dimensions of the screen components. Homogeneity is a measure of how evenly the objects are distributed throughout the screen. Balance can be defined as the distribution of optical weight in a picture. Rhythm refers to regular patterns of change in the elements.

DESIGN OF THE SOFTWARE APPLICATION

In order to evaluate the screen layouts which simulated a VLE, a software application using Authorware, authoring software from Macromedia, Inc., has been developed. This was necessary because Authorware is prone to deal with images and text, allowing programming at high level and offering accessible maintenance to a non-expert user on computer programming languages. The Authorware suite presents an interactive method of programming,

using flowchart, icons and windows (oriented to objects as well as to events).

The flowchart (Figure 1) developed for this software application consisted of seven modules, as follows:



Figure 1. The modules.

- setups: initial configurations that mainly indicate the variables to be used in the application. This initial step records information from each participant.
- set 0 – screens: evaluation of the design principles by presenting five screens, with two layouts, one showing the principle well applied and the other badly applied. The participant has to choose the layout he or she perceives as being the best.
- set 1 - home: showing examples of the first page that the user sees when logging on the VLE (Figure 2).
- set 2 - content: showing examples of the content material pages.
- set 3 - email: showing examples of the most used communication tool.
- 12 screens: presentation of the same screens where the participants have been asked to evaluate the perceived usability. The screens were presented individually following the same procedure employed to evaluate the aesthetics aspects.

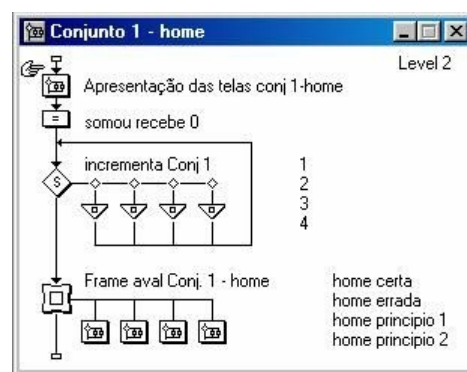


Figure 2. Programming structure for evaluation of home set.

- 3 screens: in this part of the experiment participants are shown the first page of each set of screens, aiming at evaluating the hedonic attributes of the environment on a printed questionnaire based on bipolar verbal anchors proposed by [7]. It also contains a thank-you screen and a password to initiate the next participant's evaluation.

When in use, sets 1, 2 and 3 were each prepared in four different layouts. However, they have the same graphic elements, colours and typography. The design principles were applied as described in the Material section.

At this stage, the participants should evaluate the aesthetic aspects of the screens. The methodology used has involved an initial presentation of each screen for two seconds (e.g. allowing participants to have a first exposure to the visual stimulus). Following this, the same screens were presented individually, allowing the participants to evaluate them. When the assessment finished, the program moved to the next screen. An evaluation module was implemented, with clickable buttons which give grades from 1 to 5.

The data gathering used a Latin Square design, a method used in psychology experiments to counterbalance any potential order effects.

The data of each evaluation, including date, starting and finishing time have been recorded in a text file.

PILOT STUDY

Graphical style

The graphical style chosen for this experiment is based on ‘cartoon drawings’ with the purpose of creating a more informal environment and clean interface using a compositive area of the golden rectangle. The elements are simple, using complimentary colors (mainly, orange and blue) to create the necessary contrast. The typography used was Verdana (black), which gives a good legibility on this kind of media.

Study Design

The pilot study was conducted with five participants to ensure that the program would function correctly and could collect their ratings about the interface and its navigation. From these preliminary results, refinements on the guidance screens have been added.

One interesting aspect observed in the last part of the experiment is that the approach to collect information (i.e. the paper questionnaire to evaluate three screens) proved not to be efficient. Three participants finished the experiment without completing the paper questionnaire, even though there has been an intermediary screen with instructions explaining about how to proceed on this specific part. This deficiency has been minimised by placing an informative sentence before the button “Continue” and having a b&w image of the respective screens printed together with the paper questionnaire.

Participants

The pilot study involved two lecturers and five members of staff of a Higher Education Institution in Brazil (CEFET-PR). All participants received financial compensation.

Material

Three set of screens that simulated specific areas of a Virtual Learning Environment elaborated. It was used a non interactive software that just allowed the participants to give grades for each of the layouts. The first set (home) intends to give the participant a general idea about the environment. The content set simulated an introduction to a Module of Photography. Finally, the e-mail set enables

the participant to have access to the most commonly-used communication tool. Each set of screens has four different pages layout:

- Page 1: follows the design principles (unity, proportion, homogeneity, balance and rhythm);
- Page 2: violates all of them;
- Page 3: complies to the first three design principles and violates the other two;
- Page 4: violates the first three and obeys the last two design principles.

Method

First, the participants received spoken information and instruction about the study in which they were about to participate. After that, they filled in the consent form (in Portuguese, their native language, and English, as part of the proceedings of the departmental Ethical Committee) and then were guided to start the experiment.

The instructions for the participants were also presented on individual screens during the experiment, orientating them to what they would be evaluating, as well as guiding them as the experiment progressed. The pilot study was carried out in a single day with one participant at a time.

Procedure

First, the participants were presented with five screens that encompass five design principles derived from [3]. Each screen shows two figures representing one of the principles, where one was followed all the principles and the other did not. Participants were instructed to choose the one they would prefer (screen A or screen B).

After that, they see an intermediary screen which they were told just to observe. Then, the first set of screens were shown so that the participants had a chance to see each of the four screens during four seconds, like a little “trailer preview”. The reason for doing this was to give the participant an opportunity to see all four screen layouts before evaluating them. After that, the screens were presented again and the participants had to evaluate them using a five-point Likert scale, varying from Unattractive (1) to Attractive (5), for the aesthetic aspect. At this point they were asked to evaluate how they perceived the organization of screen elements and how motivating the screen layouts seemed to be. This process was repeated for the three sets of screens (home, content and e-mail).

The second part was intended to evaluate the perceived usability of the same sets of screens, using the same five-point Likert scale, ranging from Difficult to use (1) to Easy to use (5). At this point they could not see the “trailer preview” and were instructed to evaluate each screen (twelve in total) taking into account their perception of how easy to use the environment was, and the user satisfaction when navigating this particular VLE.

They could spend as much time as they wanted to evaluate each screen and only when they clicked on the button (giving grades) would the screen change to the next one, registering the data in a separate file. In addition, they could not return to the previous screen as

this would cause problems in recording and interpreting the data acquired.

Preliminary results

- The statistical analyzes of the first set of screens (home) found a high correlation (0.76) between the aesthetics aspects (mean = 4.25) and apparent usability (mean = 3.05); the standard deviation was found to be the same on both aspects (0.41) (Figure 4);

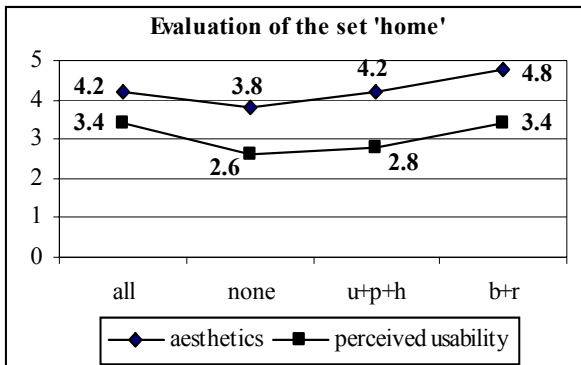


Figure 3. Aesthetics versus perceived usability.

- Design principles seems to be important in influencing aesthetic judgments;
- Unity and rhythm seems to be the most important principles, followed by proportion and homogeneity; balance seems to be the less important (Figure 5).

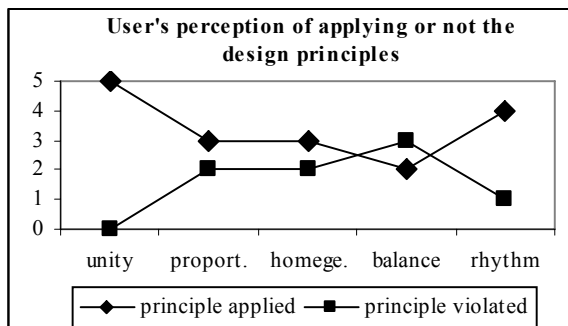


Figure 4. Perception of the design principles.

CONCLUSIONS

This pilot study has allowed the identification of necessary corrections to improve our main experiment. This involved 279 participants, divided into seven different categories, and will be reported elsewhere.

The results have shown that there is a positive correlation between the aesthetics aspects and perceived usability of VLEs interfaces, but the findings need to be confirmed with the data from the main experiment.

ACKNOWLEDGMENTS

I, R. Parizotto-Ribeiro, express thanks to CNPq for sponsoring this PhD. CEFET-PR that welcome the research and the staff and research students of LaMid that gave all the support needed to make the software application and run the experiment. I'll be forever grateful to Elizandro Ribeiro that supported me throughout the project (24/7).

REFERENCES

1. Chang, D., Dooley, L. and Tuovinen, J. Gestalt Theory in Visual Screen Design: A New Look at an Old Subject, *Australian Computer Society, Inc.* (2002). [online] Retrieved 02 February 2004, from <http://crpit.com/confpapers/CRPITV8Chang.pdf>
2. Kurosu, M. and Kashimura, K. Apparent Usability versus Inherent Usability Experimental analysis of the determinants of the apparent usability, *CHI'95 Mosaic of Creativity Proceedings* (1995), pp.292-293.
3. Ngo, D., Teo, L. and Byrne, J. Modeling Interface Aesthetics. *Information Sciences* (2003), vol.152, pp.25-46.
4. Norman, D. Emotional Design: Why we love or have everyday things. Basic Books, New York, USA, 2004.
5. Tractinsky, N. Aesthetic and Apparent Usability: Empirically Assessing Cultural and Methodological Issues, *CHI'97 Proceedings* (1997), pp.22-27.
6. Tractinsky, N., Shoval-Katz, A. and Ikar, D. What is Beautiful is Usable, *Interacting with Computers* (2000), vol.13 pp.127-145.
7. Hassenzahl, M. The interplay of beauty, goodness and usability in interactive products. *HCI Journal* ; in press.

The Online Community Framework's Performance as an Epistemic Tool

Clarissa M^a de A.
Barbosa¹
cbarbosa@inf.puc-rio.br

Carla Faria Leitão^{1,2}
cfaria@inf.puc-rio.br

Clarisse Sieckenius de
Souza¹
clarisse@inf.puc-rio.br

Jenny Preece³
preece@umbc.edu

1. SERG – Departamento de Informática da PUC-Rio

Rua Marques de São Vicente, 225, Gávea, Rio de Janeiro – RJ, CEP: 22453-900
(21) 31141500, ramal 3323

2. Departamento de Psicologia da PUC-Rio

Rua Marques de São Vicente, 225, Gávea, Rio de Janeiro – RJ, CEP: 22453-900
(21) 31141183

3. Information Systems Department, UMBC

1000 Hilltop Circle – Baltimore, MD 21250 - USA

RESUMO

Neste artigo apresentamos nossas conclusões atuais sobre o uso do Online Community Framework como uma ferramenta epistêmica para ajudar o avaliador a analisar e compreender a influência da tecnologia sobre atividade social online.

ABSTRACT

In this paper we present our current conclusions about the performance of the Online Community Framework as an epistemic tool to help evaluators analyze and understand how social activity online is enabled and affected by technology.

Keywords

Online Communities, User Experience Evaluation, Online Community Framework

INTRODUCTION

Communities are complex structures, characterized by the intense and varied social interaction among their members, as well as by their dynamic nature – they develop and continuously evolve. Therefore, developing successful online communities is not a trivial task. An important requirement for achieving it is to understand the impacts of software design on the evolution of online communities. This requirement motivated the development of the Online Community Framework (OCF) [de Souza & Preece, 2004], a theoretically-based analytic tool that can be used to leverage knowledge about how social activity is enabled and affected by technology.

We have been observing some Brazilian online health support communities since October, 2003. The observation is based on our knowledge of and experience with semiotic engineering [de Souza, 1993; 2001; forthcoming], discourse analysis theory and research [Wetherell, Taylor e Yates, 2001a e 2001b; da Silva et al., 2003], and online communities [Preece, 2000].

VI Simpósio sobre Fatores Humanos em Sistemas Computacionais — Mediando e Transformando o Cotidiano. Hotel Bourbon, Curitiba, 17 a 20 de outubro de 2004. UFPR, CEIHC—SBC.

The Multiple Sclerosis Sufferers Society (MSSS) [SPEM, 2004] is one of these communities. It was created in 2001, by a Multiple Sclerosis¹ (MS) sufferer. Its purpose is to enable people affected by MS to clarify their doubts, to get information about the disease, and also to share their experience with each other.

In January 2004 there was a great change in MSSS. Its website was totally restructured and a new socializing environment was created to which everyone should migrate. A new *Interaction* area was provided to facilitate contact with people affected by MS and with the website's designer. Forum, group, chat, and e-mail tools can be used since then for a variety of purposes.

The forum tool is a third-party product created on December 2001, by another MS sufferer – not the one who created MSSS. Since March 2003, however, the forum has been managed by the creator of MSSS. By the end of March 2004, 10823 messages had been exchanged, although there was no information available about the number of members.

One of the two group tools that can be used is an MSN Group. We have no information about when it was created, but the manager of this group, the MSSS creator, joined the community on June 2002. By late March 2004, there were 308 registered members, 374 messages had been posted, and 94 photos had been published.

The other group tool is a Yahoo Group. It was also created by the MSSS creator, in January 2004, as a response to the technical problems faced by members of the forum tool. By the end of March 2004, 94 visitors had become members, 1141 messages had been posted, and 14 photos had been published.

MSSS deserves special attention because it now faces a delicate transition period, which threatens its evolution. Up until January 2004, messages exchanged among their members generally characterized it as a successful online community. However, since then, the content of the messages has changed. The intense emotional support they used to offer one another has been gradually

¹ Multiple Sclerosis is a chronic neurological disease whose cause is unknown and for which there is no cure.

replaced by factual information exchanges about the disease and its treatment. In some messages members have explicitly manifested their dissatisfaction and sadness with this change.

This a prime opportunity to evaluate OCF’s performance as an analytic tool. Our goal is to find out if technology has in any way caused or facilitated this problematic change (as perceived by members) in the community. This is OCF’s main purpose as an analytic tool. We also want to explore and understand more deeply the type of knowledge that an OCF-based analysis can offer to an evaluator, as well as the costs and benefits of using it for extensive studies like this one.

Next we briefly present OCF and report the analysis we made. Then we discuss our current conclusions about OCF’s performance as an analytic tool, and present the main items in our future research agenda.

THE ONLINE COMMUNITY FRAMEWORK

OCF is a theoretically-based analytic tool for helping designers understand existing technology-supported social activity online. By doing so, it assists them in the development of computer applications that support social interaction online while making them reflect on how their design may impact the birth and the evolution of the community. It has a remarkably epistemic nature, in the sense that it seeks to empower designers with knowledge and knowledge-increasing strategies.

The main characteristic of OCF is its theoretical nature. OCF follows the principles of semiotic engineering, a theory in which interactive software is a one-shot message sent from designers to users about how the latter should interact with the application in order to achieve a certain range of goals and experiences intended by the former. In this perspective, the perceived quality of interactive software is not only a function of how well designers understand and interpret users’ requirements, but also of how well they communicate this back to users through their designs. This theoretical foundation explains OCF’s emphasis on communication.

Another important aspect for understanding OCF is its underlying definition of an online community (OC): ‘a group of people, who come together for a purpose online, and who are governed by norms and policies’ [Preece, 2000]. This definition permeates all of OCF’s constituents, namely: the online community constituent; the usability and sociability constituent; and the interpretive constituent. The three constituents and the relations among them are shown in Figure 1.

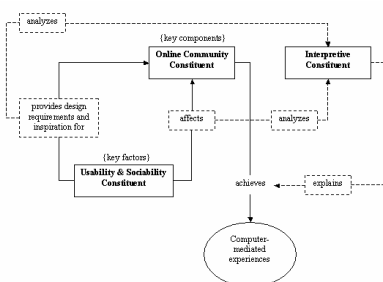


Figure 1. The overall constituent structure of OCF

The online community constituent (OC Constituent) represents an idealized abstraction of an OC. It contains a set of entities, relations among them, and attributes of both entities and relations. The four key entities in it are Community, People, Purposes and Policies, which recursively unfold into other entities and relations. The communications entity is particularly important in view of the fundamental role played by communication (both between designers and users, and among users themselves) in the success of an OC.

The usability and sociability constituent refers to the OC goals and needs that must be considered in the development of computer applications meant to support the social activity of communities online.

Finally, the interpretive constituent (Figure 2) enables evaluators to assess how technology is affecting human interaction, both between designers and users, and among community members. The seed for this evaluation are the messages in the OC Constituent’s communications entity.

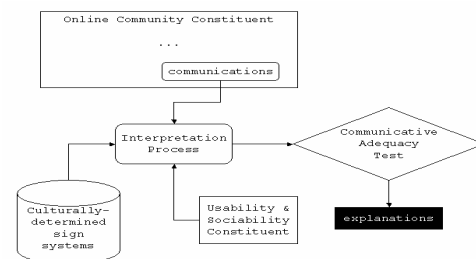


Figure 2. The interpretive constituent of OCF

An important element of the interpretive constituent is the Communicative Adequacy Test (CAT). It helps evaluators assess communication, and come up with explanations for computer-mediated communication experiences. The evaluator inspects communication online by answering the following questions: (i) Do I understand this message?; (ii) Can I detect whom the message is from?; (a) Who means it?; (b) Who has written it?; (iii) Is the system interfering with the communication?

CAT results derive from the answers given to the questions above, and may vary depending on: which values are used to answer the questions; the usability and sociability guidelines adopted by the evaluator; and the set of sign systems taken as reference for the test.

The Analysis Process

We have analyzed MSSS based on the messages exchanged both between its creator and users, and among its members. An important point about the community’s creator is that he *designed* (to a considerable extent) the online environment where the community evolves. He thus shares with the designers of MSN Groups and Yahoo Groups the responsibility for how the *software* appears to users. We were interested in “listening” the designers’ one-shot message, with a special emphasis on knowing the MSSS creator/designer’s understanding of who is MSSS and which are, for him, the adequate means of communication for this community. So we have concentrated on messages that deal with these two issues – the messages that introduce MSSS and the set of

socialization tools made available for users. For each message, we have: (I) instantiated the `communications` entity of the OC Constituent, (II) performed CAT; and (III) consolidated the knowledge obtained in steps (I) and (II).

In general, the first step consists in attributing values to the `communications` entity's attributes depending on the type of message. CAT consists in answering the set of questions listed above. And the last step consists in instantiating other entities of the OC Constituent that are implied in the message. A detailed description of the analysis can be found in [Barbosa et al., 2004].

Results of the Analysis

The OCF-based analysis revealed that there are problems in the communication between the MSSS designer and members regarding all three key components of an OC: People, Policies and Purposes.

With respect to People, there is a lack of coherence among the various ways to communicate who are the intended members of all three groups in MSSS. This gives rise to doubts about this community's identity, as explicitly manifested by some of its members in exchanged messages.

With respect to Policies, there is a lack of information about this community's policies in the message that presents MSSS. This problem may be compensated by more clearly stated policies that are presented separately in each of the various environments where members actually interact.

With regard to Purposes, the purpose of all three groups is unclear. This has negative effects on communication among MSSS members, since it leaves room for conflicting purposes in using the Group and Forum tools. Members can follow their individual inclination, and eventually contribute to a weakening of the community's cohesion. This is actually happening in MSSS. Some members have arbitrarily chosen to use only one of the interaction tools, while others preferred to use more than one, for purposes and contexts that are unclear. Thus, in practice the members of the Yahoo Group, the MSN Group and the Forum are not the same, which disperses and weakens the community.

The OCF-based analysis also revealed that communicative breakdowns concerning People, Purposes and Policies constitute a threat for the evolution of MSSS, since they jeopardize the success of interaction among members, an essential activity for the survival of online communities. Messages exchanged among MSSS members clearly demonstrate that they not only perceive, but also regret the difficulties they are facing.

CURRENT CONCLUSIONS AND FUTURE RESEARCH

The main contributions of this research are twofold. First, we acquired deeper knowledge about the community itself and the use of OCF as an analytic tool. Second, it generated interesting and relevant items for our research agenda. The kind of knowledge gained about MSSS has been presented in the previous section. So we will now describe our experience while carrying out the

OCF-based analysis and present the next research steps we intend to take.

Semiotic engineering is aligned with Schön's view of what design is, and consequently what kind of knowledge designers need to have in order to be successful [Schön, 1983]. Schön advocates that designers are constantly faced with unique problematic situations, and not with clearly defined problems. Thus, one of the most fundamental tasks in design is to frame the situation as a new *problem*. Since the problem is most probably unique, its solution is likewise unique.

Schön then advocates that designers must have a particular kind of knowledge that will help them understand the problematic situation, frame the problem(s) perceived in it, decide how to solve it (them), and compare alternative solutions. In his view, designers should be equipped with essentially epistemic tools, namely tools that increase their understanding of the problem itself and of candidate solutions.

The nature of the design tools derived from semiotic engineering is thus essentially epistemic. These tools trigger the designer's reflection on numerous and unique problematic situations. By so doing, such tools enable designers to increase constantly their knowledge, to improve considerably the decision-making process inherent to design, and to improve consistently the quality of their products.

The analysis of MSSS described in the previous section illustrates the kind of knowledge generated by an epistemic tool such as OCF. The kind of analysis it supports calls the evaluator's attention to certain aspects of an OC that are systematically related to each other and to the success of the community (*e.g.*, the ones that make up the OC Constituent). It also reveals the consequences of some design decisions, deliberate or not, for the success and evolution of an OC (*e.g.*, the way the designer communicates his/her design view to the users). While leading evaluators to acquire a deeper understanding of the community, the OCF-based analysis invites them to consider or produce possible solutions for the detected problems, as well as to judge the complexity of their implementation.

The OCF-based analysis of MSSS increased our knowledge on the community as a whole, as well as on the probable effects of certain design decisions on the community's success and evolution. Thus, our experience in applying OCF to evaluate an OC reinforces the contribution of this framework to the outcome of a knowledge-intensive design activity.

When compared to predictive inspection methods (*e.g.*, heuristic evaluation, compliance with design guidelines, and so on), we realize that OCF-based analyses supplement them by providing a clearer and richer description of how the constitutive elements of an OC affect one another and the community itself. Predictive inspection methods of analysis typically result from statistically significant empirical studies that aim at associating certain features of interactive software with particular kinds of users' reactions. These methods usually do not have any commitment to describing how these features relate to each other, and their explanatory

power is causal in a very strict sense (i.e., they typically provide explanations base merely on co-occurrence).

Direct and indirect messages sent from MSSS designer to member users demonstrate that he knew exactly what he wanted Forum members to do – namely that they should all migrate to the newly created Yahoo Group. However, when he restructured MSSS website, he did not manage to communicate his intentions to the users successfully. The gap between what the designer intended to communicate and what the receivers of his communication actually got from his messages underlines the need for tools that will help designers get their message across to users. OCF is one such tool in the sense that it helps the designer detect the problem, find an explanation for it, and hopefully find a better solution than the original one.

Upon his spontaneous demand, we will give the MSSS designer feedback on his design based on the analysis we reported here. We will also give him suggestions about how to redesign the community's website in order to improve the communication of his design vision to the users. Our intention is to compare the website's usability and sociability before and after his redesign, highly valuable evidence of what epistemic tools can actually do in this kind of design situation.

We will also analyze MSSS using other methods derived from theories that supplement semiotic engineering. The purpose of such analysis is to triangulate our findings about MSSS. Likewise, we intend to examine possible relations between CAT results and usability and sociability problems detected by other methods of analysis.

Acknowledgments

The authors would like to thank CNPq and CAPES for supporting this work financially, and to thank Marcelo Morita for his invaluable contribution for this research.

REFERENCES

1. Barbosa, C.M.A., Leitão, C.F., de Souza, C.S., and Preece, J. *An Online Community Framework-based Analysis of an Existing Online Community*. Technical Report. Departamento de Informática, PUC-Rio. 2004.
2. da Silva, E.J., de Souza, C.S., Prates, R.O., Nicolacida-Costa, A.M. What They Want and What They Get: a study of light-weight technologies for online communities. *Proc. CLIHC 2003*, 135-146.
3. de Souza, C.S. and Preece, J. A Framework for Analyzing and Understanding Online Communities. *Interacting with Computers*, 16, 3 (2004), 579-610.
4. de Souza, C.S. *The Semiotic Engineering of Human-Computer Interaction*. Cambridge, MA: The MIT Press. Forthcoming.
5. de Souza, C.S. Editorial of the Special Issue on Semiotic Approaches to User Interface Design. *Knowledge Based Systems*, 14, 8 (2001), 415-418.
6. de Souza, C.S. The Semiotic Engineering of User Interface Languages. *International Journal of Man-Machine Studies*, 39 (1993), 753-773.
7. Preece, J. *Online Communities: Designing Usability, Supporting Sociability*. Chichester, UK: John Wiley & Sons, 2000.
8. Schön, D.A. *The reflective practitioner: how professionals think in action*. New York, NY: Basic Books, 1983.
9. SPEM, 2004. *Sociedade dos Portadores de Esclerose Múltipla*. <http://www.spem.kit.net/>. Acessado em 26/03/2004.
10. Wetherell, M.; Taylor, S.; Yates, S.J. (ed) *Discourse as Data: a guide for analysis*. London, Sage Publications, 2001a.
11. Wetherell, M.; Taylor, S.; Yates, S.J. (ed) *Discourse Theory and Practice: a reader*. London, Sage Publications, 2001b.

Um Projeto de Interfaces para um Jogo Multimídia Direcionado a Portadores de Síndrome de *Down*

Carla Gonçalves Brum

Centro Universitário Franciscano
Rua dos Andradas, 1614
97010-032 – Centro
Santa Maria/RS
carlabrum@smail.ufsm.br

Ana Paula Canal

Centro Universitário Franciscano
Rua dos Andradas, 1614
97010-032 – Centro
Santa Maria/RS
apc@unifra.br

Marcos Luís Cassal

Centro Universitário Franciscano
Rua dos Andradas, 1614
97010-032 – Centro
Santa Maria/RS
cassal@unifra.br

RESUMO

As pessoas portadores de Síndrome de *Down* têm a deficiência mental como uma das características relevantes, apresentando limitações no seu desenvolvimento motor, cognitivo e social. A educação dessas pessoas, também conhecida como educação especial, é uma atividade complexa e ao mesmo tempo importante, pois proporciona a inclusão dos portadores na sociedade. Neste contexto, a informática desempenha um papel significativo como ferramenta que auxilia o aluno em seu processo de desenvolvimento e aprendizado. Em um sistema é importante pensar nos aspectos de interface e a área de Interação Homem-Computador (IHC) estuda como as pessoas interagem com o sistema computacional e como os computadores afetam as pessoas e a sociedade. Este artigo aborda a elaboração de um projeto de interfaces para um jogo multimídia voltado à portadores de Síndrome de *Down* que estão em processo de alfabetização.

ABSTRACT

The main characteristic of individuals with Down Syndrome is mental deficiency. They have limitations in your motor, cognitive and social growing. The education of those people, also known as special education, is a complex and important activity, because this education provides the individuals inclusion in the society. In this context, computer science is like a tool to help students in your process of growing and learning. When a computer system is being done, is important thing about the user interfaces design. Researches in Human Computer Interface are concerned at people interaction with computing systems and how computers affect people and society. This paper is about the development of user interfaces for a multimedia game appropriated for individuals with Down Syndrome who are being alphabetized.

Palavras chave

Interação Homem-Computador, Síndrome de *Down*, Educação Especial

INTRODUÇÃO

A tecnologia computacional está inserida na sociedade e, sobretudo, na educação, promovendo mudanças nos métodos de ensino tradicionais. Ao ser construído um *software*, que de alguma forma auxilie no processo de aprendizagem, é importante considerar os vários perfis de usuários da aplicação, o seu grau de dificuldade em aprender e identificar quais habilidades podem ser desenvolvidas com o uso do *software*.

Aproximadamente 14,6% da população brasileira é portadora de alguma necessidade educacional especial [3]. As pessoas com Síndrome de *Down* possuem atraso mental, e grande parte dos portadores apresentam problemas de audição e visão [8]. Os portadores desta síndrome possuem necessidades educacionais especiais, pois apresentam limitações na capacidade de aprendizagem e em seu desenvolvimento físico, mental e social [6].

Este trabalho tem como objetivo propor interfaces interativas, para um jogo multimídia voltado a portadores de Síndrome de *Down* que estão em processo de alfabetização. Para o processo de desenvolvimento das interfaces realizaram-se as etapas de análise de usuários e a análise de tarefas. Após desenvolveu-se o projeto, construindo o modelo conceitual da aplicação e os cenários de uso e definindo os componentes de interação bem como os estilos de interação das interfaces. A observação dos alunos especiais da Associação de Pais e Amigos dos Excepcionais (APAE) de Santa Maria/RS, na realização de suas tarefas com o computador e a interação com a professora responsável subsidiaram o desenvolvimento deste estudo.

SÍNDROME DE *DOWN*, EDUCAÇÃO E INFORMÁTICA

A Educação Especial desempenha um papel importante no processo de desenvolvimento dos portadores de necessidades especiais, proporcionando que estes sejam incluídos na sociedade e que exerçam sua cidadania. Os recursos de computação estão cada vez mais presentes na educação, incluindo a educação especial, com o objetivo de aprimorar e estimular o ensino de crianças e adultos.

O uso da computação na educação especial aumentou o grau de interesse dos alunos em aprender novos conteúdos e o interesse em trabalhar em equipe [9]. Algumas pesquisas e projetos têm sido feitos com relação

ao uso dos recursos de *software* na educação especial. Dentre esses projetos, cita-se a criação do Núcleo de Informática na Educação Especial – NIEE, pela UFRGS [7]; o desenvolvimento de um jogo multimídia adaptativo pelo Departamento de Computação da Universidade Estadual de Londrina – PR [1]; a adaptação do *software HagáQuê* pelo Instituto de Computação da UNICAMP [9].

PROJETO DAS INTERFACES

A metodologia para construção do projeto de interfaces descrita neste artigo refere-se a MANDEL [5] e LEITE [4]. Esta seção aborda a aplicação do processo de desenvolvimento de interfaces para um jogo multimídia direcionado a portadores de Síndrome de *Down*. Para a visualização das interfaces desenvolvidas, um protótipo, inicialmente, está sendo desenvolvido.

Através da observação dos portadores da Síndrome de *Down* durante a execução de suas atividades no computador, verificou-se que a percepção e o processo de aprendizado são diferentes entre eles. Enquanto alguns alunos ainda aprendiam a desenhar utilizando o computador, outros já digitavam os textos que a professora escrevia no quadro. A partir destes aspectos, propõe-se uma interface interativa e adaptável aos fatores cognitivos dos alunos especiais.

Na análise, identificou-se que todos os usuários da aplicação estão em processo de alfabetização, utilizam o computador quatro horas por semana, sendo considerados iniciantes no domínio da informática. Além disso, notou-se que os alunos especiais possuem graus de conhecimento diferenciados, quanto à alfabetização. A motivação é outro fator de importância, pois todos os usuários são motivados a explorar determinada informação, quando esta vem seguida do uso de imagens, sons, cores e animações.

Para construir interfaces para *softwares* destinados a portadores da Síndrome de *Down* é necessário seguir algumas recomendações como a livre interação, o uso de ícones grandes, textos curtos, letras de forma e maiúsculas semelhantes ao teclado, animações, filmes, som, imagens e cores [2]. A interface também deve conduzir o usuário e fornecer respostas a todas as suas interações.

As interfaces propostas enfatizam tarefas que estimulem as percepções visual, auditiva e linguagem oral do aluno especial, auxiliando no processo de aprendizagem e no desenvolvimento cognitivo. Uma das formas de desenvolver a linguagem oral e percepção auditiva e visual é utilizar um comando de voz para dizer que tipo de objeto o aluno deve escolher dentre os vários mostrados a ele. Para estimular a leitura pode-se fazer com que o aluno relacione figuras a suas letras iniciais. Para incentivar também a escrita pode ser usada uma imagem de uma história relacionada a sua realidade, onde o aluno visualiza uma figura ouve uma história sobre aquela figura e copia as frases sobre a imagem em um editor de textos.

Os estilos de interação utilizados nas interfaces são o GUI - *Graphical User Interface*, WIMP - *Windows, Icons, Menus e Pointers* e Manipulação Direta, por serem os mais apropriados ao perfil dos usuários e às considerações mencionadas anteriormente. Procurou-se organizar os componentes de interação da interface de forma harmoniosa para facilitar a visualização e interação com os vários elementos. Houve o cuidado em apresentar na tela somente os componentes úteis na execução da tarefa determinada. Outros aspectos observados foram as cores, onde optou-se por combinações de cores com contraste, o uso de animações, sempre que possível, como forma de estimular o usuário e o uso de áudio, como resposta do sistema à interação do usuário.

O *software* proposto é um jogo que possibilita ao educando passar de um nível mais fácil para um nível mais difícil conforme o seu desempenho. Outra alternativa para o uso do jogo é o professor escolher o nível em que o jogo irá começar, através do menu de configurações, de acordo com as habilidades do aluno.



Figura 1. Interface - Categoria.

O jogo possui três níveis: fácil, médio e difícil. O protótipo até o momento possui os níveis fácil e médio. O nível fácil é dividido em categorias, como, por exemplo, frutas, animais, e transportes (ver Figura 1). O aluno começa no nível fácil, escolhe uma categoria e a atividade a ser executada.

As interfaces apresentadas a seguir referem-se à atividade “A Festa dos Animais”, pertencente à categoria “Animais” (ver Figura 2).



Figura 2. Interface Atividade – A Festa dos Animais

O nível fácil traz atividades que estimulam a percepção visual e auditiva através do reconhecimento de animais, frutas e transportes. Na atividade “A Festa dos Animais” um comando de voz pergunta ao aluno quem está na festa, o aluno clica no botão da formiga (ver Figura 3), outra interface é apresentada ao aluno. Nesta interface

(ver Figura 4), um comando de voz informa o nome do animal que o aluno deverá escolher dentre as cinco opções mostradas. Se o aluno demorar aproximadamente 15 segundos para escolher o animal, o sistema informa novamente o animal que ele deve escolher, por meio de um comando de voz.



Figura 3. Interface Atividade – A Festa dos Animais



Figura 4. Interface Atividade – A Festa dos Animais

Ao clicar no animal que o sistema informou, um comando de voz avisa que o aluno acertou e ganhou uma estrela como prêmio e o animal escolhido aparecerá (ver Figura 5). Se o aluno clicar no animal errado, uma outra interface é apresentada, e através de um comando de voz, o sistema orienta o aluno a escolher novamente o animal correto.



Figura 5. Interface Atividade – A Festa dos Animais.

O aluno joga várias vezes, de forma que todos os animais sejam escolhidos. No final desta atividade é apresentada ao aluno, uma interface que o parabeniza, através de um comando de voz, são mostradas figuras festivas e as estrelas que ele ganhou como prêmio (ver Figura 6).



Figura 6. Interface Atividade – A Festa dos Animais

Para continuar, o aluno pode clicar no botão do boneco, retornando, assim, à interface das categorias. Se isto não acontecer, passados aproximadamente 15 segundos, a interface das categorias é novamente mostrada ao aluno (ver Figura 1).

O nível médio do jogo traz atividades para o reconhecimento de letras e os números, além de estimular a percepção visual e auditiva. A metáfora é de um jogo com cartas. Na atividade com letras, o aluno é estimulado a associar a letra inicial com a correspondente figura do objeto mostrado, clicando respectivamente na letra do alfabeto e em seguida no desenho, ou vice-versa. Se ele acertar, aparece na tela uma estrela e é emitido um comando de voz, informando que acertou. Se errar, um comando de voz informa a letra e o desenho correspondente (ver Figura 7).

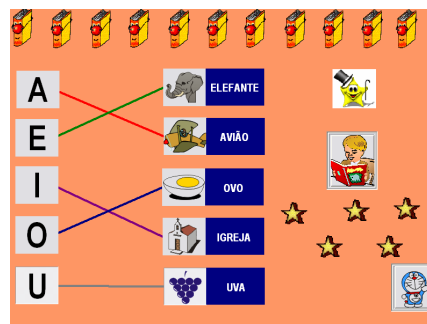


Figura 7. Interface Atividade - Ligue as Letras

Ao organizar os componentes de interação dentro das interfaces foram levados em consideração a localização, o tamanho, as cores e as figuras dos objetos de interação, com o objetivo de destacar para o aluno com quais botões ele deve interagir. Observa-se que o botão da formiga (ver Figura 3) está localizado no centro da tela, da mesma forma que os botões para escolher os animais (ver Figura 4). Nota-se também que o botão da formiga, é um pouco maior que os outros presentes na interface. Isto possibilita que este componente fique destacado em relação aos outros da tela.

Animações para atrair a atenção do aluno e estimulá-lo a desempenhar a atividade proposta foram utilizadas. Alguns exemplos são os bichinhos da interface (ver Figura 2), o cão amarelo (ver Figura 4), bolo (ver Figura 5) e as estrelas (ver Figura 6).

Para cada interação do aluno, há um tipo de resposta do sistema, através de um comando de voz, animação, som

ou música. Esta resposta do sistema serve para orientar o aluno na execução de uma atividade, para parabenizá-lo e ainda como fator motivador para o uso do jogo.

Os textos presentes nas interfaces são escritos em letras de forma, maiúsculas e semelhantes às letras do teclado. Este tipo de letra foi escolhido porque aparece com maior frequência no dia-a-dia dos alunos, como por exemplo, nas placas de ônibus e sinalizações, o que proporciona a eles uma maior facilidade na memorização e na digitação de textos ou palavras.

O protótipo inicial foi implementado no ambiente visual *Borland Delphi*. Para as animações utilizou-se o componente GIFIMAGE. O controle do tempo que o aluno leva para executar alguma ação e o momento da resposta do sistema foi feito com a utilização do componente *Timer*. A reprodução dos comandos de voz e das músicas foram possíveis pelo componente *MediaPlayer*. As figuras foram selecionadas do CD *Gallery Cliparts* do *CorelDraw* e de sites da Internet.

O protótipo construído foi apresentado à professora da APAE de Santa Maria, com o objetivo de obter uma avaliação inicial da interface. A professora identificou aspectos relevantes como: a resposta do sistema a cada interação do aluno; a escolha e combinação das cores; figuras e animações usadas, destacando quais componentes o aluno deve interagir; quanto às atividades, contribuem para o desenvolvimento do educando no processo de alfabetização, estimulando também a percepção visual e auditiva. Após esta validação, alguns pontos foram melhorados neste protótipo, como o uso de tela inteira e a interação do usuário, de forma que a interface passe a conduzir o aluno, sem a necessidade de obtenção de ajuda por um botão específico.

CONSIDERAÇÕES FINAIS

Com este trabalho foi possível observar o quanto é importante construir interfaces que atendam as expectativas dos usuários. Estes devem sentir-se estimulados a aprender e a explorar a aplicação.

Aqui foram apresentadas algumas diretrizes para o desenvolvimento de interfaces. O objetivo foi propor interfaces que estimulem e facilitem a interação homem-computador de pessoas com Síndrome de *Down* e que estão sendo alfabetizadas.

O jogo proposto visa estimular a comunicação, linguagem, escrita, percepção visual e auditiva, contribuindo para desenvolvimento cognitivo do aluno especial.

Os trabalhos futuros consistem em definir outras atividades para as categorias do nível fácil, implementando suas interfaces. E para o nível médio elaborar outras atividades, desenvolvendo o projeto das interfaces. O nível difícil pretende abordar atividades para trabalhar a escrita e leitura. Um exemplo de atividade que está sendo desenvolvida é possibilitar que o aluno ouça uma história, visualize uma imagem relacionada à história e copie frases conforme aparecem em um editor de textos.

Pretende-se também validar as interfaces junto aos alunos especiais da APAE de Santa Maria - RS.

AGRADECIMENTOS

Agradecimentos à direção da APAE de Santa Maria-RS e à professora Maritê pelo apoio no desenvolvimento deste estudo.

REFERÊNCIAS

1. Amaral, M., Teixeira, E., Ribas, H. Desenvolvimento de Jogo Multimídia Adaptativo para Crianças Portadoras de Síndrome de *Down*, *Anais do Simpósio Brasileiro de Sistemas Multimídia e Web*, 2003, Salvador-Ba.
2. Campos, M. B., Silveira, M. S. Tecnologias para Educação Especial. <http://lsm.dei.uc.pt/ribie/docfiles/txt2003423195334167.PDF>.
3. IBGE. Censo de 2000. <http://www.ibge.gov.br>.
4. Leite, J. C., Souza, C. S., Prates, R. O., Barbosa, S.D.J. Projeto de Interfaces de Usuário. <http://www.dimap.ufrn.br/~jair/piu>
5. Mandel, T. *Elements Of User Interface Design*. New York: John Willey & Sons, 1997.
6. Melo, S., Silva, N. C. A Interação dos portadores de Síndrome de *Down* com seus colegas na sala de aula em uma abordagem inclusiva. http://www.nead.unama.br/monografias/pdf/INTERA_CAO_PORTADORES_SINDROME_DOWN_SEUS_COLEGAS.PDF.
7. NIEE. Núcleo de Informática na Educação Especial. <http://www.niee.ufrgs.br/historico/hist1.html>.
8. Saúde Interativa. “Pais e filhos: que herança genética é deixada?”. Ano 2. Nº 9, Set. 2003.
9. Tanaka, E. Tornando um *Software* acessível às pessoas com necessidades educacionais especiais. <http://www.nied.unicamp.br/~hagaque/publicacoes/rel-at-cient01.pdf>.

Uma experiência de construção da ajuda online segundo o paradigma da engenharia semiótica

Thaís Chiappetta Salgado

Clarisse Sieckenius de Souza

SERG, Grupo de Pesquisa em Engenharia Semiótica
Departamento de Informática, PUC-Rio
Rua Marquês de São Vicente 225
Rio de Janeiro, RJ - Brasil
{thais, clarisse}@grad.inf.puc-rio.br

RESUMO

A construção de sistemas de ajuda é freqüentemente negligenciada em projetos de *software* e, usualmente, não é feita por profissionais com grande conhecimento em IHC. Para que seja de fato utilizado, portanto, um novo paradigma para o desenvolvimento de sistemas de ajuda *online* deve ser fácil de ser entendido e aplicado mesmo por pessoas que não o conheçam anteriormente. Este artigo visa, então, apresentar os resultados de uma avaliação conduzida com o objetivo de investigar a dificuldade de compreensão e de uso do paradigma proposto pela engenharia semiótica a partir do ponto de vista de desenvolvedores de *software* que não possuem conhecimentos extensos na área de IHC.

Palavras Chave

Sistemas de ajuda *online*, engenharia semiótica, avaliação com desenvolvedores.

INTRODUÇÃO

O paradigma para o desenvolvimento de sistemas de ajuda *online* proposto em [3] baseia-se na teoria da engenharia semiótica e utiliza as expressões empregadas no método de avaliação de comunicabilidade [1] como base para estabelecer um conjunto de perguntas através das quais o usuário pode acessar a ajuda *online*. Além disso, ele aproxima o sistema de ajuda do momento em que ocorreu a ruptura na comunicação *designer*-usuário, disponibilizando essas perguntas diretamente na interface da aplicação e, assim, contextualizando-o.

Associado a este paradigma, é proposto ainda um método para o desenvolvimento desses sistemas [3]. Este método engloba a utilização de diversos modelos de IHC — modelo de usuários, de tarefas, de interação, entre outros — e também sugere que o projeto da ajuda seja feito ao longo do desenvolvimento da aplicação em si.

Nem sempre, entretanto, será possível projetar o sistema de ajuda desta forma. É provável, além disso, que os seus desenvolvedores não tenham conhecimentos extensos na área de IHC, o que torna difícil a adoção integral do paradigma em questão.

Assim, o objetivo deste trabalho é avaliar este paradigma a partir do ponto de vista de desenvolvedores que não conheçam profundamente nem a teoria da engenharia semiótica nem o paradigma em si. Pretende-se, desta forma, investigar a dificuldade de compreensão e de aplicação do mesmo, bem como captar a opinião manifestada pelos desenvolvedores ao travar o primeiro contato informal com ele.

MÉTODO DE AVALIAÇÃO

Foi realizada uma avaliação informal, de caráter exploratório, cujo objetivo primordial era colher evidências de dificuldades inesperadas — pelos avaliadores — enfrentadas pelos participantes, ao tentarem utilizar pela primeira vez o paradigma de construção de ajuda *online* proposto pela engenharia semiótica.

Para alcançar os objetivos mencionados, foi desenvolvido e aplicado um teste no qual se pedia a cada participante que desenvolvesse o esboço do sistema de ajuda para a tarefa de incluir um usuário na lista de contatos do ICQ. É interessante notar que a avaliação foi baseada na interpretação da metacomunicação *designer*-usuário, já que, embora todos os participantes tenham sido selecionados por ter boa familiaridade com este sistema, obviamente nenhum deles havia desenvolvido a aplicação para qual esboçou a ajuda.

Perfil dos participantes

O teste contou com a participação de quatro desenvolvedores iniciantes, alunos do curso de graduação de Bacharelado em Informática na PUC-Rio. Estes alunos estavam distribuídos entre o terceiro, sétimo e oitavo períodos do curso e não haviam concluído nenhuma disciplina específica da área de IHC. Nenhum deles possuía conhecimento prévio do paradigma considerado.

Tarefa a ser executada

A escolha do ICQ e da tarefa de incluir um usuário na lista de contatos justifica-se porque o ICQ é amplamente difundido entre os alunos da graduação em informática na universidade citada. Além disso, a tarefa escolhida é bastante importante dentro da aplicação e precisa ser regularmente executada pelos usuários do ICQ.

Ela também apresenta características interessantes do ponto de vista do sistema de ajuda, notadamente poder ser dividida em subtarefas e possuir mais de um método de

execução. Ambas são informações importantes que deveriam ser descritas na ajuda.

Procedimento de aplicação do teste

A aplicação do teste decorreu em três etapas. Na primeira, o participante recebeu e leu um texto explicando informalmente o paradigma que deveria ser aplicado, bem como a tarefa a ser executada. Uma característica especial deste texto é que ele foi intencionalmente redigido em um estilo muito próximo do estilo retórico em que o produto final do discurso de ajuda é produzido, neste paradigma. Desta forma, ele próprio constituía um exemplo do que cada participante deveria fazer.

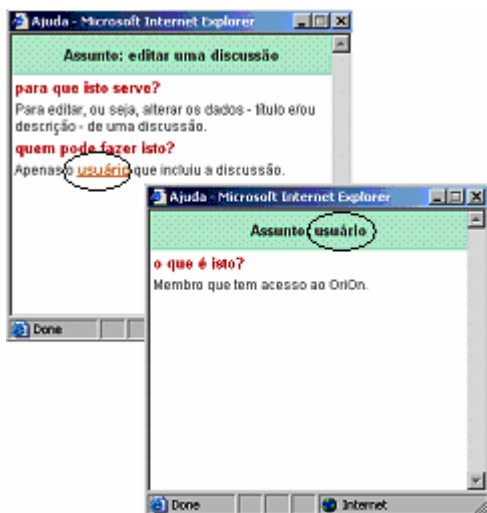


Figura 1: Exemplo da Ajuda Online no Paradigma proposto

Para ilustrar, mostramos a seguir um exemplo da ajuda produzida por este paradigma (Figura 1) e um trecho do texto que utilizamos como instrumento para o teste (Figura 2). O texto, com quatro páginas de comprimento, continha ilustrações cuidadosamente escolhidas de como a ajuda se apresenta e como funciona (veja-se o “Quer ver um exemplo?” na Figura 2). As imagens da Figura 1 foram de fato extraídas do instrumento do teste e integram o exemplo apresentado aos participantes.

Apesar de ser fornecido aos participantes um pequeno conjunto de perguntas mais comumente utilizadas, o texto ainda deixava explícito que, caso eles sentissem a necessidade de criar novas perguntas, isto poderia e deveria ser feito livremente. Desta forma, o esboço de *help* projetado pelos participantes não ficaria restrito somente às informações dadas.

A segunda etapa consistia no projeto da ajuda em si. Os participantes podiam explorar a aplicação livremente no computador e deveriam escrever no papel o esboço da ajuda proposta. Esta etapa possuía um tempo limite de quarenta e cinco minutos.

Por fim, a terceira etapa era uma entrevista pós-teste na qual visava-se descobrir as dificuldades encontradas pelos participantes tanto em relação ao teste quanto em relação ao entendimento e à aplicação do paradigma, bem como sua opinião sobre o mesmo.

O que é esse novo modelo para sistemas de ajuda?

O modelo proposto — e que desejamos avaliar através da aplicação deste teste — é uma maneira diferente de apresentar para o usuário o conteúdo da ajuda.

Tudo bem, mas como é que ele funciona?

Bom, para entender isso, a primeira coisa que você precisa fazer é imaginar a interação do usuário com o sistema como uma conversa. O usuário, por exemplo, ‘diz’ ao Word® que quer abrir um novo documento e, em seguida, *voilà*, o Word® abre um novo documento para ele!

Só que as respostas do sistema não são só do tipo “aqui está!” ou “pronto”, até porque o usuário não está só dizendo “me dá isto” ou “faz aquilo”. A gente tem também de pensar que o usuário às vezes está querendo conversar sobre “e se eu fizesse isto?” ou “dá para fazer aquilo? por que não?”. Muitas vezes ele quer saber “qual é melhor?” ou “para que serve isto?”. A resposta para este tipo de pergunta é que vai fazer diferença para o usuário, e levá-lo talvez muito mais rápido para um grau “avançado” de eficiência e conhecimento com o nosso sistema. E quem é que tem estas respostas? Nós, que fizemos o sistema! Logo, somos nós que temos de dá-las, isto é: se o usuário conseguir fazer a pergunta...

Mas, perai! Como é que eu vou saber tudo o que vai na cabeça do usuário?

Aí é que está: não é possível saber. Mas, voltando à nossa analogia, quando você está conversando com alguém e não entende algo que a outra pessoa diz, o que você faz? Numa conversa normal, você interrompe a pessoa e pergunta: “o que você está querendo dizer?”.

Então? Não seria ótimo se nós pudéssemos fazer o mesmo com os *softwares*?

É para isso que esse novo modelo serve?!

Sim. Enquanto desenvolvedores, nós somos capazes de identificar ao menos alguns momentos problemáticos nessa conversa entre o sistema e o usuário — como, por exemplo, quando usamos uma palavra difícil, ou um ícone com uma gravura que pode gerar dupla interpretação, ou ainda quando o sistema possui alguma função que o usuário não sabe exatamente para que serve ou por que deve ser realizada.

Nesses momentos é que a ajuda deve estar disponível de forma fácil e rápida, sem longas explicações desnecessárias e indo direto ao ponto.

Quer ver um exemplo?

Figura 2: Exemplo de um trecho do texto utilizado como instrumento do teste.

Dois aspectos principais nortearam a utilização deste tipo de avaliação ao invés de, por exemplo, avaliações baseadas em questionários, ou avaliações com um maior número de participantes, um experimento mais extenso, e com maior controle de variáveis. O primeiro é a **ausência de suficiente conhecimento prévio** sobre o uso deste paradigma de construção de sistemas de ajuda *online* por parte de pessoas não familiarizadas com a engenharia semiótica. Um dos objetivos do teste é justamente levantar mais conhecimento para que futuramente se possam realizar testes mais controlados, mais extensos, com uma variedade maior de participantes, e assim por diante. O segundo é a inexistência, neste momento de pesquisa, de suficiente infraestrutura — e.g. ferramentas instrumentadas de construção — para coletar, acervar e analisar um volume maior de dados. Providenciar esta infraestrutura é um de nossos próximos passos.

Em contrapartida, a vantagem desta avaliação informal — mesmo feita com um número reduzido de participantes, que possuíam apenas alguma informação básica sobre IHC e pouca experiência profissional — é identificar, a baixo custo, as oportunidades e necessidades para um próximo passo de pesquisa, no qual se possa realizar uma avaliação mais profunda e/ou mais extensa, centrada em tópicos específicos surgidos da presente experiência.

RESULTADOS OBTIDOS

Nas quatro sessões de teste realizadas, nenhum dos participantes encontrou dificuldades em entender ou em executar a tarefa proposta. Além disso, eles também não manifestaram problemas em compreender o paradigma da ajuda [3] e o aplicaram corretamente.

Apesar da longa duração do teste, nenhum deles desistiu antes de concluir a tarefa. Além disto, nas entrevistas que foram realizadas após concluírem, os participantes também manifestaram entusiasmo com o paradigma e comentaram sobre como a contextualização da ajuda a torna mais interessante e útil para os usuários.

Analisando os esboços produzidos durante os testes, foi possível observar uma grande predominância das informações operacionais e táticas sobre as estratégicas [2,3]. Isto é, todos os participantes apresentaram uma tendência por fornecer aos usuários informações relativas à definição de signos e à operacionalização da tarefa, deixando de lado as intenções do *designer*. Como nenhum dos participantes é o real desenvolvedor da aplicação, não se pode esperar que eles sejam capazes de identificar todas as intenções dos verdadeiros *designers*, mas ainda assim eles deveriam conseguir inferir algumas destas intenções.

Um exemplo de que isto poderia ser feito pode ser dado por uma das raras perguntas nos esboços da ajuda produzidos que envolviam informações estratégicas:

“Por que não posso ver a lista de pessoas que tiveram muitos matches?”

escreveu um dos participantes, referindo-se ao fato de que o ICQ mostra os resultados de uma busca por usuários apenas até um número máximo, pré-determinado pela aplicação.

Poder-se-ia esperar que houvesse mais perguntas ou observações deste tipo nos esboços produzidos, mas, ao contrário, os participantes mostraram uma grande preferência pelas perguntas operacionais ou táticas, como, por exemplo:

“O que é o número do ICQ?”.

Além da contextualização da ajuda, outro diferencial do paradigma avaliado em relação a sistemas de *help* comuns é a sua capacidade de dar aos *designers* os meios necessários para explicar aos usuários quais eram as suas intenções ao projetar algo e, assim, fazê-los compreender melhor a aplicação. O que se observou nos testes, contudo, foi que os participantes não aproveitaram esta oportunidade. Ao contrário, eles forneceram aos usuários predominantemente aquelas informações que normalmente já são encontradas em qualquer sistema de

ajuda. Ou seja, eles utilizaram o paradigma avaliado apenas para formatar estas informações de uma maneira diferente e mais fácil para o usuário acessar, mas não modificaram o conteúdo do *help* em si.

É interessante observar também que dois dos participantes, além de produzirem o esboço da ajuda, forneceram sugestões sobre mudanças que deveriam ser realizadas na interface da aplicação, o que pode indicar que o esforço de interpretar a comunicação *designer*-usuário com o objetivo de projetar o *help* para uma aplicação já pronta leva a uma reflexão sobre a mesma e, possivelmente, à sua reconstrução.

Um destes participantes, por exemplo, disse que:

“Eu explicaria que o “View My Details” não faz nada a não ser mostrar os meus dados. (Inclusive, não sei se deixaria esse botão nesse local da interface)”

mostrando claramente o seu questionamento em relação à decisão de projeto tomada.

Estes mesmos participantes forneceram também sugestões sobre como o paradigma poderia ser implementado. Isto é particularmente interessante, porque ele engloba uma arquitetura e um método para o desenvolvimento de sistemas de ajuda [3], mas não uma implementação definida.

Por fim, as perguntas que os participantes utilizaram podem ser, na maioria das vezes, mapeadas para aquelas que já aparecem no paradigma em questão. Ainda assim, entretanto, três perguntas diferentes e interessantes surgiram:

- Qual devo usar? — esta é relevante quando o usuário precisa fornecer uma informação para a qual há um conjunto de possíveis respostas;
- Por que não posso fazer isso? — o paradigma [3] contém a pergunta “Por que devo fazer isso?”, mas não a sua negativa, que também pode ser útil aos usuários; e, finalmente,
- Como isso funciona? — ao contrário da maioria das perguntas, que é centrada no usuário e no que ele deve fazer, esta é relevante quando o *designer* precisa explicar a ele como algum mecanismo da aplicação funciona e como isto pode impactar na forma através da qual o usuário deve se relacionar com ela.

CONSIDERAÇÕES FINAIS

Em relação aos objetivos iniciais desta avaliação, observou-se uma reação positiva dos participantes do teste em seu primeiro contato com o paradigma em questão. Mesmo sem conhecimentos profundos da teoria da engenharia semiótica ou do paradigma em si [3], eles foram capazes de entendê-lo, assim como de aplicá-lo corretamente para uma tarefa simples. Ademais, manifestaram entusiasmo com esta forma diferente de projetar sistemas de ajuda, sugerindo inclusive implementações distintas possíveis.

É importante enfatizar, entretanto, o fato de que esta avaliação baseou-se na interpretação da comunicação *designer*-usuário de uma aplicação já existente. Isto contraria o método de desenvolvimento sugerido [3], que propõe que a construção da ajuda ocorra ao longo do projeto da aplicação, já que o processo de coletar informações para o *help* invariavelmente gera uma reflexão sobre a aplicação em si.

O fato deste teste ter sido realizado sobre uma aplicação já pronta, contudo, não elimina esta reflexão. Ao contrário, a presença de sugestões de mudanças na interface em dois dos testes realizados talvez indique que a construção da ajuda para uma aplicação existente seja capaz de fornecer insumos para o re-projeto da mesma ou até de provocá-lo, na medida em que ela ajuda o *designer* a enxergar melhor o problema e a pensar mais sobre a sua solução.

Esta avaliação contribuiu ainda tanto para que se ratificasse o conjunto de perguntas que já faz parte do paradigma da ajuda — e que foi utilizado pelos participantes do teste — quanto para que se identificassem novas perguntas que talvez sejam necessárias para algumas aplicações.

Por fim, o motivo da ausência de informações estratégicas nos esboços de *help* produzidos pelos participantes merece consideração. Todos eles consideraram a contextualização da ajuda [3] como uma grande vantagem do paradigma avaliado, mas nenhum mencionou a oportunidade de fornecer informações que normalmente não se encontram em sistemas de ajuda como benefício.

O motivo para isto talvez se origine no fato de que os próprios participantes, por não estarem acostumados a ver as intenções dos *designers* sendo explicitamente explicadas em sistemas de ajuda comuns, naturalmente as deixaram de lado, preferindo informações mais diretamente relacionadas à operacionalização da tarefa.

Não se pode desconsiderar, contudo, a possibilidade deles terem excluído estas informações por não as acharem

relevantes, isto é, por acreditarem que os usuários não estão interessados em entender por que o *designer* projetou a aplicação de determinada forma, mas apenas em aprender como executar a tarefa desejada.

De forma geral, este teste foi útil para que se pudesse avaliar a reação de desenvolvedores de *softwares* sem conhecimentos profundos de IHC diante de um novo paradigma para a construção de sistemas ajuda [3]. Além disso, ele foi capaz de fornecer insumos para pesquisas mais profundas sobre a contribuição do projeto da ajuda em gerar reflexão sobre a aplicação como um todo — especialmente no caso de uma aplicação já pronta — e levantou questões sobre a presença de informações estratégicas [2,3] no conteúdo da mesma.

AGRADECIMENTOS

As autoras agradecem ao CNPq que, através de bolsa de produtividade em pesquisa e bolsas de Iniciação Científica e Tecnológica, ligada ao Projeto Oré, financia o desenvolvimento deste projeto. Agradecem também aos participantes anônimos e voluntários que gentilmente concordaram em participar dos testes relatados neste trabalho e assim contribuíram fundamentalmente para o andamento do projeto.

REFERÊNCIAS

1. Prates, R.O.; de Souza, C.S.; Barbosa, S.D.J.; “A Method for Evaluating the Communicability of User Interfaces.” *Interactions* 7, 1. New York, NY: ACM Press, 31-38, 2000.
2. de Souza, C.S.; Prates, R.O.; Carey, T. (2000) Missing and Declining Affordances: Are these Appropriate Concepts? . *Journal of the Brazilian Computer Society*. v.7, n.1, p.26 - 34, 2000
3. Silveira, M. S. Metacomunicação Designer-Usuário na Interação Humano-Computador: Design do Sistema de Ajuda. Tese de Doutorado, Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro. Dezembro de 2002.

Uma proposta para o desenvolvimento de interfaces de usuário multi-plataformas com tecnologia Web

Macilon Araújo Costa Neto

PPgSC-UFRN
Campus Universitário – Lagoa Nova - Natal
+55 84 215 3814
makz@ppgsc.ufrn.br

Jair C Leite

DIMAp-UFRN
Campus Universitário – Lagoa Nova - Natal
+55 84 215 3814
jair@dimap.ufrn.br

RESUMO

Este trabalho tem como objetivo apresentar uma proposta para o processo de desenvolvimento de interfaces de usuário para múltiplas plataformas, descrevendo os principais problemas com que se deparam os *designers* na realização desta tarefa. A solução aqui proposta consiste no desenvolvimento de interfaces para aplicações Web, utilizando-se das abordagens do W3C para independência de dispositivos.

Palavras-chave

Interface de Usuário multi-plataforma; Interfaces de Usuário Adaptáveis; Independência de Dispositivos.

ABSTRACT

The goal of this work is to show a proposal to cross-platform user interface development, stating the main problems faced by designers for achieving this task. The solution suggested by this approach consists of Web applications user interface development, making use the W3C's approach for device independence.

INTRODUÇÃO

A computação móvel e a variedade de novos dispositivos de interfaces de usuário têm propiciado novas possibilidades. Uma delas é o desenvolvimento de aplicações que possam ser utilizadas em qualquer lugar e a partir de diferentes tipos de dispositivos computacionais. Um cenário que ilustra possibilidades interessantes é poder acessar e editar um mesmo arquivo de trabalho em casa, utilizando um computador de mesa, no carro através de um PDA, ou numa reunião no escritório, utilizando uma tela de grande escala, com interação por controle remoto.

Diversos desafios precisam ser resolvidos para tornar este cenário uma realidade. Eles vão desde problemas de infraestrutura de comunicação, ao desenvolvimento de interfaces de usuário (IU) para cada um dos diferentes tipos de dispositivos. As aplicações e os seus dados devem permitir acesso remoto e visualização por diferentes tipos de IU. Isto requer uma infraestrutura de rede com pontos de acesso os mais amplos possíveis. É necessária ainda uma arquitetura de sistema cliente-servidor de forma que a aplicação e os dados possam ser

acessados a partir dos diversos clientes.

A Web oferece as condições necessárias para que uma aplicação desenvolvida possa ser acessada e utilizada a partir de diversos clientes. Cada cliente, no entanto, deve poder oferecer as condições adequadas de utilização da aplicação nos diferentes tipos de dispositivos que acessam a Web. A razão da opção pela Web é que já é possível o acesso a partir de uma variedade enorme de dispositivos como telefones celulares, TV, câmeras digitais e outros. Além disso, existe um grande esforço para o acesso a informações independente de dispositivos, em qualquer lugar, por qualquer um, a qualquer hora e de qualquer forma [4].

Existem várias propostas para o desenvolvimento de IU multi-plataforma. A estratégia mais simples, mas também a mais trabalhosa, é desenvolver diversas versões de uma mesma IU para cada uma das diferentes plataformas. Esta solução requer que o desenvolvedor conheça cada uma das plataformas alvo e as diversas variações que cada uma delas possa requerer. Este processo trabalhoso agrava-se ainda mais quando for necessária uma modificação da interface ou com o surgimento de um novo tipo de dispositivo.

Uma solução mais interessante é a criação de uma versão da IU que possa ser adaptada, de forma automática ou parcialmente automática, para cada um dos diferentes dispositivos alvos. Isto requer o desenvolvimento de uma versão da interface que seja independente de dispositivo e que possa ser transformada de modo a adequar-se às características de cada um dos dispositivos alvos.

O foco deste trabalho está nos desafios associados ao desenvolvimento de IU multi-plataformas utilizando a Web. Nosso objetivo é apresentar uma proposta de solução para que as aplicações Web possam ser utilizadas em diferentes dispositivos. Isto requer que as interfaces de usuário sejam adaptadas para os diferentes tipos de dispositivos de acesso à Web.

DESENVOLVIMENTO MULTI-PLATAFORMA

O principal problema ao se desenvolver interfaces para múltiplas plataformas é quantidade de fatores envolvidos e isso causa uma grande variabilidade do contexto de uso. O contexto de uso para sistemas interativos é definido por três classes de entidades [1]:

- **Usuário:** representa o estereótipo dos usuários do sistema, aqueles que pretendem usar e/ou que usam efetivamente o sistema;

- **Plataforma:** dispositivos computacionais (hardware e software) que podem ser usados para interação com o sistema;
- **Ambiente:** ambiente físico onde as interações podem ocorrer, isto envolve qualquer fator que possa influenciar o comportamento do sistema e/ou dos usuários.

Apesar do desenvolvimento de interfaces para múltiplas plataformas já ser algo bem concreto atualmente, ainda há muito por se fazer, para obter efetivamente interfaces com essas características. Os principais desafios são [1]: (i) a construção e manutenção de versões da interface, (ii) a checagem da consistência entre versões para garantir uma interação sem tortura e (iii) a implantação nessas versões da habilidade de responder dinamicamente às mudanças no ambiente, tais como conectividade da rede, localização do usuário e as condições de sonoridade ou iluminação do ambiente.

Este trabalho considera apenas os aspectos relativos à plataforma ou dispositivos. Em relação a este aspecto, um dos principais problemas é a variabilidade nas modalidades de entrada e saída. Dispositivos como celulares possuem apenas um teclado numérico e suporte à voz, os *palms* não possuem teclados, mas seus *displays* relativamente pequenos e resolução razoável são sensíveis ao toque e geralmente são providos com canetas para auxiliar na entrada de dados, as TVs interativas possuem apenas o controle remoto para permitir interações.

O impacto mais óbvio é a variabilidade no tamanho de tela dos dispositivos, onde um Desktop tem resolução de 640x480 até 1280x1024 numa tela de 8 até 20 polegadas, um celular tem uma resolução de 60x80 numa tela de duas polegadas, displays de parede como o do projeto i-Land, que tem uma resolução 3000x1000 numa tela de 4.5 metros de comprimento por 1.1 metros de altura e o da Universidade de Stanford com resolução de 3796x1436 numa tela de 1.83 metros de largura por 0.61 metros de altura [6].

Estas características demonstram que desenvolver IU multi-plataforma é lidar com inúmeras variáveis. As estratégias de desenvolvimento vão de um extremo, no qual o desenvolvedor cria diferentes versões de acordo com as características de cada um dos dispositivos alvos, a um outro extremo, com o desenvolvimento de uma versão independente de dispositivo que possa se adaptar automaticamente a cada um deles.

Mesmo sendo uma solução possível, o desenvolvimento de diferentes versões da interface da aplicação, tem tido sua viabilidade muito questionada, isto porque ela incorre em alguns problemas, como os destacados em [7]: (i) requer esforços extras no desenvolvimento e manutenção, aumentando os custos e complicando o gerenciamento de configuração; (ii) a proliferação de diferentes versões dilui os recursos disponíveis para

engenharia de usabilidade e requer uma cara manutenção da consistência da interface entre as várias plataformas; e (iii) a deficiência das arquiteturas e ferramentas de desenvolvimento em suportar a criação de aplicações capazes de mudar dinamicamente em resposta às mudanças em seu contexto de uso, ou que apenas compartilhem dados entre vários tipos de dispositivos.

A adaptabilidade é uma questão importante no desenvolvimento de aplicações para múltiplas plataformas. Assegurar esta capacidade às interfaces tem sido um grande desafio. Nesse sentido, o W3C tem empregado um grande esforço para prover esta característica à Web e assegurar a independência de dispositivos de suas aplicações. Nosso trabalho segue a proposta de adaptabilidade defendida pelo Grupo de Trabalho em Independência de Dispositivos do W3C.

UMA SOLUÇÃO BASEADA EM MODELOS COM ADAPTAÇÃO DINÂMICA

Nossa proposta para o desenvolvimento de interfaces multi-plataforma utilizará a adaptação dinâmica. Para isto, utilizaremos a abordagem de desenvolvimento baseado em modelos que requer a criação de modelos abstratos que possam ser mapeados em soluções específicas de cada plataforma de aplicação.

A abordagem de desenvolvimento baseado em modelos requer a construção de diversos modelos que descrevam o domínio de aplicação, as características dos usuários, as tarefas que os usuários podem realizar utilizando a interface e a estrutura e o comportamento da interface. Dentre as vantagens desta abordagem, está a possibilidade de mapear modelos abstratos, independentes de dispositivos, em modelos concretos, específicos de dispositivos. Esta estratégia vem sendo utilizada em desenvolvimento multi-plataforma. Puerta e Eisenstein consideram como abstratos os modelos de *usuários, domínio e tarefas*, e concretos, os de *apresentação e diálogo* [8].

Descrição independente de dispositivo com a IMML

O processo de adaptação dinâmica requer um mapeamento automático entre os modelos abstratos e concretos. Este mapeamento muitas vezes não é muito direto e tem gerado interfaces com resultados imprevisíveis e insatisfatórios [6]. Para minimizar este problema, nossa estratégia considera uma descrição independente de plataforma dos modelos do domínio, da interação e da apresentação. Cada um destes modelos tem por objetivo descrever, respectivamente, a funcionalidade, interatividade e comunicabilidade da interface. Esta descrição é feita numa linguagem baseada em XML, chamada IMML (*Interactive Message Modeling Language*) [3]. Embora existam diversas linguagens para a descrição de IU, muitas com propósito de desenvolvimento multi-plataforma e baseadas em XML, a IMML possui a vantagem de integrar os aspectos funcionais, interativos e comunicativos da IU numa descrição abstrata.

```

<task-environment>
  <command-panel>
    <select>
      <Activate Command="Set">
      <Activate Command="Print">
    </select>
  </command-panel>
</task-environment>

<Command name="Set" domain-function="Setting">
  <Join>
    <View> To set an application you must enter
    the information and ...</View>
    <Sequence>
      <Enter-info Domain-Object="Name" />
      <Select-info Domain-Object ="Type" />
      <Activate Control="Start" />
      <Activate Control="Cancel" />
    </Sequence>
  </Command>

```

Figura 1: Descrição em IMML da IU independente de dispositivo

O código IMML da figura 1 descreve uma IU na qual o usuário deverá selecionar dois comandos (*Set* ou *Print*). A escolha ocorre através da ativação de uma das opções. A descrição abstrata não determina como as duas opções de seleção serão apresentadas para o usuário, nem como elas serão ativadas. A escolha de uma delas ativa o comando que permite ao usuário fornecer os dados e executar a respectiva função (a figura 1 mostra apenas o comando da função *set*).

Perfis de dispositivos com CC/PP

O processo de adaptação dinâmica requer a geração de código dependente de dispositivo a partir da descrição da IMML e das informações do perfil de dispositivos. A descrição em IMML será processada para gerar uma descrição em uma linguagem específica de dispositivo, como XHTML, WML, etc.

Entretanto, apenas a descrição de uma linguagem específica de dispositivo não é suficiente. Uma mesma linguagem como a HTML pode ser utilizada em dispositivos com grandes variações de tamanho de tela, cores, formas de entrada de dados, etc. Neste caso, é necessário utilizar informações da plataforma alvo, de maneira que o código gerado na linguagem seja adequado às características do dispositivo.

A idéia é armazenar informações sobre cada dispositivo em um local público da Web e utilizando um vocabulário padrão. Para isto, iremos utilizar o CC/PP, um protocolo desenvolvido pelo W3C que oferece um vocabulário para a descrição de um perfil do *contexto de entrega*. Um contexto de entrega define um conjunto de atributos que caracteriza as capacidades de um dispositivo e as preferências do usuário [4].

O CC/PP (*Composite Capability/Preference Profile*) é uma coleção de informações sobre a capacidade da plataforma – hardware, software, aplicações – e as preferências específicas do usuário [2]. Estas informações podem incluir tipo do dispositivo (telefone, impressora, computador pessoal, etc.), linguagem preferida, exibição ou não de imagens ou sons, tamanho

da tela, banda de rede, versão da linguagem HTML e várias outras características.

O perfil CC/PP de um dispositivo é descrito em RDF (*Resource Description Framework*), a linguagem do W3C para modelagem de metadados – descrições informativas sobre itens na Web. Cada informação em RDF é sempre ligada a um endereço Web. Por exemplo, “*um dispositivo com um endereço www.dp.com/perfil tem o seguinte perfil...*” Isto significa que ao enviar um identificador URI para um perfil de dispositivo, todos os dados sobre ele tornam-se imediatamente disponíveis. Com base nestas informações, o servidor pode escolher as características específicas da IU que sejam mais adequadas aquele dispositivo.

A figura 2 mostra um pequeno trecho de uma descrição de perfil de dispositivo em CC/PP. Estão descritos as características de hardware, software e do browser, como por exemplo, que a tela para display tem 160x100 pixels, dentre outras. Com base nestas informações, o processo de adaptação dinâmica deverá determinar como as informações serão exibidas.

O processo e as regras de adaptação dinâmica

O processo de geração da IU com adaptação dinâmica ocorre em tempo de execução. À medida que o usuário está utilizando a aplicação, as partes da interface vão sendo montadas no servidor a partir das *regras de adaptação*.

Entretanto, para que a geração dinâmica funcione, é necessário que as regras de adaptação sejam geradas em tempo de design. Neste processo de design, o *designer* da IU irá ajustar a interface definindo um conjunto de regras de adaptação para cada perfil de dispositivo. São construídos protótipos de forma que se possa visualizar as saídas possíveis para cada tipo de dispositivo.

O processo de adaptação dinâmica em tempo de execução combina a descrição abstrata da IU, a descrição em CC/PP do dispositivo e as regras de adaptação. Cada requisição que um dispositivo faz a uma página (indicado pelo seu identificador URI), que seja parte das páginas que compõem a IU, envia junto um ponteiro, na forma de um outro identificador URI, que indica onde estão as informações do perfil específico do dispositivo em questão. O ponteiro URI vai direto para o banco de dados CC/PP.

A geração das IU específicas para cada um dos dispositivos utilizará ainda as características modulares das linguagens XHTML, CSS (*Cascading Style Sheets*) e SMIL (*Synchronized Multimedia Integration Language*). Esta característica permite que o servidor possa gerar dinamicamente diferentes versões da mesma interface. Por exemplo, pode-se gerar uma versão apenas com módulos de texto ou com total capacidade gráfica. Para isto, é necessário que as informações de perfis indiquem quais módulos de XHTML, CSS e SMIL serão suportados. O processo de geração automática deverá confrontar as informações do perfil da IU com as informações do perfil do dispositivo.

```

<?xml version="1.0"?>
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-
  rdf-syntax-ns#"
  xmlns:ccpp="http://www.w3.org/2002/11/08-
  ccpp-schema#"
  xmlns:ex="http://www.dp.org/schema#">

  <rdf:Description
    rdf:about="http://www.dp.org/perfil#MeuPerf
    il">

    <ccpp:component>
      <rdf:Description
        rdf:about="http://www.dp.org/perfil#Termina
        lHardware">
        <rdf:type
          rdf:resource="http://www.dp.org/schema#Hard
          warePlatform" />
        <ex:displayWidth>160</ex:displayWidth>
        <ex:displayHeight>100</ex:displayHeight>
        </rdf:Description>
      </ccpp:component>

      <ccpp:component>
        <rdf:Description

          rdf:about="http://www.dp.org/perfil#Termina
          lSoftware">
          <rdf:type
            rdf:resource="http://www.dp.org/schema#Soft
            warePlatform" />
          <ex:name>EPOC</ex:name>
          <ex:version>2.0</ex:version>
          <ex:vendor>Symbian</ex:vendor>
          </rdf:Description>
        </ccpp:component>

        <ccpp:component>
          <rdf:Description

            rdf:about="http://www.dp.org/perfil#Termina
            lBrowser">
            <rdf:type

              rdf:resource="http://www.dp.org/schema#Brow
              serUA" />
              <ex:name>Mozilla</ex:name>
              <ex:version>5.0</ex:version>
              <ex:vendor>Symbian</ex:vendor>
              <ex:htmlVersionsSupported>
                <rdf:Bag>
                  <rdf:li>3.2</rdf:li>
                  <rdf:li>4.0</rdf:li>
                </rdf:Bag>
              </ex:htmlVersionsSupported>
              </rdf:Description>
            </ccpp:component>

          </rdf:Description>
        </rdf:RDF>

```

Figura 2: Exemplo perfil de dispositivo em CC/PP

CONCLUSÃO

Este trabalho apresentou uma proposta para o desenvolvimento de interfaces de usuário (IU) multi-plataforma. Esta proposta utiliza as tecnologias para aplicações Web, baseadas no modelo cliente-servidor.

A solução adotada utiliza uma descrição da IU independente de dispositivo que, junto com a descrição do perfil do dispositivo (contexto de entrega), permite a definição, em tempo de design, de regras de adaptação. A descrição independente de dispositivo é feita em IMML, uma linguagem de descrição de interface que permite descrever os modelos de domínio, de interação e de apresentação. Os perfis de dispositivos são feitos utilizando o padrão CC/PP proposto pelo W3C. As regras de adaptação são utilizadas por um programa no lado-servidor responsável pela geração dinâmica da IU, em tempo de execução.

Esta solução ainda está nos seus estágios iniciais e várias questões sobre as linguagens e tecnologias ainda estão abertas.

REFERÊNCIAS

1. Calvary, G.; Coutaz, J.; Thevenin, D.; Limbourg, Q.; Bouillon L.; Vanderdonckt, J. "A unifying reference framework for multi-target user interfaces". *Interacting with Computer*, 15, 289-308, 2003.
2. Klyne, G.; Reynolds, F.; Woodrow, C.; Ohto, H.; Hjelm, J.; Butler, M.; Tran, L. "Composite Capability/Preference Profiles (CC/PP): Structure and Vocabularies 1.0", W3C Recommendation 15 January 2004, This version: <http://www.w3.org/TR/2004/REC-CCPP-struct-vocab-20040115/>
3. Leite, J. C. "Specifying a User Interface as an Interactive Message" HCI International 2003 - 10th International Conference on Human-Computer Interaction, Creta, 2003.
4. Lewis, R. "An Introduction to Device Independence", 2001. <http://www.w3c.org/2001/di/IntroToDI.html>.
5. Lewis, R. "Glossary of Terms for Device Independence", W3C Working Draft 25 August 2003, version: <http://www.w3.org/TR/2003/WD-di-gloss-20030825/>
6. Myers, B.; Hudson, S.; Pausch, R. "Past, Present and Future of UI Software Tools" *ACM Transactions on Computer-Human Interaction*, Vol. 7, No. 1, March 2000, Pages 3–28.
7. Paternò, F.; Santoro, C. A Unified Method for Designing Interactive Systems Adaptable to Mobile and Stationary Platforms, *Interacting with Computers*, Vol.15, N.3, pp 347-364, Elsevier, 2003.
8. Puerta, A.; Eisenstein, J. "Towards a general computational framework for model-based interface development systems", *Proceedings of the 4th international conference on Intelligent user interfaces*, 1999, ACM Press, 171--178.

Utilizando ontologia para a descrição da interação em componentes de interface de usuário

Lirisnei Gomes de Sousa

lirisnei@consiste.dimap.ufrn.br

Jair Cavalcanti Leite

jair@dimap.ufrn.br

DIMAp-UFRN
Campus Universitário – Lagoa Nova - Natal
+55 84 215 3814

RESUMO

As técnicas do Desenvolvimento de Software Baseado em Componente (DSBC) têm sido utilizadas no desenvolvimento de interfaces de usuário (IU). Visando a melhoria do DSBC surgiram algumas técnicas e ferramentas para auxiliar o processo de busca e recuperação dos componentes, mas estas técnicas consideram os componentes de IU da mesma maneira que os outros componentes, desconsiderando aspectos de interação entre o usuário e os componentes de IU. Este trabalho apresenta uma ontologia para a melhoria do processo de busca e recuperação de componentes de IU, onde será possível classificar componentes de IU baseando-se também em aspectos de interação.

ABSTRACT

The Component Based Software Development (CBS) techniques have been applied to improve User Interface (UI) development. CBS requires some techniques and tools to help component search and retrieval. However, these techniques do not consider that the UI component is a special kind of component. It is necessary to consider the interaction aspects between the user and the UI components. This work presents an ontology to improve the search and the retrieval of UI components. Using this ontology it will be possible to classify UI components taking in account interaction aspects.

Palavras-chave

Ontologia, Recuperação de componentes, Interface de Usuário, Componentes de Interface de Usuário.

INTRODUÇÃO

No Desenvolvimento de Software Baseado em Componente (DSBC) um software é criado conectando-se partes, chamadas de *componentes*. Segundo Hopkin [2] um componente é uma unidade de software bem definida e que pode ser utilizada em conjunto com outros componentes para formar grandes unidades de software. Sugumaram [8] relata algumas vantagens que foram identificadas nos softwares desenvolvidos com componentes, tais como: mais confiáveis; desenvolvidos em menor espaço de tempo; menor custo de

desenvolvimento. Um componente tem funcionalidade própria para resolução de um problema específico que é oferecida através de uma interface que permite acessar os seus *métodos*. Além dos métodos um componente possui *eventos e propriedades*.

As técnicas de DSBC também são utilizadas no desenvolvimento de interface de usuário (IU), proporcionando também redução no tempo desenvolvimento. Os componentes de IU proporcionam ao desenvolvedor de UI um trabalho mais fácil, pois o ele, na maioria dos casos, está livre da codificação de componentes. A IU é desenvolvida juntando-se e ajustando-se pequenas partes como caixa de seleção, botão, caixa de texto, etc. Os componentes de IU também são chamados de *widgets*.

Os componentes de IU são projetados para resolver problemas de interação entre o usuário e o sistema. A interação pode ocorrer de várias formas tais como entrada de dados, seleção de opções em menus, caixas de checagem, perguntas ao usuário, etc. A descrição das formas de interação que o componente proporciona determina o seu *modelo de interação*. Estes aspectos são fundamentais na aplicação de um componente de IU, pois serão determinantes para a usabilidade do sistema.

Um componente, além de ser extensível e reutilizável também precisa ser alcançável através de mecanismos de busca e recuperação do mesmo. O processo de busca e recuperação é um dos grandes problemas do DSBC. Em alguns casos, ele utiliza linguagens de consultas muito restritas e está baseado no casamento de padrões, comparando palavras-chaves (da busca) com *strings* contidas na descrição dos componentes [4].

Nos últimos anos, algumas pesquisas apresentaram soluções com o uso de conteúdo semântico associado aos componentes descritos através de ontologias. Uma ontologia é utilizada para descrever conceitos e relacionamentos entre os mesmos. Sugumaram apresenta uma solução que se baseia no uso de ontologia e na criação de um conteúdo semântico para os componentes [8], possibilitando uma busca mais inteligente. A utilização de ontologia apresenta bons resultados na busca e recuperação de componentes, pois facilita a criação de ferramentas que permitam buscas em linguagens mais naturais.

VI Simpósio sobre Fatores Humanos em Sistemas Computacionais — Mediando e Transformando o Cotidiano. Hotel Bourbon, Curitiba, 17 a 20 de outubro de 2004. UFPR, CEIHC—SBC.

As ontologias normalmente utilizadas na busca e recuperação de componentes não oferecem soluções adequadas para os componentes de IU. Os mecanismos tratam os componentes de IU como qualquer outro componente, não considerando aspectos de interação entre o componente de IU e o usuário (final). Entretanto, é fundamental o conhecimento do modelo de interação do componente de forma que o desenvolvedor possa prever aspectos de sua usabilidade.

A descrição do modelo de interação através de uma ontologia pode permitir a busca e a recuperação de forma mais eficiente, considerando as necessidades de interação e contribuindo para a usabilidade da aplicação. Por exemplo, podemos ter dois componentes de IU que tenham os mesmos métodos, eventos e propriedades, mas que apresentem diferentes mecanismos de interação com o usuário.

Este trabalho apresenta uma ontologia para a melhoria do processo de busca e recuperação de componentes de IU que considera, além dos aspectos operacionais dos componentes (métodos, propriedades e eventos), aspectos de interação entre o usuário e o componente de IU. Os restantes do trabalho estão estruturados da seguinte forma. A seção 2 apresenta alguns mecanismos que foram desenvolvidos para a recuperação de componentes. A seção 3 descreve alguns aspectos sobre componentes de IU. A seção 4 descreve a ontologia que foi desenvolvida para descrever conceitos de interação entre o usuário e os componentes de IU. Finalmente, a seção 5 contém as conclusões deste trabalho.

BUSCA E RECUPERAÇÃO DE COMPONENTES

A procura por mecanismos para auxiliar o processo de busca e recuperação de componentes tem sido um dos grandes interesses dos pesquisadores que trabalham para melhorar o Desenvolvimento de Software Baseado em Componentes. A seguir estão descritas algumas técnicas utilizadas para a busca e recuperação de componentes.

A busca por palavras-chaves baseia-se no casamento entre os padrões (strings) especificados, pelo desenvolvedor que procura os componentes, no momento da busca e as palavras (métodos, eventos, etc) do componente. Mili [4] propõe uma solução que faz o uso de especificações formais, utilizando lógica de primeira ordem, para descrever aspectos funcionais dos componentes. Assim, quando o desenvolvedor submete uma busca, além do casamento de padrões é feita inferências para saber se os componentes satisfazem algumas premissas. Esta técnica apresenta deficiência, pois o resultado contém muitos componentes não esperados e deixa de fora outros que seriam importantes.

A classificação de faces, proposta por Pietro-Diaz [6], baseia-se na idéia de que os atributos dos componentes são compostos por dois objetos: (1) o texto, que é a descrição textual do atributo em linguagem natural, e (2)

o conjunto de valores que podem ser atribuídos ao atributo. Estes valores são definidos em um vocabulário de atributos. Assim, as buscas podem ser feitas utilizando, além das palavras-chaves, um intervalo de valores atribuíveis ao atributo. Esta abordagem apresenta melhoria na classificação dos componentes, mas esta melhoria está limitada aos valores dos atributos.

O uso de semântica na busca e recuperação de componentes permite a resolução de alguns problemas existentes nas técnicas anteriores e também viabiliza o uso de linguagens mais naturais. Sugumaram [8] propõem uma abordagem que faz uso de domínios, processos, ações, atores e ontologias de termos de domínios. Nesta abordagem é possível a criação de consultas com uma linguagem natural através de uma interface Web. As consultas são processadas fazendo (1) uso das ontologias de domínio e (2) e buscas nos repositórios de componentes, que são implementados em JavaBeans. As ontologias de domínio descrevem termos do domínio onde o componente será utilizado. Por exemplo, se um componente é utilizado para processar lances em um leilão, a ontologia deve conter descrições semânticas tais como: "o que são lances em um leilão", "por quem são feitos", etc.

UMA ONTOLOGIA PARA COMPONENTES DE IU

Nesta seção apresentamos uma ontologia para o modelo de interação de um componente de IU que define conceitos relacionados à interação entre o usuário e o componente.

A ontologia foi descrita utilizando a OWL [5] e pode ser representada graficamente através de um grafo, como os exemplificados nas Figuras 1 e 2, que mostram os conceitos e os relacionamentos entre os mesmos. De acordo com a OWL [5], cada elipse representa um conceito e as setas representam os relacionamentos entre os conceitos. A elipse de onde a seta parte é o *sujeito*, a seta representa o *predicado* e a elipse de chegada da seta é o *objeto*.

A ontologia que descreve o modelo de interação está representada na Figura 1. Esta ontologia foi baseada nos conceitos propostos pela IMML (*Interactive Message Modeling Language*) [3]. Baseando-se na abordagem da Engenharia Semiótica, de que uma interface é uma mensagem, formada por signos, que o designer envia ao usuário [1], a IMML é uma linguagem que suporta a descrição desta mensagem. Desta forma, a IMML propõe um modelo conceitual para a IU que é dividido em: *modelo de domínio*, *modelo de interação* e *modelo de apresentação*. Com estes modelos é possível descrever as interações, os comandos, e as tarefas que o usuário deve desempenhar. Neste trabalho foram utilizados os conceitos mapeados pelo modelo de interação e pelo modelo de apresentação.

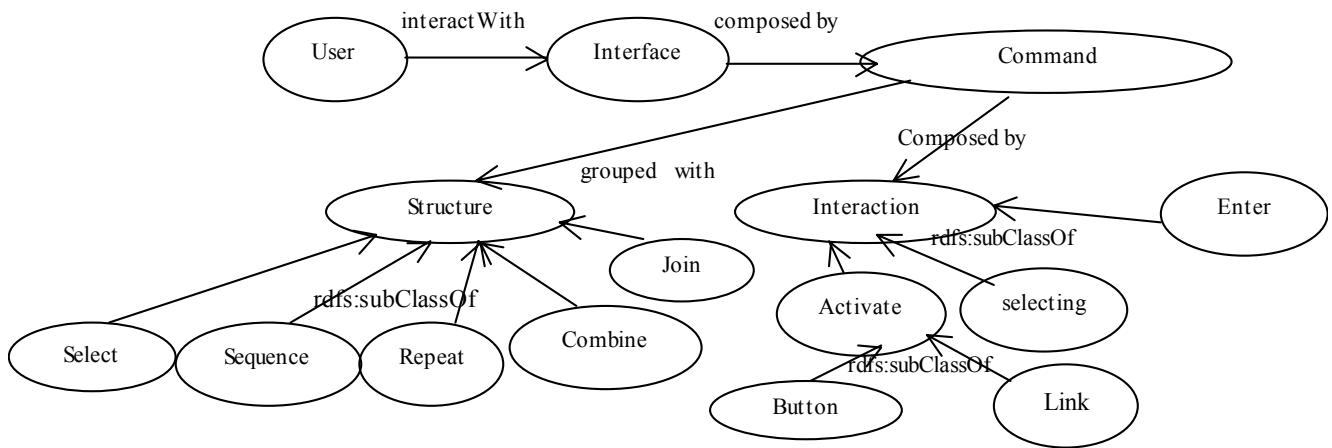


Figura 1: Ontologias representando a interação entre o usuário e os componentes de IU

Como mostra a Figura 1, os conceitos iniciam no usuário que interage (**interactWith**) com a interface do componente, e a interface é composta de um ou mais *comandos*. Normalmente, quando um componente é composto de outros componentes menores, a interface deste componente maior possui vários comandos. Um **comando** é composto por interações básicas ou estruturadas. Exemplos de interação básica são **enter-information**, **select-information** e **activate**. As interações básicas podem ser compostas por elementos de estruturação: *seqüência (sequence)*, indicando que as interações devem ser feitas de forma seqüencial; *seleção (select)*, indicando que o usuário pode escolher entre as interações disponíveis; *repetição (repeat)*, no qual as interações podem ser repetidas; *combinação (combine)*, indicando uma dependência semântica ou sintática entre as interações e *agrupamento (join)* onde as interações podem ser realizadas de forma livre.

A ontologia permitirá buscas por componentes de IU que ofereçam determinados tipos de interação. Por exemplo, o desenvolvedor de IU pode procurar componentes que tenham um campo de inserção de dados seguido de uma caixa de seleção e de um botão.

A ontologia também possui uma parte que descreve as características estruturais dos componentes, baseadas na proposta de Sugumaram [8]. Esta parte dá suporte às buscas que são baseadas em aspectos estruturais. Por exemplo, quando o desenvolvedor de IU está procurando um componente que resolve uma fórmula matemática, a busca pode ser feita procurando um componente com o nome de *form* ou *exp*.

Como mostra a Figura 2, um pacote possui uma propriedade chamada **hasComponent** cujo o valor é um componente. O componente, por sua vez, pode ter duas propriedades chamadas de **hasInterface** e **hasSignature**, onde estas têm como valor uma interface e a assinatura do componente respectivamente. Definimos que a assinatura de um componente possui *métodos*, *eventos* e *atributos*. A ontologia também descreve sinônimos para os conceitos. Por exemplo, o conceito **package** possui dois sinônimos que são **directory** e **library**.

O desenvolvimento de ontologias requer alguns cuidados para evitar ambigüidade. Uma ontologia representa conceitos, e cada conceito (ontologia) é identificado por palavra(s). As palavras podem ser ambíguas, mas os conceitos não. No desenvolvimento da nossa ontologia foram necessários ajustes para evitar ambigüidade, na representação de alguns conceitos. Um exemplo é o fato de que o conjunto dos nomes dos métodos e das propriedades de um componente recebe alguns nomes tais como: *assinatura*¹, *assinatura da interface* e *interface*. A imagem gráfica de um componente que é chamada de *interface gráfica* também pode ser chamada só de *interface* do componente. Com isso, “a interface de um componente de IU” possui dois sentidos, um referente aos métodos e outro referente à imagem gráfica, com a qual o usuário interage. Neste caso foi decidido que na segunda parte, relacionada aos aspectos estruturais, o conjunto de métodos será referenciado das seguintes formas: *assinatura*, *assinatura dos métodos*. Na parte relacionada à interação, a interface gráfica do componente será referenciada das seguintes formas: *interface gráfica*, *interface*, *interface do usuário*, *interface do componente*.

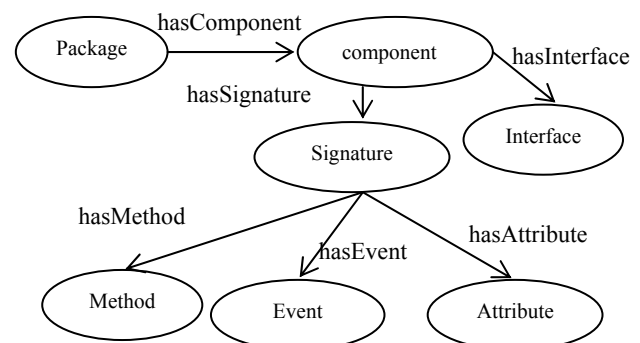


Figura 2: Ontologia das características estruturais dos componentes de IU

¹ As ontologias foram descritas em inglês (como mostram as Figuras 1 e 2), os nomes em português são ilustrativos.

Para o processo de busca e recuperação de componentes utilizando ontologias, além da definição das ontologias, também é necessário um conteúdo semântico contendo *instâncias* dos conceitos representados nas ontologias. Para o uso das ontologias apresentadas neste trabalho, o conteúdo semântico referente às características estruturais do componente será *baseado* na linguagem XICL [7], ou seja, o uso da linguagem seguindo as regras da OWL para a *instanciação* dos conceitos. O conteúdo semântico para a representação da interação de um componente de IU está baseado na IMML, também segundo as regras da OWL.

O processo de busca e recuperação dos componentes

O processo de busca e recuperação de componentes através do uso de ontologias é feito por ferramentas que fazem a interpretação das buscas e do conteúdo semântico associado aos componentes. O conteúdo semântico de um componente contém informações sobre os aspectos de interação de sua interface, e possivelmente de alguns aspectos estruturais. Assim, é importante que o desenvolvedor descreva semanticamente todas as possíveis maneiras que o usuário pode interagir com o componente. Estas informações serão as instâncias dos conceitos descritos anteriormente. Por exemplo, se a interface de um componente apresentar dependências semânticas é importante que todas estas dependências sejam descritas com o uso de **combine**. Se as descrições semânticas forem bem feitas as recuperações serão bem sucedidas.

Atualmente, estamos desenvolvendo um sistema Web para a busca e recuperação de componentes, onde esta ontologia será utilizada. A busca por componentes poderá ser feita através de consultas em linguagem natural restrita. A consulta será analisada pelo *módulo de processamento de consulta* e, depois, os componentes serão recuperados através de *inferências* sobre o conteúdo semântico associado aos componentes definido na ontologia. Primeiramente utilizaremos os componentes desenvolvidos na linguagem XICL [7]. O uso dos componentes XICL se deve a dois fatores: (1) a nossa experiência com a linguagem e (2) e ao fato da XICL ser baseada em XML, o que facilita o seu processamento.

CONCLUSÕES

Este trabalho apresentou uma ontologia para melhoria do processo de busca e recuperação de componentes de interface de usuário (IU). A ontologia considera aspectos de interação entre o usuário e o componente de IU de

modo a permitir que o desenvolvedor realize a busca e recuperação de componentes de acordo com os mecanismos de interação entre o usuário e os componentes de IU.

A ontologia melhora o processo de busca, pois permite a recuperação utilizando aspectos semânticos relacionados à interação. Além disso, ela possibilita a criação de mecanismos de inferência que permite uma busca inteligente, verificando termos sinônimos e consultas em linguagem natural.

Uma ferramenta está sendo desenvolvida para realizar a busca e recuperação de componentes utilizando a ontologia descrita. Também faremos testes para avaliar o processo de busca e recuperação com a ontologia, e, possivelmente, melhorar a representação dos conceitos de interação.

AGRADECIMENTOS

Os autores agradecem a Agencia Nacional do Petróleo – PRH 22 pelo apoio financeiro a este trabalho.

REFERÊNCIAS

1. de Souza, C.S. The Semiotic Engineering of user interface language design, in Int. J. of Man-machine Studies, 1993.
2. Hopkin, J. “Component Primer”, Communications of the ACM, 2000. Vol. 43, No. 10, pp. 27-30.
3. Leite, J.C. “Specifying a User Interface as an Interactive Message” HCII’2003, Creta, 2003.
4. Mili, A., Mili, R. Mittermeir, R. “Storing and Retrieving Software Components: A Refinement-Based System” IEEE Transactions on Software Engineering, 1994. Vol. 23, No 7, pp. 445-460.
5. Smith, M.K., Welty, C., McGuinness, D.L. “OWL – Web Ontology Language” W3C Recommendation 10 de Fevereiro de 2004, 2004.
6. Pietro-Diaz, R., Freeman, P. “Classifying Software for Reuse”, IEEE Software, 1987. Vol. 4 No. 1, pp. 6-16.
7. Sousa, L. G., Leite, J. C. “XICL – uma linguagem para o desenvolvimento de interfaces de usuário e de seus componentes” CLIHC 2003 - Latin American Conference on Human-Computer Interaction, Rio de Janeiro, 2003.
8. Sugumaram, V., Sotrey, V.C. A (2003) “A Semantic-Based Approach to Component Retrieval” The DATA BASE for Advances in Information Systems – Summer 2003, Vol. 34, No.3.

Oficinas

Designing Interfaces for Interactive Television

Felipe Afonso de Almeida

Davi D'Andréa Baccan

Giuliano Araujo Bertoti

felal@ida.liu.se

luke@widesoft.com.br

gbertoti@hotmail.com

LINCOM - Laboratório de Interação, Comunicação e Mídia
Divisão de Ciência da Computação
Instituto Tecnológico de Aeronáutica

ABSTRACT

For decades the television has been a part of the technological advances. However, new technologies related to TV (such as digital television as well novel set-top boxes capabilities) have enabled television producers and broadcasters to bring some interactivity into TV programmes. Thus, Traditional Television has become an Interactive Television (iTV). Essentially, an iTV is a result of the convergence between TV and computer, incorporating features of both. As the same way that this new media brings novel opportunities for all the players belonging to this setting, it brings a set of challenges that HCI community has to deal. How does one design interfaces for interactive television?

Keywords

Interfaces, Interactive Television.

INTRODUCTION

Since the last decade, the interest in interactive television (iTV) has grown steadily in the academic and industrial communities, some has been conducted to bring interactivity into television. iTV can be classified as a service that can be accessible on the TV set that differs from continuous succession of broadcast programmes, whether relating to broadcast programmes and based on iTV technologies.

Briefly, an iTV system can be viewed as a system that is resultant of the convergence between television and computing systems, incorporating features of both. Thus, an iTV programme is, essentially, a video programme which incorporates some style of interactivity.

Indeed, iTV programmes can offer a wide variety of opportunities to make the viewer truly interact with the TV programme. This interaction can lead to a simple textual information, services related to t-commerce (electronic transactions conducted via TV), or t-learning (distance learning via TV), and so forth. As the same way that iTV brings new and interesting possibilities, novel challenges and questions arise. Indeed, it has a lot of consequences for both the nature of television and further for the way people watch television.

Technological advances related to iTV will ensure this medium and its possibilities, but thinking about designing the interfaces is essential. Indeed, the challenge resides in how these interfaces will be designed.

GOALS

The goals of this workshop are to bring together researchers from different areas of knowledge such as psychology, arts, anthropology, computing, sociology,

linguistics, philosophy, ergonomics, semiotics and communication, in order to provide a mixed community of researchers to discuss the impact of the iTV setting in the human being. It will focus mainly on outlining guidelines to bring the design of interfaces for this new media less emphasis on purely technological issues. In addition, it is expected to bring the design of iTV interfaces to be a issue that HCI Brazilian community will be concerned with.

This workshop proposal is based on the firm belief that technologies will converge and the TV, both content and TV set can be a portal to access a whole range of new possibilities. It will change the way people watch a TV programme and will allow a truly interaction between TV viewers and the programme being screened. The shape of this portal is yet to be defined. But the authors feel that HCI Brazilian community has a major responsibility in the process of designing interfaces for this new media as well as dealing with their associated challenges.

Content

During the workshop, problems with the use of iTV interfaces will be addressed and guidelines will be established to the design of future ones. The issue of interfaces will be addressed within this context while the full scope of issues will depend on the contribution of the participants. The workshop will cover a wide range of topics with the following questions acting as guidelines:

- What are the features that make the TV set a familiar and usable device?
- Could the iTV interface preserve the easy interface that existing TV provides?
- Should iTV designers maintain the existing interfaces?
- What new metaphors can be used to design iTV interfaces?
- What input devices can be used to make TV viewer really interact with TV?
- Do iTV interfaces require major cognitive efforts to be learned than existing TV interfaces? Why?

Organizers' Background

Felipe Afonso de Almeida, PhD., is assistant professor at the Department of Computer Science of Technological Institute of Aeronautics (ITA), São José dos Campos, Brazil. He is the leader of the Laboratory for Interaction, Communication and Media - LINCOM. Felipe has worked on the theoretical aspects of

Interactive TV systems. He is responsible for Interactive Television and Interactive Digital Media postgraduate level disciplines.

Davi D'Andréa Baccan is Msc. student in Computer Science at ITA (funding by CNPq). Davi received his B.S. degree in Computer Science at School of Engineering of Piracicaba (EEP), Piracicaba, Brazil. He is a researcher at LINCOM. His currently research includes: Virtual, Mixed and Augmented Reality, Social Computing, Computer-Mediated Communication, and Interactive Television. He is teacher assistant of professor Felipe of Interactive Television and Interactive Digital Media disciplines.

Giuliano Araujo Bertoti is Msc. student in Computer Science at ITA (funding by CAPES). Giuliano received his B.S. degree in Computer Science at University of Mogi das Cruzes (UMC), Mogi das Cruzes, Brazil. He is a researcher at LINCOM. His currently research includes: Software Architecture and Interactive Television. He is teacher assistant of professor Felipe of Interactive Television and Interactive Digital Media disciplines.

ACKNOWLEDgements

The authors Davi D'Andréa Baccan and Giuliano Araujo Bertoti would like to thank, respectively, CNPq and CAPES.

REFERENCES

1. Baccan, D.D., Almeida, F.A., Bertoti, G.A., Fortes, R.S., Developing content for iTV: a case study. *2nd European Conference on Interactive Television*, University of Brighton, UK, 2004. In prelo.
2. Consulting, S., "iTV Handbook: Technologies and Standards", Prentice Hall, 1ª Edição – 2003.
3. Eronen, L. and Vuorimaa, P. User interfaces for digital television: a navigator case study. *Proceedings of the working conference on Advanced visual interfaces*, ACM Press (2001), 276-276.
4. van Vliet, H., Where Television and Internet meet... New experiences for rich media. <http://comcom.kub.nl/e-view/02-1/vliet.htm>
5. Jääskeläinen, K. Strategic Questions in the Development of Interactive Television Programs. *Master's thesis*. University of Art and Design Helsinki.
6. Jones, O. D. and Carey, R. Interactive television: strategies for designing useful and usable services. *CHI '01 extended abstracts on Human factors in computing systems*. ACM Press (2001), 503–503.
7. de Vos, L. Searching for the Holy Grail: Images of Interactive Television. Technical Report, University of Utrecht, Department of Media and Communication (2000), Netherlands. <http://www.globalxs.nl/home/l/ldevos/itvresearch>
8. Quico, C. Interactive television - a new media industry in portugal? Analysis of the current and future development of products and services. *Master's thesis*. Universidade Nova de Lisboa (2000).
9. Shneiderman, B., Universal usability, *Communications ACM*, ACM Press, pp. 84-91, 2000.

Artigos Aceitos para Participação

Acessibilidade na TV Digital para Pessoas com Necessidades Especiais

Andréa dos Santos Rodrigues (Faculdade Área1) Guido Lemos de Souza Filho (Universidade Federal da Paraíba)

Desenvolvimento de Conteúdo Educacional direcionado à Escola de Ensino Fundamental mediatizada pela TV Digital Interativa

Sergio Ferreira do Amaral (Faculdade de Educação, UNICAMP)

Função Fática e Interatividade no Discurso Televisivo

Guilherme Jorge de Rezende (UFSJ)

Programa de "Eduteinment" para TV Interativa

Gil Barros, Alexandra C. Alves (LSI, Universidade de São Paulo)

Televisão Digital e Interativa: a modelação social como variável na avaliação de usabilidade

Manuel José Damásio (Universidade Lusófona de Humanidades e Tecnologias)

Uma arquitetura para criação de interfaces adaptativas para televisão interativa

Fabrizio Jailson Barth, Edson Satoshi Gomi, (Laboratório de Engenharia de Conhecimento (KNOMA), Escola Politécnica da Universidade de São Paulo)

Uma Proposta para Formalização de Ciência de Contexto: Uma Abordagem Aplicável a Ambientes de Televisão Interativa

João Benedito dos Santos Junior (PUC Minas)

Design e Avaliação de Interfaces para Ambientes Educacionais

Raquel Oliveira Prates

Instituto de Matemática e Estatística - UERJ
60. andar – Bloco B
R. São Francisco Xavier, 524
Maracanã, Rio de Janeiro, RJ, 20550 – 013
raquel@ime.uerj.br

M. Cecília C. Baranauskas

Instituto de Computação e Núcleo de Informática
Aplicada à Educação - UNICAMP
Av. Albert Einstein 1251
Cx. Postal 6176 13083-970 Campinas, SP
cecilia@ic.unicamp.br

RESUMO

Tecnologias da informação e comunicação têm sido a base de ambientes educacionais especialmente nos últimos anos. Tais tecnologias viabilizam espaços de interação capazes de favorecer determinados comportamentos sociais e influenciar nos processos de aprendizagem, através das interfaces desses sistemas. O objetivo desta oficina é reunir profissionais e pesquisadores atuando no desenvolvimento e avaliação de interfaces para ambientes educacionais, para discutir suas contribuições. Desta forma, a oficina poderá promover um maior conhecimento do que está sendo feito na área bem como uma identificação mais precisa do que ainda necessita-se fazer. A partir desta discussão espera-se possibilitar uma maior colaboração entre pessoas e projetos na área.

ABSTRACT

Information and Communication Technologies have been the basis for educational systems specially in the last years. The interfaces of these technologies support different social behavior and learning processes through interaction. The goal of this workshop is to bring together professionals and researchers working on interface design and evaluation for educational systems. The workshop will allow for participants to enhance their knowledge of the work being done in the area in Brazil, as well as identify issues that need further discussion and work. The workshop intends to define a common ground for collaboration among participants and their projects.

Palavras-Chave

Interface de Usuário, Sistemas e Ambientes de Aprendizagem, Design e Avaliação de interfaces.

Introdução

Um dos grandes desafios da educação na atualidade é integrar as Tecnologias da Informação e Comunicação (TIC) às práticas de ensino exercidas em escolas e universidades, bem como em contextos não formais e ambientes de trabalho. Ambientes educacionais baseados em TIC têm sido usados tanto para apoiar e complementar o ensino feito em sala de aula, quanto para possibilitar o ensino a distância e atividades de formação profissional. Estas novas práticas permitem que alunos

VI Simpósio sobre Fatores Humanos em Sistemas Computacionais — Mediando e Transformando o Cotidiano. Hotel Bourbon, Curitiba, 17 a 20 de outubro de 2004. UFPR, CEIHC—SBC.

tenham experiências novas e enriquecedoras, como por exemplo simulando experiências de laboratório, que não seriam possíveis em um laboratório real de uma escola, ou interagindo com outros estudantes em trabalho colaborativo através de sistemas computacionais. O ensino a distância, por sua vez, possibilita que pessoas que não tinham acesso ao ensino, devido ao horário de trabalho ou ao local em que residem, possam investir na sua educação. Desta forma, fica patente a abrangência de ambientes educacionais e a sua importância ao permitirem que se eleve a qualidade da formação e se ofereça à sociedade um acesso mais amplo à educação, principalmente em países em desenvolvimento como o Brasil.

Um dos aspectos determinantes do sucesso de um software é a qualidade da sua interface com os usuários. A disciplina de IHC tem investido em teorias e práticas que possam fundamentar e apoiar o projeto e desenvolvimento de interfaces. Ambientes educacionais apresentam especificidades e desafios para o projeto e avaliação de software [2], em particular para suas interfaces. Estas devem, além de oferecer boa usabilidade [5] e comunicabilidade [4], possibilitar que o usuário aluno atinja o seu principal objetivo: aprender. Para isso estas interfaces devem ser criadas de modo a possibilitar ao aluno construir o conhecimento desejado e atingir os objetivos de aprendizado pretendidos pelo educador, e serem avaliadas para se verificar se alcançam seu objetivo também do ponto de vista educacional.

A importância dos ambientes educacionais baseados em TIC definiu a área de Informática e Educação, que busca justamente discutir e permitir a criação de software educacional de qualidade, conforme apontam as principais conferências nacionais e internacionais na área (Workshops de Informática na Escola WIE-SBC, Simpósios Brasileiros de Informática na Educação SBIE, World Conferences on Computers in Education WCCE, Computer-Supported Collaborative Learning CSCL, entre outras). Para isso muita pesquisa tem sido conduzida sobre o uso de software no ensino e avaliação dos alunos ([7], [3]), bem como no design e avaliação de sistemas computacionais em diversos contextos educacionais ([6],[1]). Como interfaces de qualidade são fundamentais para possibilitar a concretização do objetivo educacional desejado por meio do software, é necessário que se invista mais na pesquisa de IHC para o domínio educacional. Assim, teorias e práticas de IHC devem ser revistas e situadas para o design e avaliação de interfaces

de ambientes educacionais. Ao mesmo tempo, o design e avaliação de ambientes educacionais sob o ponto de vista de suas interfaces e possibilidades de interação é um conhecimento a ser re-significado no domínio da Informática e Educação.

A importância e atualidade do tema

Ambientes educacionais mediados pelas tecnologias abrem amplas possibilidades tanto como complementação ao ensino sendo feito em sala de aula, como permitindo a educação a distância, em contextos formais e não formais. Em um país como o Brasil, tais ambientes podem ter um papel ainda mais relevante, uma vez que têm o potencial de permitir que nossa sociedade supere alguns aspectos sócio-econômicos que representam desafios para a educação.

Para termos software educacional de qualidade é fundamental que as suas interfaces não apenas apresentem boa usabilidade e comunicabilidade, mas que favoreçam o aprendizado desejado. Desta forma, torna-se necessário que teorias e práticas de IHC sejam situadas no contexto de aplicações educacionais em nosso cenário social. Para isso é preciso que a comunidade de IHC amplie a discussão sobre este nicho da disciplina.

A relevância da oficina para a comunidade de IHC do Brasil

Na comunidade de IHC do Brasil várias pessoas têm pesquisado e investido em possibilitar o projeto e avaliação de interfaces de qualidade para ambientes educacionais, o que pode ser percebido pelo crescente número de publicações sobre o tema nos Simpósio Brasileiro de Fatores Humanos em Sistemas Computacionais (IHC's). Além disso, na área de Informática e Educação, muitas vezes pesquisadores e desenvolvedores se deparam com os desafios destas interfaces. Embora muitos destes trabalhos sejam apresentados no IHC ou em eventos de Informática e Educação, não foi ainda oferecido um fórum de discussão específico para os aspectos de design e avaliação de interfaces em ambientes educacionais e seus desafios. Esta oficina pretende fomentar uma discussão aprofundada dos trabalhos existentes na área e possibilitar a definição de uma agenda de pesquisa para a mesma.

Tópicos a serem abordados

Neste workshop busca-se identificar as peculiaridades que ambientes educacionais mediados pela tecnologia da computação possuem em termos de suas interfaces de usuário, possibilitar um mapeamento do que tem sido feito em termos metodológicos e o levantamento de linhas de investigação necessárias à área em questão. Assim, os tópicos de interesse incluem:

- Caracterização de interfaces de usuário em ambientes educacionais: requisitos e desafios
- Usos, especialização ou teorias que fundamentem a área
- Design Centrado no Aprendiz e outros paradigmas e métodos de design específicos para estas interfaces
- Métodos de avaliação específicos para estas interfaces

Dinâmica da Oficina

A oficina é proposta para um grupo limitado de participantes selecionados previamente pela sua contribuição à discussão pretendida. A partir de breve exposição dos participantes, temáticas específicas serão sugeridas para discussão em sub-grupos de trabalho. Uma discussão geral concluirá os resultados da oficina quanto ao estado da arte em interfaces para ambientes educacionais e próximos passos em relação a uma agenda de pesquisa. As organizadoras coordenarão os trabalhos e terão o papel de facilitadoras da discussão.

Sobre as organizadoras

Raquel O. Prates é professora adjunta do Departamento de Informática da Universidade do Estado do Rio de Janeiro, UERJ desde 1999. Ela se doutorou em Informática pelo Departamento de Informática da PUC-Rio, tendo se especializado na área de IHC. Raquel atua como pesquisadora colaboradora do Grupo de Pesquisa em Engenharia Semiótica na PUC-Rio. Sua pesquisa tem focado questões de Engenharia Semiótica, interfaces multi-usuários, avaliação de interfaces, interfaces para ambientes educacionais e comunidades virtuais.

M. Cecília C. Baranauskas é Professora Associada no Instituto de Computação e Pesquisadora no Núcleo de Informática Aplicada à Educação da Universidade Estadual de Campinas, Unicamp. Doutorou-se em Engenharia Elétrica, pela Unicamp. Seus interesses de pesquisa envolvem aspectos de IHC na análise, design e avaliação de interfaces de usuário em aplicações orientadas a diferentes domínios (sistemas colaborativos para aprendizado, sistemas de informação geográfica, sistemas para a prática de trabalho).

REFERÊNCIAS

- 1 BARANAUSKAS, M. C. C.; GOMES NETO, N. G.; BORGES, M. A. F. Learning at work through a multi-user synchronous simulation game. *International Journal Of Continuing Engineering Education And Life Long Learning*. Switzerland: v.11, n.3, p.251 - 260, 2001.
- 2 LEWIS, C., BRAND, C., CHERRY, G., RADER, C; (1998) *Adapting User Interface Design Methods to the Design of Educational Activities*. CHI 98, Los Angeles, pp 18-23.
- 3 PINTO, S. C. C. S. XIII Simpósio Brasileiro de Informática na Educação (SBIE 2002). Sociedade Brasileira de Computação, 2002.
- 4 PRATES, R. O.; de SOUZA, C. S.; BARBOSA, S. D. J.; "A Method for Evaluating the Communicability of User Interfaces." *Interactions* 7, 1. NY: ACM Press, 31-38, 2000.
- 5 PREECE, J.; ROGERS, Y.; SHARP
- 6 ROMANI, L.; H. (2002). *Interaction Design: Beyond Human-Computer Interaction*. New York, NY: John Wiley & Sons.
- 7 ROMANI, L.; ROCHA, H. (2001). *A Complexa Tarefa de Educar a Distância: Uma Reflexão sobre o Processo Educacional Baseado na Web*; *Revista Brasileira de Informática na Educação*, n. 8, abril 2001.

8 SAMPAIO, F. F.; MOTTA, C. L.R.; ELIA, M. F. XIV
Simpósio Brasileiro de Informática na Educação - Inclusão

digital como instrumento de inclusão social. Sociedade
Brasileira de Computação, 2003, v.1. p.858.

Artigos Aceitos para Participação

Ciclo de Elicitação de Requisitos de Learnware

Alex Sandro Gomes

Espaços Educacionais Virtuais Abertos e Flexíveis: Design com ou para o Usuário?

Amanda Meincke Melo, M. Cecília C. Baranauskas, Janaína Speglich de Amorim, Susie de Araújo Campos Alcoba

Uma abordagem Centrada no Usuário para o Apoio a Atividades Docentes em Ambientes de Educação a Distância

Apuena Vieira Gomes

Avaliação de Eficiência de Ícones no Software Educacional EDITOR MUSICAL

Maria Júlia S. Benini, Marcelo K. Zuffo, Leandro Leonardo Batista

Design de Interação, Estética do Efeito, e Informática em Educação: uma primeira aproximação

Lindsay Azambuja, Luiz Ernesto Merkle

Diretrizes para Interface de Software Educacional

Luciana Soares Fernandes, André L. A. Raabe, Fabiane B. V. Benitti, Luciane M. Schindwein

A Engenharia Semiótica para o Domínio Educacional

Raquel Oliveira Prates

Estética e usabilidade percebida de Ambientes Virtuais de Aprendizagem (AVAs)

Rosamelia Parizotto-Ribeiro

Avaliação de Interface no Software HagáQuê

Sílvia Amélia Bim

Modelagem de soluções ubíquas para uso em salas de aula do Ensino Fundamental

Taciana Pontual da Rocha Falcão, Alex Sandro Gomes

Processo para Avaliação de Interfaces de Softwares Educacionais

Teresa Cristina Motta Gurgel

Estratégias Cognitivas na Composição do Hiperdocumento com Material Instrucional para EAD

Vânia Paula de Almeida, Júnia Coutinho Anacleto Silva

Tutoriais

Análise e Modelagem de Tarefas

Marco Antonio Alba Winckler
LIHS-IRIT,
Université Paul Sabatier, Toulouse 3

Marcelo Soares Pimenta
Instituto de Informática
Universidade Federal do Rio Grande do Sul

RESUMO

O objetivo geral deste tutorial é apresentar aos participantes os conceitos fundamentais de Análise e Modelagem de Tarefas. Este tutorial de 4 hrs inicia com uma introdução sobre conceitos de tarefa, análise e modelagem de tarefas seguida de uma revisão de alguns modelos de tarefas mais conhecidos. Em particular pretende-se apresentar a notação CTT (Concurrent Task Tree) e dicas para uso do ambiente CTTE (CTT Environment), um dos ambientes mais difundidos e utilizados na atualidade para modelagem de tarefas. Exemplos extraídos de estudos de casos reais serão utilizados para demonstrar a importância e o poder de expressão de modelos de tarefas. Finalmente, serão discutidos os possíveis usos de modelos de tarefa em algumas etapas do ciclo de desenvolvimento de uma aplicação interativa.

Palavras-chave: Tarefas, Modelos de Tarefa, Análise de tarefa, Modelo CTT e Ambiente CTTE, Usos de Modelos de tarefa.

Objetivos

Hoje há um consenso entre os desenvolvedores de software de que a qualidade do desempenho do usuário no uso de um sistema interativo está ligada à usabilidade, um critério de qualidade associado ao componente conhecido como Interface com o Usuário (IU) deste sistema.

Sistemas computadorizados são projetados para auxiliar as pessoas a executarem tarefas. Logo, tarefas deveriam ser de interesse central para os desenvolvedores de software [...]. De fato, diversos autores da comunidade IHC (Interação Homem-Computador) convergem para a idéia central de que para projetar sistemas com maior usabilidade devemos compreender melhor as tarefas executadas pelas pessoas de modo a aplicar nosso entendimento das tarefas no desenvolvimento de aplicações [...].

A principal meta dos enfoques de projeto de IHC é aumentar a qualidade da interface com o usuário produzindo sistemas interativos não só funcionais e confiáveis mas também usáveis. Os enfoques de projeto baseados em modelos (model-based) permitem representar várias informações da interface e de seu

VI Simpósio sobre Fatores Humanos em Sistemas Computacionais — Mediando e Transformando o Cotidiano.

Hotel Bourbon, Curitiba, 17 a 20 de outubro de 2004. UFPR, CEIHC—SBC.

design em um alto nível de abstração. Isto permite várias vantagens como acoplamento a um enfoque metodológico de concepção, rastreabilidade e reuso de modelos, geração de (partes de) interfaces a partir destes modelos, e melhor reflexão sobre as decisões e exploração de alternativas do design. Entre os modelos comumente presentes nestes enfoques há modelos de usuário, diálogo, apresentação, domínio, contexto, plataforma tecnológica e tarefas para citar apenas alguns. Em particular, o uso de Análise de tarefas e Modelos de Tarefas visa um melhor entendimento de propriedades das tarefas realizadas pelos usuários em suas atividades e a aplicação deste entendimento no processo de construção da interface.

O objetivo geral deste tutorial é apresentar aos participantes os conceitos fundamentais de Análise e Modelagem de Tarefas. Em particular, pretende-se: i) apresentar fundamentos de Análise de Tarefas e Modelagem de Tarefas; ii) apresentar alguns modelos de tarefas existentes; pretende-se em particular demonstrar o uso da notação CTT (Concurrent Task Tree) e do ambiente CTTE (CTT Environment), um dos ambientes mais difundidos e utilizados na atualidade para modelagem de tarefas; e iii) discutir os possíveis usos de modelos de tarefa em algumas etapas do ciclo de desenvolvimento de uma aplicação interativa. Estudos de casos reais serão preferencialmente utilizados para exemplificar os conceitos apresentados.

Este tutorial está estruturado da seguinte forma. Após esta introdução, a seção 2 estabelece algumas definições de tarefa, análise de tarefa e modelos de tarefa. As seções 3 e 4 apresentam respectivamente uma introdução sobre análise de tarefa e modelagem de tarefas. A seção 5 contém uma descrição mais detalhada da notação CTT (Concurrent Task Tree) e do ambiente CTTE (CTT Environment). Na seção 6 discutem-se possíveis usos de modelos de tarefas no ciclo de desenvolvimento de sistemas interativos

Marco Antonio Alba Winckler é membro do laboratório de pesquisa *Logiciels Interactifs et Interaction Homme-Système* - LIHS (Softwares Interativos e Interação Ser Humano Sistema) no *Institut de Recherche en Informatique de Toulouse - IRIT* (Instituto de Pesquisa em Informática de Toulouse), na França. Tem especial interesse por métodos de design e avaliação de aplicações Web, tendo por tópicos: design Web; métodos de avaliação de usabilidade de aplicações web; ferramentas de suporte a avaliação de usabilidade; diretivas para interfaces Web; Interação Homem-Computador;

Hiperdocumentos e sistemas hipermídia.

Marcelo Soares Pimenta é Doutor em Informática, Universidade Toulouse 1, França (1997). Professor Adjunto do Departamento de Informática Aplicada (INA) do Instituto de Informática (II) da Universidade Federal do Rio Grande do Sul (UFRGS). Presidente da Comissão Organizadora do IHC2000 – III Simpósio Brasileiro de Integração de Fatores Humanos a Sistemas Computacionais,

Presidente da Comissão de Programa do IHC2001 - IV Simpósio Brasileiro de Integração de Fatores Humanos a Sistemas Computacionais, Presidente da Comissão Especial da SBC para IHC (2000/2001). Tem por áreas de Interesse: Interação Humano-Computador (IHC), Engenharia de Requisitos, Engenharia de Software, Computação e Música, Fatores Humanos e Sociais no Desenvolvimento de Software.

Padrões de Interação - o contexto WEB

Americo Talarico Neto

Júnia Coutinho Anacleto Silva

André Constantino da Silva

Rosângela A. Dellosso Penteado

Departamento de Ciência da Computação
Universidade Federal de São Carlos

OBJETIVO

O objetivo desse tutorial é fornecer um “primeiro contato” com o tema padrões para se ter uma visão do que são, o que não são, onde encontrá-los, como identificar uma aplicação, como utilizá-los, entre outras informações. Pretende-se, para esse primeiro contato, apresentar aos participantes um conjunto de padrões para projetos WEB, motivando-os a identificar e utilizar padrões em seus projetos.

A principal característica de um padrão, inicialmente propostos por Alexander para o domínio da arquitetura com o objetivo de aumentar a participação do usuário nos seus projetos, é que ele expressa conhecimento através de uma boa solução para um problema recorrente em um determinado contexto. Esse conhecimento é resultado da captura da prática e organizado em um formato bem definido com certo grau de abstração, permitindo uma fácil compreensão por parte de seus usuários.

Nesse contexto, pesquisadores de IHC estão propondo padrões que podem ser utilizados para o reuso do conhecimento, útil principalmente para profissionais com pouca experiência, e também para um maior envolvimento do usuário, aumentando a comunicação do usuário com o especialista através do fornecimento de um vocabulário comum. Em especial, padrões de interação estão sendo propostos e podem ser utilizados em projetos WEB.

Dada a importância do envolvimento do usuário em projetos e a crescente demanda dos projetos WEB, a IHC tem-se preocupado com a identificação e aplicação de padrões. A comunidade brasileira de IHC, atualizada em seus temas de pesquisa, também está atenta à questão, embora ainda existam poucos trabalhos sobre padrões realizados por tal comunidade.

O sucesso da abordagem de padrões no ambiente computacional impulsionou o interesse na utilização de padrões para o apoio ao projeto da interação entre o

homem e o computador. O objetivo inicial dos padrões, proposto na Arquitetura, e adotado pela IHC é que padrões apoiem a participação do usuário no projeto do ambiente com o qual ele interage.

Nesse contexto, há a necessidade de estimular as pesquisas para a identificação, organização e divulgação de Padrões no domínio da IHC, de forma que permita ao desenvolvedor utilizar soluções comprovadas para problemas recorrentes nesse domínio e que seja acessível a todos os participantes em um projeto interdisciplinar, como é o caso do projeto WEB.

Americo Talarico Neto, com graduação em Ciência da Computação, pela Universidade Federal de São Carlos, é mestre em Ciência da Computação pela mesma instituição.

André Constantino da Silva, Graduado em Ciência da Computação pela Universidade Federal de Goiás, é Mestre em Ciências da Computação. Universidade Federal de São Carlos.

Rosângela A. Dellosso Penteado é doutora em Física Computacional pela IFSC-USP, e professora do Departamento de Computação da Universidade Federal de São Carlos. tem por áreas de Engenharia de Software e a Especificação e Desenvolvimento de Sistemas Reativos.

Júnia Coutinho Anacleto Silva, professora do Departamento de Computação da Universidade Federal de São Carlos, é Doutora em Física Aplicada Computacional pela USP. Tem por áreas de pesquisa a Interação Humano-Computador, a Educação a Distância, a Engenharia de Software, e a Realidade Virtual.

Cognitive, social and emotive aspects of computer-mediated human activity

Annette Aboulafia

Computer Science & Information Systems,
University of Limerick
Limerick, Ireland

ABSTRACT

The tutorial is first of all intended to support a discussion of theories and methods in this areas of research, as well as to provide a more in debt point of view where Psychology, and especially Activity Theory, as a generic psychological approach, may support an understanding of computer-mediated activity, mediation, and 'transformation'.

keywords:

Activity, mediation, 'transformation'/development

OBJECTIVE

In the introduction for 'Call for Contributions' for this conference, there are a few concepts I find of particular interest such as the notions of 'human activity', 'mediation' ('everyday life mediates through technology') and 'transformation' (transformation of everyday life). Outside the field of Psychology, in particular, these concepts or phenomena are used in many different contexts as well as having many different meanings. This is often a problem for ongoing research – and especially in multidisciplinary areas such as e.g. human computer interaction.

The meaning of these concepts or phenomena have been explored especially within the discipline of Psychology, and to a great extent within the approach called Activity Theory and/or Cultural Historical School. We would even say that the notions of Activity, Mediation and Transformation - or what we would call development - are the very key phenomena within Psychology, and as such its objects of research. Referring to General Psychology for clarity of the basic understanding of these phenomena thus seems sensible, and is one of the aims of this tutorial.

Next to specifying the basic understanding of these concepts, an important objective of HCI is "the evaluation of human activities mediated by computing systems" – also mentioned in the introduction. Evaluating processes such as e.g. cognitive or social activity – with or without specific technological mediation – is never easy as psychological processes are always in a constant development – they are never static. This poses the methodological question of how to evaluate such 'none

stable' processes? This was also a question posed by Vygotsky in the 1920th, and which he sought to resolve arguing for a developmental and historical method. Within the broad area of 'human-computer interaction', however, it is rare to find such methods being applied, and this suggests the question of the adequacy of the methods being applied in this field of research. The method of research has to be adequate with its research object. As such a closer look on 'task and user analysis', methods trying to capture the notion of 'user needs', or broadly speaking 'human activities mediated by computing systems' seems sensible and worthwhile. The tutorial will address the question of method being one of the most important problems for understanding human forms of psychological activity.

The diversity of disciplines applied in the area of human-computer interaction (in broad) makes this area very complex to deal with. The variety of disciplines draw on (meta)theories and methods within all 3 branches of science: the natural science, the social science and the Humanities. This is reflected in the three areas of research named HCI, CSCW (Computer Supported Collaborative Work) and the new trend often named 'affective design'.

Trying to understand the demand of 'new' theoretical approaches this tutorial will start with identifying the emergence of the HCI field of practice, and its need for theoretical framework. Focusing merely on cognitive issues (based on the Information processing approach) HCI is heavily criticized from a Social Science point of view. Arising from this critique a new trend emerges, the so-called CSCW (Computer Supported Cooperative Work), where it is argued that social processes have an effect on the usage of IT and thus also needs another meta-theoretical point of view. Yet another, more recent approach has emerged, the 'affective approach' focusing on the emotive or motivational aspects of computer-mediated activity.

Although adding yet another 'missing position' in the area of research do provide a broader view of the field of research it also raises methodological questions. As neither of the approaches in themselves provides a sufficient understanding of computer-mediated activity (which is also the reason for the emergence of other approaches), it is however not simply a matter of combining these aspects or perspectives. We need a framework that is able to conceptualize all three aspects, their relationship and their dynamics from the very start of the analysis, i.e. as fundamental principles of human activity. In this context Activity Theory, based on Dialectic-Materialistic Philosophy, is very useful and

VI Simpósio sobre Fatores Humanos em Sistemas Computacionais — Mediando e Transformando o Cotidiano. Hotel Bourbon, Curitiba, 17 a 20 de outubro de 2004. UFPR, CEIHC—SBC.

beneficial. As a generic psychological framework Activity Theory has identified some basic principles from which more specific questions within the area of computer-mediated activity (work, learning and play) can be based. It is argued that without basic or generic principles it is not possible to understand specific and concrete phenomena in everyday life. Drawing on this framework it is possible to characterize the three approaches as the cognitive, the social and the emotive aspect of computer-mediated activity. Arguing that all three aspects of human activity are necessary if we are to understand how humans behave, think and feel in the 'real world', such a framework is necessary if we are to investigate research questions such as e.g. the psychological consequences of computerization, develop models for design, evaluation criteria, etc. Next to presenting the above ideas the tutorial will present a framework for working life which incorporate the four most basic aspects or motives ('user needs'): the economic aspect, the cognitive aspect, the social aspect, the meaning aspect.

The four-hour tutorial is divided into 3 sessions. Each session ends with a discussion of the presented ideas (about 15 min.):

Session I

- Who am I? (education, areas of research)
- General view of applied psychological knowledge in the area of computer mediated activities (in the areas of HCI, CSCW, and 'affective design')
- Discussion of problems and prospect

Session II

- Methodological problems solved by Vygotsky (the psyche-physical problem in research, the area of validity of Activity Theory)

- Basic principles of Activity Theory (especially the concepts of Activity, Mediation and Transformation/Development)
- Examples of applying the basic principles to computer-mediated activity (e.g. the concept of 'user needs' in a framework for working life incorporating the cognitive, social and emotive/meaning aspects of working life).

Session III

- Research methods of human activity mediated though technology (e.g. task analysis, experimental methods, Ethnographic studies, historical/developmental methods)
- Multidisciplinary research (problems and prospects)
- General discussion and evaluation of workshop

Annette is a University of Limerick Foundation research fellow. She is involved in the Shared Worlds project and has previously worked as research officer on the Enterprise Ireland MMS project. She is Cand.Psych. from Copenhagen University, Psychological Laboratory. Her main research areas has been in the areas of General Psychology, Cognitive Psychology, Work and Organisational Psychology, and Philosophy of Science. She has previously worked on three EU ESPRIT projects - all concentrated around different aspects of new technology and working life issues. Her Ph.D. - also from Copenhagen University - deals with some generic aspects of HCI and CSCW, focusing on the cognitive and the social aspects of human work activity.

Computação Ubíqua: Conceitos, Exemplos e Desafios

Maria da Graca Pimentel

ICMC - USP - São Carlos

OBJETIVOS

O tutorial tem como objetivos: (a) introduzir o conceito de computação ubíqua; (b) apresentar o estado da arte em termos de pesquisa nessa área, com foco em aspectos da interação do usuário com aplicações e dispositivos a ela associados; (c) demonstrar aplicações que executam em Tablet PCs e PDAs; (d) discutir estudos de caso relativos a aplicações desenvolvidas pelo grupo bem como as lições aprendidas.

Descrição clara e detalhada do tema

A evolução da computação e o surgimento de novas tecnologias têm alterado a dinâmica da vida cotidiana das pessoas e apresentando-lhes novos valores. Não é a tecnologia em si que realmente importa, mas sim a maneira com que as pessoas fazem uso dela. Mark Weiser vislumbrou no início da década de 1990 uma forma pouco intrusiva de fazer uso do computador no cotidiano das pessoas. Segundo ele, facilidades computacionais devem ser incorporadas ao ambiente a fim de auxiliar atividades humanas mudando minimamente a forma como tais atividades são realizadas. Essa forma transparente de integrar tecnologia às atividades diárias das pessoas foi denominada computação ubíqua. As visões de Weiser previam a proliferação de dispositivos computacionais ubíquos de diversos tamanhos: pequenos e pessoais (inch-scale), de médio porte e pessoais (foot-scale), e grandes e de uso coletivo (yardscale). De fato, isso acabou acontecendo: dispositivos pessoais de pequeno e médio porte tais como PDAs (Personal Digital Assistants), laptops tornaram-se comuns no final da década de 90, e Tablet PCs estão ficando cada vez mais comuns. Da mesma forma, dispositivos maiores, como lousas eletrônicas, passam a fazer parte de ambientes de uso comum tais como salas de reuniões, salas de aula e laboratórios. Hoje faz-se uso de dispositivos que extrapolam as escalas inicialmente previstas por Weiser, chamados wall-sized. Para explorar o uso desses novos dispositivos surgiram novas aplicações computacionais, cujo desenvolvimento está diretamente associado a três temas, que concentram o foco de pesquisa da área de computação ubíqua. Segundo Gregory D. Abowd, e Elizabeth D. Mynatt, estes são interfaces naturais, captura e acesso de atividades humanas e computação ciente de contexto.

O estado da arte correspondente a essas áreas será apresentado durante o curso. Várias aplicações serão

demonstradas com o uso de PDAs e Tablet PCs e, se possível, lousa eletrônica — algumas dessas aplicações foram construídas pelo grupo. As lições aprendidas com a construção das aplicações serão apresentadas e novas oportunidades de pesquisa e de desenvolvimento serão discutidas.

Relevância do tema

São vários os problemas a serem atacados no desenvolvimento de aplicações associadas à computação ubíqua, problemas esses que envolvem várias das áreas de computação tais como sistemas distribuídos, sistemas de informação e interação usuário-computador.

Além disso, a tendência é que vários dispositivos portáteis do tipo PDAs and Tablet PCs fiquem cada vez acessíveis. Assim, se faz necessária a compreensão, por parte do profissional em computação, da área de computação ubíqua de modo geral, da relevância de aplicações que executam em dispositivos móveis em particular, bem como de problemas associados à construção de aplicações que fazem uso de tais tecnologias.

Relevância da tutorial para a comunidade de IHC no Brasil

A área de computação ubíqua de modo geral é um tema relativamente novo, principalmente em termos de esforços de pesquisa e desenvolvimento da área de IHC no Brasil. Assim, se faz necessária que o tema seja apresentado à comunidade e que oportunidades de pesquisa e de desenvolvimento sejam discutidas.

Cronograma Preliminar de Atividades, com estimativas de tempo

- apresentação, histórico e definições (1h);
- estado da arte e discussão de problemas (2h);
- apresentação e demonstração de tecnologias (2h);
- apresentação de estudos de caso e lições aprendidas (2h);
- discussão de oportunidades de pesquisa e consideração finais (1h)

Tópicos e sub-tópicos: apresentação, histórico e definições (1h); estado da arte e discussão de problemas (2h); interfaces naturais; aplicações de captura e acesso; computação ciente de contexto; calm technology; wearable computing; apresentação e demonstração de tecnologias (2h); aplicações que exploram lousa eletrônica; aplicações que exploram Tablet PCs; aplicações que exploram PDAs; aplicações que exploram sensores; apresentação de estudos de caso e lições aprendidas (2h); iClass: uso de lousa eletrônica em aplicações de captura e acesso; wiClass: uso de PDAs com o iClass; labClass: uso de TabletPCs com o iClass; Operações alternativas de anotações em PDAs; Lições

VI Simpósio sobre Fatores Humanos em Sistemas Computacionais — Mediando e Transformando o Cotidiano. Hotel Bourbon, Curitiba, 17 a 20 de outubro de 2004. UFPR, CEIHC—SBC.

aprendidas; discussão de oportunidades de pesquisa e consideração finais (1h).

Maria da Graça Pimentel é Professora Livre-docente do ICMC-USP, onde investiga os problemas associados à construção de aplicações de computação ubíqua que, gerando documentos como resultado da captura de experiências ao vivo, permitem que esses

documentos sejam manipulados de modo aberto e extensível. Foco é dado à estruturação das aplicações de modo a permitir o armazenamento, recuperação, extensão e visualização dos hiperdocumentos gerados, e, mais recentemente, à utilização de dispositivos móveis. Ministra o curso de interação usuário-computador na USP/São Carlos desde 1996 na graduação e na pós-graduação.

ISBN 857669031-4



9 798576 690312