

Tutor hiper m dia baseado no modelo de autoria NCM para o Interpretador Gr fico de Estrutura de Dados

Gilberto F. Sousa Filho¹, Dorgival Pereira da S. Netto¹, Lettiery D'Lamare P. Procopio¹, Andrei A. Formiga², Alisson V. Brito²

¹Departamento de Ci ncias Exatas – Universidade Federal da Para ba (UFPB)

²Centro de Inform tica – Universidade Federal da Para ba (UFPB).

{gilberto, dorgival.netto, lettiery.dlamare}@dce.ufpb.br, {andrei, alisson}@ci.ufpb.br

Abstract. *IGED is a pedagogical tool created to support the teaching of data structures and algorithms. Its guiding principles are based on a constructive approach that includes interaction and graphical visualization of the structures. The previous version of the tool allowed the creation of constructive activities, but did not support the creation of presentations or other types of didactic activities. In this paper it is proposed a new architecture for IGED that solves this limitation by adding hypermedia-authoring capabilities, based on the NCM model. This addition makes the IGED tool more capable in supporting teaching.*

Resumo. *O IGED   uma ferramenta pedag gica criada para auxiliar no ensino das estruturas de dados e seus algoritmos, baseado em uma abordagem construtiva que inclui intera o e visualiza o gr fica das estruturas. A vers o anterior do IGED permitia apenas a defini o de atividades construtivas, n o tendo suporte   cria o de apresenta es ou atividades mistas. Neste artigo   proposta uma nova arquitetura para o IGED que inclui a capacidade de criar e definir sequ ncias did ticas hiper dia seguindo o modelo NCM. Essa adi o torna o IGED mais flex vel e capaz na defini o de atividades did ticas pelo professor.*

1. Introdu o

A dificuldade no ensino das Estruturas de Dados e seus algoritmos   amplamente conhecida entre professores e alunos dos cursos de computa o. Uma dificuldade comumente citada pelos alunos   a capacidade de abstra o necess ria para entender o funcionamento das estruturas de dados, e a possibilidade de visualizar (e interagir com) as estruturas esquematicamente   julgada de grande ajuda, como visto em Netto et al. (2011). O IGED – Interpretador Gr fico para Estruturas de Dados –   uma ferramenta de programa o criada para o ensino das Estruturas de Dados em cursos de computa o, baseada nas ideias de visualiza o utilizando hiper dia e na abordagem construtiva para objetos de aprendizagem, como relatado em Netto et al. (2011). Este artigo aborda a extens o da arquitetura do IGED para incluir um ambiente de cria o de objetos hiper dia, permitindo aos professores a ado o de apresenta o de slides aninhados a anima es e a atividades no contexto da disciplina.

H  muito tempo se recomenda que o processo de ensino-aprendizagem inclua o uso de recursos tecnol gicos com o objetivo de facilitar a aprendizagem e torn -la mais atrativa. Com nios, pai da did tica moderna, mostra em seus tratados a necessidade da

aplicação de recursos áudios-visuais no ensino, visto que a aprendizagem se dá pelos sentidos e pela experiência.

Segundo Scaveta e Lauffer (1997), **hipermídia** é a reunião de várias mídias num suporte computacional, sendo o ambiente hipermídia, desde sua estruturação até o acesso interativo de informações, um modelo de representação aberta que apresenta interfaces com o usuário e cujo volume de informações aborda, paulatinamente, novas referências sígnicas, havendo uma leitura sinestésica que mobiliza os sentidos quando estas informações são dispostas em rede.

Os sistemas hipermídia têm atraído a atenção de educadores há décadas. Marchionini (1988) aponta duas características da hipermídia que são importantes para a educação: (i) o armazenamento de grande quantidade de informações representadas sob os mais diversos meios, permitindo que conteúdos extensos e variados sejam agrupados e disponibilizados aos estudantes; (ii) o alto nível de controle do sistema pelo usuário, permitindo a navegação pelo documento através de suas escolhas.

Na estruturação dos materiais didáticos, um importante conceito é o de sequência didática. Machado (2000) define uma sequência didática como a unidade de trabalho escolar, constituída por um conjunto de atividades que apresentam um número limitado e preciso de objetivos. A sequência didática estrutura as atividades particulares em uma atividade englobante, de tal forma que essas atividades tenham um sentido para os aprendizes. As sequências podem ser produzidas por um autor e aplicadas por vários professores a diversos alunos que se enquadrem em seu público alvo.

Para que um objeto hipermídia tenha uma função efetiva no ensino, o planejamento de uma sequência didática pelo professor deve preceder o seu desenvolvimento. Este trabalho propõe a adoção de um modelo de autoria hipermídia chamado NCM (*Nested Context Model*), criado por Soares e Rodrigues (2005), para a ferramenta IGED, permitindo assim que o professor-autor possa criar objetos de ensino que reflitam a sequência didática previamente elaborada e execute este objeto com os alunos, tudo em um mesmo ambiente computacional.

O trabalho está dividido em seis partes. Após esta introdução, na Seção 2 são apresentados os trabalhos relacionados na literatura; na Seção 3 é proposta uma nova arquitetura para o IGED, enquanto na Seção 4 é apresentado o tocador de aplicações do ambiente IGED. A Seção 5 traz a descrição do tutor hipermídia e um protótipo da ferramenta e, por fim, a Seção 6 traz as considerações finais sobre o trabalho.

2. Trabalhos Relacionados

Diversos projetos de ferramentas de ensino, em várias áreas, adotaram o conceito de aplicações hipermídia para abordar seus conteúdos programáticos. Nesta seção são apresentados alguns projetos relacionados ao IGED.

O sistema hipermídia de aprendizagem *Biomec* [Cola 2004] aborda aspectos biomecânicos e anatômicos do movimento humano com apoio de conceitos de mecânica. A idéia é que a integração desses conteúdos leve o aluno a entender a biomecânica do movimento humano e suas relações com a mecânica e a anatomia humana. Foram criados conteúdos com textos, vídeos e animações do corpo humano.

O trabalho de Ávila (2004) tem como propósito desenvolver um *software* sobre anatomia em radiografias panorâmicas, organizadas em um programa tutorial multimídia interativo. O sistema trabalha com o sequenciamento de imagens, uso de jogos e propõe atividades para avaliar a assimilação dos conhecimentos passados. Ferramentas como o *Biomec* e o *software* de Ávila (2004) tornam-se inflexíveis quanto a seus conteúdos, pois o usuário não pode compor novos materiais e a aula fica amarrada ao plano inicial de sua implementação.

O Moodle é um software livre e nele é possível produzir conteúdos web hipermídia. Segundo Bastos et al. (2008), o Moodle potencializa tanto a interação quanto a interatividade na comunicação mediada por computador. Algumas de suas funcionalidades são: inserir imagens às páginas, criar páginas de texto, links para arquivos ou páginas, chats e fóruns de discussão.

A ferramenta Scratch disponibiliza um ambiente gráfico de programação que permite aos autores criar animações, jogos e simulações. Maloney et al. (2004) apresenta as características básicas da ferramenta: construção de objetos através de blocos de vários tipos e tamanhos de polígonos; programação de mídias como imagens, vídeos e sons aninhadas aos eventos do usuário; objetos criados podem ser aproveitados na elaboração de objetos mais complexos.

O modelo de aninhamento de diferentes mídias adotado pelas ferramentas Moodle e Scratch é limitado, não havendo tratamento de eventos, amarrando o autor dos conteúdos a eventos exclusivos de usuários. Outro aspecto dessas ferramentas é a generalidade, não sendo ferramentas pensadas especificamente para as necessidades da criação de Objetos de Aprendizagem voltados para a disciplina de estrutura de dados. Em contraste, o IGED foi criado inicialmente com este função [Netto et al. 2011].

Segundo Schwabe (1998), o OOHDM é um modelo de desenvolvimento de aplicações hipermídia que define quatro processos: projeto conceitual, que define um esquema conceitual de representação de objetos, suas relações e colaborações no domínio alvo; projeto de navegação, responsável pela definição de diferentes visões sobre o esquema conceitual associado aos perfis de usuário; projeto de abstração de interface, descreve a interface de usuário associando cada objeto da aplicação a seu estado e sua interface de representação e por fim a implementação, que deve ser guiada pelos projetos para a criação da aplicação em seu domínio. O OOHDM, diferente do modelo NCM adotado neste trabalho, não descreve as entidades que representam as mídias, seus relacionamentos e eventos que os ativam, sendo o autor da aplicação responsável pela formalização de cada processo durante sua implementação.

Apesar de possibilitarem o uso de objetos hipermídia, os trabalhos citados não são estruturados sobre uma formalização de conteúdos hipermídia como o NCM. Isso os torna menos adaptativos a novos tipos de mídia e, principalmente, menos flexíveis. O NCM, por sua vez, tem potencial de permitir que o professor elabore sua sequência didática de forma simples e extremamente poderosa.

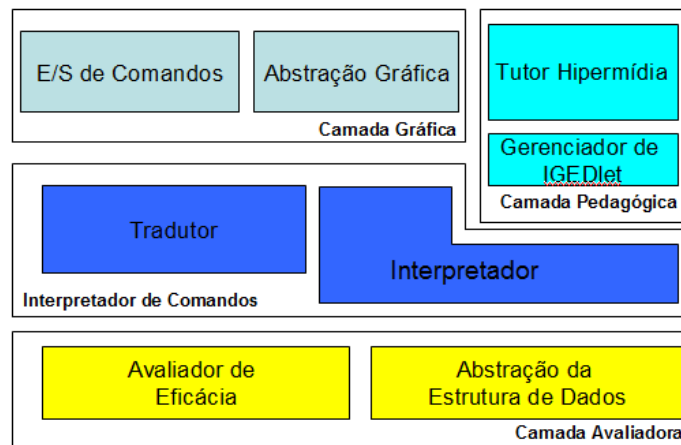


Figura 1. Nova arquitetura do ambiente IGED.

3. Nova arquitetura do IGED

Netto et al. (2011) apresentaram uma arquitetura para o Interpretador Gráfico de Estruturas de Dados (IGED). Neste trabalho é proposta uma modificação na arquitetura anterior, com a finalidade de adicionar novas funcionalidades pedagógicas ao IGED.

Esta nova arquitetura, ilustrada na Figura 1, apresenta quatro camadas distintas: Camada Gráfica, responsável pela interação com o usuário; o Interpretador de Comandos, que avalia e executa os comandos na linguagem IGED passados pelo usuário; Camada Avaliadora, responsável por comparar o resultado obtido pelos comandos passados pelo usuário com o resultado esperado em cada atividade; por fim, a Camada Pedagógica inclui os componentes Gerenciador de IGEDlets, responsável pela composição e execução de aplicações do ambiente IGED, e o componente Tutor Hipermídia, responsável pela composição e execução de sequências didáticas para o ensino de estruturas de dados. Estes dois componentes serão detalhados nas seções seguintes.

Adaptações do Interpretador de Comandos

A camada responsável pela abordagem construtiva do IGED é chamada interpretador de comandos, na qual se encontram dois componentes. O primeiro é o componente Tradutor, responsável pela tradução dos programas em linguagem de alto nível (mais simples para o aluno) para uma linguagem intermediária; isso permite que o segundo componente, chamado Interpretador, execute o código intermediário para efetuar a manipulação das estruturas de dados.

Na nova arquitetura do IGED, foi necessário adaptar o componente interpretador de comandos para dar suporte às aplicações IGEDlet, que serão descritas a seguir. O interpretador de comandos deve trabalhar com 4 modos distintos:

- Modo gráfico, onde as atualizações interpretadas são passadas apenas para a camada abstração gráfica, ficando a camada avaliadora desabilitada;
- Modo professor, onde a camada de abstração fica desabilitada e a camada avaliadora atualiza apenas as estruturas do professor;
- Modo aluno, onde as duas camadas são atualizadas, entretanto na camada avaliadora apenas as estruturas do aluno são atualizadas;
- Modo inicialização, onde as duas camadas são habilitadas e tanto as estruturas do professor quanto do aluno são atualizadas.

4. Aplicações IGEDlet

Um IGEDlet é uma pequena aplicação que realiza uma tarefa específica, com escopo no ensino de estruturas de dados, sendo executado no ambiente IGED. São definidos dois tipos de IGEDlets: a AnimaçãoIGEDlet é uma aplicação com finalidade de ilustrar animações sobre manipulações em estruturas de dados e a AtividadeIGEDlet permite a elaboração de exercícios para aprendizes também no escopo do ensino de estruturas de dados.

A Figura 2 ilustra a modelagem do IGEDlet, além dos dois tipos de aplicação, apresenta sua composição com um objeto Metadado que traz suas informações básicas como: autor, título, descrição, id e tipo, além de suas ações descritas a seguir.

Ações de uma aplicação IGEDlet:

- *execute()* : inicializa a execução da aplicação IGEDlet, lançando eventos para indicar sua inicialização e seu termino.

- *pause()* : pára temporariamente a execução do IGEDlet, guardando a posição dentro do código da aplicação que está sendo executado. Lança evento indicando que a aplicação foi pausada.
- *continue()*: retoma a execução da aplicação a partir do ponto em seu código armazenado durante a execução do *pause*.
- *interrompe()* : finaliza a execução da aplicação, lança evento para indicar sua interrupção.
- *reinicie()* : interrompe a aplicação e a inicializa novamente.

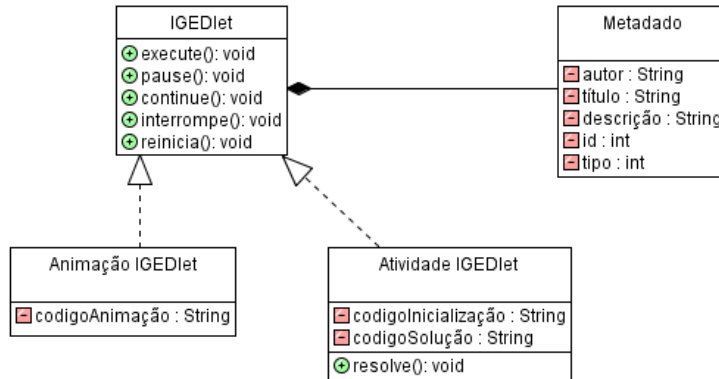


Figura 2. Modelagem das Aplicações IGEDlet.

AnimaçãoIGEDlet

Uma aplicação *AnimaçãoIGEDlet* permite ao professor elaborar animações que representam manipulações sobre estruturas de dados, criando assim objetos de aprendizagem que adotam a abordagem passiva para a apresentação dos conteúdos.

Uma *AnimaçãoIGEDlet* é composta por um código descrito na linguagem interpretada pelo ambiente IGED para a composição da animação proposta.

AtividadeIGEDlet

Uma aplicação *AtividadeIGEDlet* permite ao professor elaborar atividades sobre manipulação de estruturas de dados, onde o aprendiz pode, através da linguagem IGED, responder a questão levantada e sua resposta será avaliada pela aplicação. Com a *AtividadeIGEDlet* o professor pode criar objetos de aprendizagem com a abordagem construtiva, exigindo do aluno a construção de conhecimento guiado pelo professor.

Uma *AtividadeIGEDlet* é composta pela questão a ser resolvida, um código que descreve a inicialização das estruturas onde o aluno deve atuar para responder a questão levantada e um código com a solução do professor para o problema, todos estes códigos são escritos usando a linguagem IGED.

Além das ações herdadas de um *IGEDlet*, a *AtividadeIGEDlet* possui a ação *resolve*, que recebe como parâmetro a resposta do aluno, e além de animar graficamente a solução proposta, envia um evento indicando a avaliação da solução fornecida.

4.1 Tocador de IGEDlet

O componente Tocador de *IGEDlet* é responsável pela execução de toda aplicação no ambiente do IGED, seu primeiro passo é identificar o tipo de aplicação para tratá-la adequadamente, como descrito a seguir.

Tocando uma AnimaçãoIGEDlet

O *Tocador de IGEDlet*, ao receber uma *AnimaçãoIGEDlet*, configura o componente *Interpretador* no modo gráfico, ativando apenas o componente *Abstração Gráfica*. O código de animação é fornecido passo a passo em intervalos de tempos pré-determinados ao componente *Interpretador* para que a animação seja observada pelo aprendiz com detalhes. Caso seja pedido para reiniciar a animação, o *Tocador de IGEDlet* solicita ao *Interpretador* que limpe suas atualizações e o processo de animação é iniciado novamente.

Tocando uma AtividadeIGEDlet

Ao receber uma *AtividadeIGEDlet*, inicialmente o componente *Interpretador* será configurado no modo inicialização, que atualiza as camadas *Gráfica* e *Avaliadora*, e seu código de inicialização é fornecido ao *Interpretador*. Após esta fase, o *Interpretador* será configurado no modo professor e o código de solução fornecido pelo autor da atividade será passado para o *Interpretador*. Como neste modo a camada gráfica está desabilitada o aprendiz não visualizará a resposta proposta, apenas a cópia do professor na camada avaliadora será atualizada, em seguida, o *Interpretador* é posto em modo aluno e o código fornecido pelo usuário é passada para o *Interpretador*, que atualiza a camada gráfica, fornecendo um feedback ao aluno, e as estruturas do aluno são atualizadas na camada avaliadora. Por fim, a camada avaliadora é consultada sobre a avaliação da resposta do aluno, neste ponto, as estruturas do professor e do aluno são comparadas e, se estiverem idênticas será indicado que a atividade foi concluída com sucesso, caso contrário, será indicado erro em sua solução.

Caso seja pedido para reiniciar a atividade, o *Tocador de IGEDlet* solicita ao *Interpretador* que limpe suas atualizações e o processo de execução de uma *AtividadeIGEDlet* é iniciado novamente.

Compositor de Aplicação IGEDlet

O compositor de aplicações IGEDlet é responsável por receber os metadados que descrevem a aplicação e os localizadores dos código de inicialização e solução descritos na linguagem IGED, que ficam armazenados em arquivo texto, gerando um XML que descreva a aplicação, permitindo sua persistência e posterior execução pelo *Tocador de IGEDlets*.

5. Tutor Hipermídia para o IGED

O tutor hipermídia é proposto como um componente do IGED que permita a criação de seqüências didáticas para o ensino da disciplina Estruturas de Dados, possibilitando a criação de apresentações de conteúdo aninhadas a aplicações IGEDlet (animações ou atividades), apresentando ao aprendiz os conceitos da disciplina segundo uma ordem definida pelo autor do objeto.

Com o objetivo de definir seqüência lógica de apresentação dos conteúdos, propomos a adoção de um modelo de autoria hipermídia, o NCM, para a implementação do componente Tutor Eletrônico.

5.1 Mapeamento do NCM no Tutor Eletrônico

O NCM (*Nested Context Model*, ou Modelo de Contextos Aninhados) apresentado em Soares e Rodrigues (2005) é um modelo de autoria de documentos hipermídia, composto de grafos com nós e elos, permitindo segmentar e estruturar o documento, conforme necessário, através de nós de composição também chamados de contexto.

Para adoção do NCM na elaboração de uma sequência didática no componente Tutor Eletrônico, foram mapeados os elementos principais do modelo às entidades básicas do ambiente. Segue o mapeamento proposto:

Nó de mídia

Um nó de mídia caracteriza o objeto de conteúdo propriamente dito, seja ele um vídeo, imagem, um texto, um áudio, uma aplicação etc. O IGED trabalha com dois tipos de nós de mídia, imagens para apresentação dos conteúdos (slides) e aplicações IGEDlet para animações de ações sobre estruturas de dados ou atividades sobre os temas abordados.

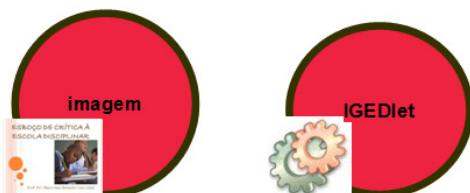


Figura 3. Nós de mídia do IGED

Todo nó de mídia terá associado a ele entidades âncoras, que tem o objetivo de especificar o relacionamento entre nós. No NCM cada nó pode ter mais de uma âncora, geralmente utilizado em mídias contínuas para referenciar partes distintas do conteúdo, nos dois tipos de nós de mídia proposto o uso de mais de uma âncora permite ao professor atribuir ao mesmo slide, várias atividades ou animações simultâneas.

Nó de Composição

Um nó de composição C é um nó cujo conteúdo é uma coleção C_L de nós que se constituem em suas unidades de informação. Os componentes de C podem ser ordenados, facilitando operações de navegação, além de possuir uma lista ordenada de âncoras que permitem acesso de nós externos a nós internos a composição.

O NCM define um tipo especial de nó de composição, a Trilha, nela seus nós de conteúdo podem ser iterados com operações de acesso do tipo próximo e anterior. No ambiente do Tutor é proposto o uso de Trilha para agrupar nós de mídia do tipo imagem, compondo uma apresentação de slides.

Para definir entradas em um nó de composição, o NCM define uma entidade do tipo interface, chamada porta, possibilitando assim associar o início de uma composição a um de seus nós de mídia.

Contexto

Os nós de contexto, tipo especial de nó de composição, servem para estruturar o documento hipermídia, definindo uma estrutura lógica hierárquica ou não. Desta forma, contextos podem ser aninhados, procurando sempre refletir a estrutura do documento e tornar sua organização mais intuitiva.

Propomos um tipo de nó de contexto, o Contexto de Sequência Didática (CSD), que agrupa apresentações e aplicações IGEDlet (animações ou atividades), delimitando objetos relacionados com um mesmo conteúdo, através de uma sequência lógica planejada pelo professor, construindo assim um Objeto de Aprendizagem que executa em sala de aula, ou em um ambiente de EaD a sequência didática pensada pelo professor autor. O CSD pode agrupar outros contextos permitindo a definição de hierarquias de conteúdo.

O NCM define que o contexto possui como atributo adicional um conjunto de elos, onde cada elo l contido no conjunto define um relacionamento entre os nós do

contexto. Através do uso de elos o professor define diferentes visões de um mesmo documento e facilita a orientação do aprendiz na navegação.

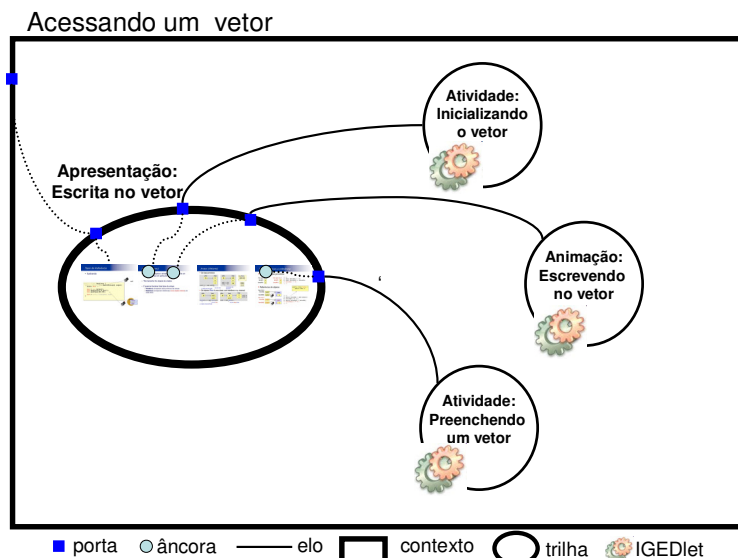


Figura 4. Contexto de Sequência Didática (CSD) “Acessando um vetor”.

A Figura 4 ilustra as entidades que modelam o Contexto de Sequência Didática intitulado “Acessando um vetor”, composto de um nó Trilha de apresentação e três aplicações IGEDlets. A Trilha, que representa uma apresentação de slides, é formada por uma coleção de nós de mídia do tipo imagem, algumas destas imagens são relacionadas com outros nós de mídia IGEDlets como demonstram os elos que ligam suas âncoras a cada mídia.

Os elos podem estabelecer relacionamento entre nós de contexto permitindo assim a criação de sequências didáticas mais completas, e atribuindo pré-requisitos entre os conteúdos abordados.

5.2 Linguagem do Tutor Hiperídia

Para que o componente Tutor Hiperídia trabalhe com a criação de sequências didáticas e sua posterior execução, adotamos uma linguagem que descreve esta sequência, podendo persisti-la e, em outro momento, ser traduzida pelo próprio Tutor Hiperídia, executando o Objeto de Aprendizagem programado.

Neste componente propomos a adaptação da linguagem NCL (Nested Context Language), uma linguagem de autoria de documentos hiperídia de propósito geral, para uma abordagem de ensino. A linguagem NCL, apresentada em Soares e Rodrigues (2005), é uma linguagem declarativa que descreve o que deve ser feito em um objeto hiperídia. A linguagem utiliza os conceitos do modelo NCM, nós e elos, para descrever a sequência lógica dos elementos do documento.

A Figura 5 ilustra um exemplo de código NCL adaptado para o Tutor Hiperídia, descrevendo o Contexto de Sequência Didática “Acessando um Vetor”, logo, podemos observar o mapeamento dos elementos deste contexto, ilustrados na Figura 4, com as tags definidas na linguagem e seu aninhamento lógico.

Assim, o Tutor Hiperídia passa a ser um compositor e tradutor de sequência didáticas programadas através de uma linguagem declarativa baseada na linguagem NCL.


```

- <body>
  <port id="pInicio" component="ctxVetor" interface="pInicio" />
  <-context id="ctxVetor">
    <port id="pInicio" component="vetor" interface="pVetor" />
    <-trail id="vetor">
      <port id="pVetor" component="slide1" />
      <media type="imagem" id="slide1" src="vetor/press/slide1.jpg" />
      <-media type="imagem" id="slide2" src="vetor/press/slide2.jpg" />
      <area id="aAtividade1" x="100" y="200" />
      <area id="aAtividade2" x="100" y="230" />
      </media>
      <media type="imagem" id="slide3" src="vetor/press/slide3.jpg" />
      <-media type="imagem" id="slide4" src="vetor/press/slide4.jpg" />
      <area id="aAnimacao1" x="100" y="200" />
      </media>
    </trail>
    <media type="igedlet" id="atividade1" src="vetor/iged/vetorAtiv1.igedlet" />
    <media type="igedlet" id="animacao1" src="vetor/iged/vetorAnima.igedlet" />
    <media type="igedlet" id="atividade2" src="vetor/iged/vetorAtivi2.igedlet" />
    <-link id="ISlide1IGEDlet1Start1" xconnector="onSelection1Start1">
      <bind component="slide2" interface="aAtividade1" role="onSelection" />
      <bind component="atividade1" role="start" />
    </link>
    <-link id="IIGEDlet1Slide1Resume1" xconnector="onStop1Resume1">
      <bind component="slide2" role="resume" />
      <bind component="atividade1" role="OnStop" />
    </link>
    <-link id="ISlide1IGEDlet1Start2" xconnector="onSelection1Start1">
      <bind component="slide4" interface="aAnimacao1" role="onSelection" />
      <bind component="animacao1" role="start" />
    </link>
    <-link id="IIGEDlet1Slide1Resume2" xconnector="onStop1Resume1">
      <bind component="slide4" role="resume" />
      <bind component="animacao1" role="OnStop" />
    </link>
  </context>
</body>

```

Portas

Âncoras

Trilha
"Escrita no Vetor"

IGEDlets

Elos

Contexto
"Acessando um Vetor"

Figura 5. Linguagem do Tutor para a sequência didática “Acessando um Vetor”.

5.3. Protótipo

Na Figura 6 é mostrado um protótipo do Tutor Hipermídia para o IGED. Nesta figura, está representada a transição entre a exibição de uma apresentação de slides e a execução de uma AtividadeIGEDlet, através da ativação de um elo entre as mídias, preestabelecido na sequência didática do professor, devido ao evento de seleção da âncora contida na imagem da apresentação.

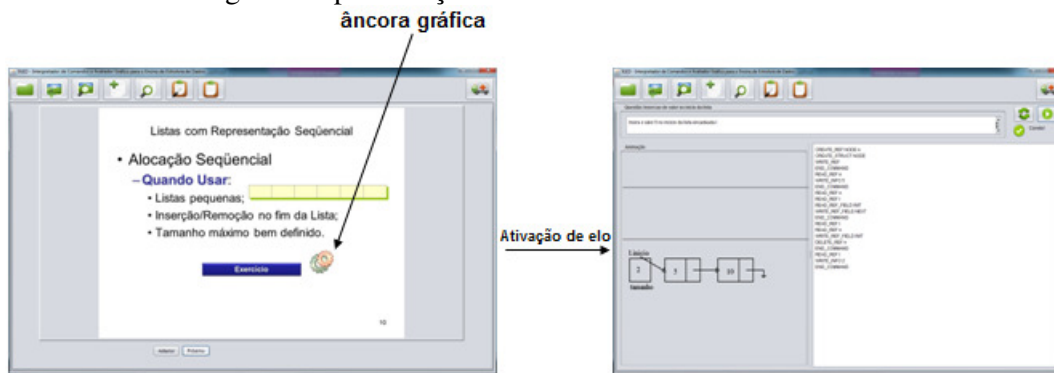


Figura 6. Interface do ambiente durante a execução de uma sequência didática.

6. Considerações Finais

Neste trabalho foi proposta uma nova arquitetura com funcionalidades pedagógicas adicionais para o ambiente IGED, ferramenta criada para auxiliar no ensino de estruturas de dados através de uma abordagem construtiva, baseada em interação e visualização. A nova arquitetura inclui um novo componente de composição e execução de sequências didáticas propostas pelo professor-autor para mediar o ensino de

conceitos da disciplina de estrutura de dados. Este componente, chamado Tutor Hiperímia, é baseado no uso do modelo de autoria de documentos hiperímia NCM. O uso do padrão NCM possibilita realizar o aninhamento das mídias através de uma sequência lógica pré-definida.

Diferente de outros ambientes hiperímia similares, o IGED dá suporte à criação de aplicações IGEDlets contendo animações e atividades voltadas para a disciplina de estrutura de dados. Essas aplicações são implementadas através de uma linguagem de alto nível. O ambiente ainda permite que as atividades propostas sejam avaliadas, indicando ao aprendiz a eficácia de sua resposta.

Como trabalhos futuros, pretende-se incluir suporte para novas mídias ao Tutor Hiperímia como texto, vídeo e outros tipos de aplicações como applets Java, animações Flash e outros padrões similares.

Referencias Bibliográficas

- Ávila, Maria A. G. (2004) “Software anatomia em radiografias panorâmicas: avaliação do método de ensino-aprendizado em Odontologia”. *f* 175. Tese Faculdade de Odontologia da USP. São Paulo.
- Bastos, F. da P. de et al. (2008) “Exemplares de interação mediada por computador na internet: atividades de estudo e colaboração utilizando a ferramenta wiki no Avea Moodle.” In: _____. Interação mediada por computador. Santa Maria: Ed. da UFSM; UAB, p. 25-45.
- Cola, C. S. D. (2004) “Biomec: um sistema hiperímia que integra conceitos de mecânica, biomecânica e anatomia humana.” (Dissertação de mestrado). NUTES-UFRJ.
- Machado, Anna Rachel.(2000) Uma experiência de assessoria docente e de elaboração de material didático para o ensino de produção de textos na universidade. DELTA [online], vol.16, n.1, pp. 1-26. ISSN 0102-4450. <http://dx.doi.org/10.1590/S0102-44502000000100001>.
- Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B., and Resnick, M. (2004). “Scratch: A Sneak Preview. Second International Conference on Creating, Connecting, and Collaborating through Computing”. Kyoto, Japan, pp. 104-109.
- Marchionini, G. (1988) “Hypermedia and learning: freedom and chaos”, Educational Technology, 28(11), 8-12, November.
- Netto, D. P.; Oliveira, T. J. ; Sousa, T. D. N.; Sousa Filho, G. F.; Formiga, A.; Brito, A. V. (2011) “Desenvolvimento de um Interpretador de Comandos e Avaliador Gráfico para o Ensino de Estrutura de Dados (IGED)”. In: XIX Workshop sobre Educação em Computação no XXI Congresso da Sociedade Brasileira de Computação. Natal.
- Scavetta, S. and Lauffer, R. (1997) “Texte, Hipertexte, Hipermedia”. Paris, Intro.
- Schwabe, D. and Rossi G. (1998). “An Object Oriented Approach to Web-Based Application Design”. Wiley and Sons, 207-225, Nova Jersey.
- Soares, Luiz Fernando Gomes, RODRIGUES, Rogério Ferreira. (2005) “Nested Context Model”. Monografia em Ciência da Computação da PUC-RJ. Rio de Janeiro.