

Abordagens Orientadas a Modelos no desenvolvimento de software em Saúde: Contribuições e Perspectivas

Rafael X. de O. Souza¹, Adicinéia A. de Olivera¹

¹Grupo de Pesquisa em Engenharia de Software (GPES)
Universidade Federal de Sergipe (UFS) – São Cristóvão, SE – Brasil

rafaelxavier73@gmail.com, adicineia@ufs.br

Abstract. *The health area has received investments around the world to the development of information systems that provide support for its activities. However, to reflect the needs of users and be more productive, there are still some challenges to be faced in the process of development for these systems. In this context, model-driven approaches are potentially relevant to deal adequately with these problems because they offer a higher level of abstraction in software development. This feature provides reusability, portability and interoperability, improving process productivity and quality of the applications. In this paper it will be discussed the application of model-driven approaches in the development of Health Information Systems (HIS).*

Resumo. *A área da saúde tem recebido investimentos em todo mundo para o desenvolvimento de sistemas de informação que proporcionem apoio às suas atividades. No entanto, existem ainda alguns desafios a serem enfrentados para que o processo de desenvolvimento destes sistemas tenha maior produtividade e reflita às necessidades dos usuários. Nesse contexto, as abordagens orientadas a modelos apresentam um potencial relevante para lidar de forma adequada com esses problemas pelo fato de oferecerem um nível mais alto de abstração no desenvolvimento de software. Tal característica proporciona reusabilidade, portabilidade e interoperabilidade, aumentando a produtividade do processo e a qualidade das aplicações. Neste artigo é apresentada uma análise da aplicação de abordagens orientadas a modelos no desenvolvimento de Sistemas de Informação em Saúde (HIS).*

1. Introdução

As empresas cada vez mais aumentam as exigências por softwares de qualidade que reflitam com exatidão os seus processos organizacionais. Diante desse cenário, a indústria de desenvolvimento de software tem trabalhado com o objetivo de criar novas tecnologias que atendam a tais necessidades.

Nesse sentido, abordagens orientadas a modelos, tais como Desenvolvimento Orientado a Modelos (*Model-Driven Development* – MDD), Arquitetura Orientada a Modelos (*Model Driven Architecture* – MDA) e Modelagem Específica de Domínio (*Domain-Specific Modeling* – DSM), apresentam um potencial relevante para contribuir na melhoria da qualidade do processo e do produto de software, pelo fato de permitirem a elevação do nível de abstração. Além disso, elas permitem a especificação do sistema de maneira a tornar possível a geração do código da aplicação de forma automática através de operações de transformações entre os modelos criados sob diferentes pontos de vista.

Segundo Bergmann (2010), em todo mundo tem sido feitos grandes investimentos para o desenvolvimento de Sistemas de Informação em Saúde (*Health Information Systems* – HIS), com o objetivo de prover um ganho de produtividade na área da saúde, através da captação dos benefícios proporcionados pelas novas tecnologias utilizadas atualmente no desenvolvimento de software. Assim, atualmente tem sido dada uma considerável atenção ao potencial das abordagens orientadas a modelos no suporte ao desenvolvimento de HIS, conforme pode ser observado em Jones (2006),

Khambati *et al.* (2008), Raghupathi e Umar (2008), Balagtas-Fernandez e Hussmann (2009), Menezes *et al.* (2010), Ding e Klein (2010) e Gomes *et al.* (2011).

Raghupathi e Umar (2008) afirmam ainda que o uso de tais abordagens na área da saúde viabiliza o desenvolvimento de HIS: (i) com baixo custo e *time-to-market*; (ii) interoperáveis; (iii) independentes de plataforma; e, (iv) melhor manuteníveis. Diante disso, diferentes pesquisas têm sido realizadas a fim de propor aplicações de abordagens orientadas a modelos, como também visando analisar os resultados da utilização dessas tecnologias na área da saúde.

O objetivo deste artigo é analisar as contribuições e as limitações das abordagens orientadas a modelos no desenvolvimento de software para a área da saúde. Este artigo contribui com uma análise que permite observar o cenário atual do uso de abordagens orientadas a modelos para o desenvolvimento de software no domínio da saúde, bem como as perspectivas futuras para as pesquisas nessa área.

Este artigo está organizado da seguinte forma: na seção 2 são apresentadas as principais abordagens orientadas a modelos usados no desenvolvimento de software. Na seção 3 são abordados os principais problemas do desenvolvimento de software para saúde. Na seção 4 é realizada uma análise do uso das abordagens orientadas a modelos para a resolução de tais problemas. Na seção 5 são apresentados alguns trabalhos relacionados. Por fim, na seção 6 são feitas as considerações finais.

2. Abordagens Orientadas a Modelos

Desde a antiguidade as disciplinas de engenharia têm a atividade de modelagem como uma técnica fundamental para lidar com a complexidade. A modelagem fornece uma maneira de facilitar a compreensão, o raciocínio, a construção, a simulação e a comunicação sobre sistemas complexos [THOMAS 2004].

Hoje em dia, existem iniciativas que buscam o aumento da abstração no processo de desenvolvimento de software, as quais se baseiam na mudança dos esforços de desenvolvimento, de código e programação para modelos e modelagem. Nesse sentido, novos conceitos e abordagens surgem como base para a disseminação desse novo paradigma de desenvolvimento de software. Dentre eles destacam-se os conceitos de MDD, MDA e DSM.

2.1. MDD

O MDD representa um avanço para o desenvolvimento de software, pois apresenta uma nova abordagem, baseada em modelos, para este processo. Mellor, Clark e Futagami (2003) definem MDD como a noção de que se pode construir um modelo de um sistema que poderá ser transformado em algo real.

Segundo Fernandes, Machado e Carvalho (2007), MDD constitui uma abordagem de desenvolvimento de software que se concentra fortemente e se baseia em modelos, através dos quais podem ser construídos modelos de softwares independentes de plataforma. Isto corresponde a um aumento na abstração de linguagens de programação de alto nível para linguagens de modelagem. Este aumento de abstração implica que um sistema poderá ser principalmente (ou totalmente) expresso por meio de modelos.

A abordagem MDD permite uma abstração dos detalhes da tecnologia da plataforma de implementação e utilização de conceitos mais próximos do domínio do problema. Consequentemente, os modelos utilizados para desenvolver o sistema são mais fáceis de especificar, entender e manter, e são resistentes às alterações da tecnologia adotada para a implementação do sistema [SELIC 2003]. Assim, o MDD promove a idéia de que através do foco nos modelos pode-se obter melhor desenvolvimento e evolução do software.

Diante disso, Fernandes (2010) enumera as principais contribuições do uso do MDD:

- Ganho de produtividade: Atkinson e Kuhne (2003) afirmam que a melhoria da produtividade dos esforços de desenvolvimento é a "motivação fundamental do MDD". Eles consideram que a produtividade seja atingida ao longo de duas dimensões. Primeiro, a produtividade de curto prazo, obtida através do quanto de funcionalidade um artefato de software pode oferecer. Segundo, a produtividade de longo prazo, obtida pelo aumento da longevidade do artefato de software.
- Conceitos mais próximos do domínio: Selic (2003) observa que o desenvolvimento de software baseado em modelos utiliza conceitos que são menos ligados à tecnologia de implementação e mais próximos do domínio do problema, o que permite, eventualmente, que não seja preciso ser um especialista em computação para "produzir sistemas". Além disso, o foco na modelagem permite a redução ou eliminação de erros e lacunas semânticas na passagem de um modelo abstrato de para um produto final de implementação, visto que "modelo é o sistema".
- Automação e menor sensibilidade às mudanças tecnológicas: Mellor, Clark e Futagami (2003) destacam como benefícios do MDD a transformação automática dos modelos de *design* de alto nível em sistemas em execução e a facilidade uso e manutenção dos modelos, os quais também são menos sensíveis à tecnologia escolhida.
- Captura de conhecimentos especializados e reutilização: MDD permite a captura de conhecimento especializado através de funções de mapeamento que transmitem informações para a transformação de um modelo para outro. Isto permite a sua reutilização quando um aplicativo ou a sua implementação muda, como também permite uma evolução independente dos modelos e leva a modelos com maior longevidade.

2.2. MDA

MDA é um *framework* de arquitetura aberto e independente de fornecedor para a construção de sistemas de software, proposto pelo OMG (*Object Management Group*), em 2001. É uma tecnologia que permite a gestão de mudanças, relações e transformações entre as definições de processos de negócios a um desenvolvimento tecnológico que suporte tal definição.

O Guia do MDA [OMG 2003] apresenta os conceitos básicos necessários para o entendimento da arquitetura, a saber:

- Modelo Independente de Computação (*Computational Independent Model – CIM*): A abordagem baseada em modelos estabelece que os requisitos do sistema sejam modelados em um CIM, também conhecido como Modelo de Domínio ou Modelo de Negócios, que independe de como o sistema será implementado.
- Modelo Independente de Plataforma (*Platform Independent Model – PIM*): Baseado no CIM elaborado anteriormente, o PIM descreve o sistema sem, no entanto, refletir todas as decisões ou detalhes relativos a questões de plataforma.
- Modelo de Plataforma (*Platform Model – PM*): Após o desenvolvimento do PIM, é escolhida uma plataforma para a sua implementação, a qual possui um modelo inerente, o "modelo de plataforma", que o arquiteto utiliza para especificar os mapeamentos do PIM para o modelo de plataforma de destino.
- Modelo Específico de Plataforma (*Platform Specific Model – PSM*): O PSM reflete o PIM do sistema, enriquecido com os conceitos, serviços e detalhes da plataforma onde o sistema será implementado. Um PIM pode gerar vários PSM, para cada plataforma diferente em que o sistema for ser utilizado.

- Mapeamentos: A transformação de um PIM para um PSM é feita através de uma especificação fornecida por um mapeamento. Esta especificação é composta por regras e/ou algoritmos que determinam como a transformação deve ser processada.

Em essência, a MDA propõe para o ciclo de vida de desenvolvimento de sistemas, o desenvolvimento de um PIM do sistema que, livre de problemas específicos da plataforma tecnológica, detalha a estrutura e o comportamento do mesmo. Dada uma plataforma tecnológica escolhida, este PIM pode ser transformado em um ou vários PSM, que incorporam todos os detalhes tecnológicos necessários inerentes à plataforma em que o sistema será implementado. A partir deste PSM, o código do sistema pode ser gerado automaticamente para a plataforma de destino.

Através da separação das preocupações de arquitetura, envolvendo o uso de PIM, transformações de modelos, PSM e código gerado automaticamente, a MDA pretende oferecer portabilidade, reusabilidade e interoperabilidade no desenvolvimento de software.

2.3. DSM

A DSM, que utiliza uma metalinguagem, com seu metamodelo, para representar cada domínio. Conforme destacam Kelly e Tolvanen (2004), a DSM tem como objetivos: (i) elevar o nível de abstração além da programação, especificando a solução em uma linguagem que usa diretamente conceitos e regras de um domínio de problema específico; e (ii) gerar produtos finais numa linguagem de programação escolhida.

Normalmente, a geração de código é ainda apoiada por um *framework*, que fornece as implementações comuns para as aplicações dentro do domínio. A automação do desenvolvimento das aplicações é possível porque a linguagem de modelagem, o gerador de código e o *framework* devem atender às exigências de um domínio de aplicação restrito. Em outras palavras, eles são de específicos de domínio e estão totalmente sob o controle de seus usuários.

Na DSM, é definida uma metalinguagem para cada domínio, diferindo de outras metalinguagens mais genéricas, como a *Unified Modeling Language* (UML) que visa representar o paradigma Orientado a Objetos. Assim, os modelos são mais específicos e completos, e recursos como *frameworks*, padrões de projeto e componentes são incluídos na modelagem com o objetivo de gerar maior quantidade de código e com melhor qualidade [GREENFIELD e SHORT 2004].

3. Desenvolvimento de Software na Área da Saúde

O uso de tecnologia está cada dia mais imerso no cotidiano das pessoas e os recursos tecnológicos são aplicados nas mais diversas áreas. Dentre elas, a área da saúde destaca-se por necessitar de grande agilidade e eficiência nas atividades desenvolvidas, as quais requerem resposta precisas e, muitas vezes, em tempo real.

Bergmann (2010) destaca que a tendência global em cuidados de saúde está voltada para modelos de atenção mais centrados no paciente, baseada na manutenção do bem-estar, ao invés dos modelos tradicionais centrados no médico, baseados nas doenças. Assim, o fornecimento de sistemas de saúde eficazes, apoiados por sistemas de software de alta qualidade é bastante importante para a melhoria dos serviços prestados pelos profissionais da área.

No entanto, o desenvolvimento de software voltado para a área de saúde tem encontrado problemas relacionados a duas questões principais: (i) o software geralmente não é adequadamente projetado para atender às necessidades de seus usuários, apoiando de forma efetiva os fluxos de trabalho existentes; e (ii) o processo de desenvolvimento de software não garante a entrega atendendo o cronograma, orçamento e as metas de funcionalidades de maneira adequada.

Além disso, existem ainda alguns desafios a serem enfrentados no desenvolvimento de HIS, tais como:

- Complexidade dos próprios processos de cuidados em saúde.
- Falta de técnicas e métodos adequados para proporcionar maior produtividade e qualidade no processo de desenvolvimento de HIS.
- Necessidade de integração e interoperabilidade entre as diferentes aplicações pertencentes ao domínio.
- Necessidade de portabilidade entre diferentes plataformas e dispositivos, visto que os serviços em saúde geralmente devem ser oferecidos em tempo real e acessados de maneira rápida, em diferentes situações.

4. Abordagens Orientadas a Modelos na Área da Saúde

As abordagens orientadas a modelos apresentam um potencial relevante na busca pela solução dos problemas do desenvolvimento de software para a área da saúde, pois permitem que desenvolvedores projetem a aplicação de acordo com um domínio específico, independente de plataforma, onde as transformações geram requisitos específicos para plataforma e aplicações distintas dentro desse domínio. Desta forma, o MDD tem sido bastante utilizado no processo de produção de software na saúde.

Nas seções 4.1 a 4.4 são discutidas as formas como tais abordagens lidam com os desafios enumerados na seção 3 e, na seção 4.5, é realizada uma análise sobre o potencial do uso das mesmas para a melhoria do desenvolvimento de software para a área da saúde.

4.1. Complexidade

Os desenvolvedores de HIS têm que lidar com a natureza altamente desconectada e complexa dos processos de saúde que eles estão tentando modelar. Observa-se hoje uma proliferação de aplicações com diferentes padrões e plataformas, refletindo tal complexidade.

Como resultado disto, além das próprias dificuldades para o desenvolvimento, os softwares produzidos tornam-se pouco reusáveis, apresentando baixa capacidade de manutenção, o que, conseqüentemente, aumenta os custos inerentes ao processo.

Teixeira *et al.* (2009) e Gomes *et al.* (2011) observam que a modelagem de dados clínicos geralmente sofre constantes mudanças devido à complexidade existente na transcrição do conhecimento do profissional médico para dentro do sistema. Tal fato pode ainda acarretar problemas de manutenibilidade e interoperabilidade a médio e longo prazo, exigindo um esforço muito maior do que o devido por parte da equipe de desenvolvimento.

Nesse contexto, o conjunto de especificações proposto pela Fundação OpenEHR [BEALE 2002], a qual visa o desenvolvimento de padrões abertos para a construção de HIS, fornece meios para uma separação entre a modelagem de dados clínicos e a modelagem da informação. Assim, os analistas de sistemas seriam responsáveis por definir um modelo de referência, o qual receberia restrições definidas pelos chamados arquétipos de dados clínicos produzidos pelos profissionais de saúde envolvidos.

Com um processo de desenvolvimento baseado em modelos, o MDD estabelece que a maior complexidade do sistema seja tratada nas fases iniciais do desenvolvimento, onde o foco está na abstração do domínio do problema, fazendo com que as fases subseqüentes, principalmente a manutenção, tenham seu esforço minimizado.

Através do uso da MDA, a complexidade fica restrita ao PIM, o que permite que o desenvolvimento de um PSM seja menos custoso e trabalhoso. Além disso, as alterações no PIM são automaticamente transmitidas para os PSM através de transformações, de maneira que não é necessário que haja um esforço para modificar cada um deles individualmente.

Já o uso de DSM permite a construção de um metamodelo específico para o domínio da saúde. Isso permite que as abstrações sejam mais precisas, pois o escopo fica mais bem definido.

Desta forma, as abordagens orientadas a modelos tornam a separação entre modelos de dados clínicos e modelos de informação mais natural e eficiente, reduzindo assim a complexidade inerente ao processo de desenvolvimento de HIS.

4.2. Produtividade e Qualidade

Através da separação entre o modelo conceitual (PIM) e o modelo físico (PSM) com o uso da MDA, o desenvolvedor não precisa reconstruir modelos conceituais a cada mudança de plataforma. O uso de ferramentas automatizadas para gerar PSM por meio de transformações do PIM também pode aumentar a produtividade do desenvolvimento. Além disso, os desenvolvedores podem mudar o foco do código para o PIM, portanto, prestando mais atenção à resolução do problema em questão.

O processo MDD proporciona maior produtividade no desenvolvimento de HIS pelo fato de mudar o foco para os modelos, que oferecem um nível mais alto de abstração. Desta forma, há uma menor sensibilidade às mudanças, além de maior qualidade no produto de software produzido. O ganho de produtividade também pode ser obtido graças à maior capacidade de automação das atividades, com a geração automática de código e o uso massivo de ferramentas de apoio.

Com o uso de DSM, torna-se possível a automação do desenvolvimento das aplicações, pois a linguagem de modelagem, o gerador de código e o *framework* que apóia a geração do código atendem às exigências de um domínio de aplicação restrito.

4.3. Integração e Interoperabilidade

PSM geradas a partir de um PIM podem ter relacionamentos, os quais são chamados de pontes. Quando os PSM são direcionados a diferentes plataformas, eles não podem falar uns com os outros diretamente. Há uma necessidade de transformação de conceitos de uma plataforma em conceitos usados em outra plataforma.

Como no MDD os conceitos relacionados ao sistema estão mais próximos do domínio. Com isso, os detalhes de plataforma podem ser abstraídos, tornando mais fácil a integração e a comunicação entre as aplicações desenvolvidas dentro do domínio em questão. Já na MDA, o problema da interoperabilidade é tratado quando não apenas os PSM são gerados, mas também as pontes necessárias entre eles. A interoperabilidade entre plataformas pode ser implementada com ferramentas que geram tanto PSM, quanto as pontes entre eles e, em alguns casos pontes para outras plataformas também.

O uso do padrão de arquétipos na construção de HIS permite compartilhar conceitos e conhecimento clínico, formando sistemas a partir da união de diversos deles. Assim, com a utilização de um padrão de representação formal destes arquétipos, é possível que as especificações de conceitos clínicos criados a partir de tal formalismo resultem em HIS fundamentados em conceitos sólidos e interoperáveis [NARDON, FRANÇA e NAVES 2011].

Na DSM a solução é especificada em uma Linguagem Específica de Domínio (*Domain-Specific Language* – DSL), que usa diretamente conceitos e regras de um domínio de problema específico. Assim, uma DSL pode ser usada neste contexto para a representação formal de arquétipos.

4.4. Portabilidade

No desenvolvimento de HIS, é preciso lidar com as necessidades distintas que envolvem as aplicações. Como as tarefas exigem um maior grau de agilidade, os softwares podem ter que operar em múltiplas plataformas e dispositivos, usando diferentes tecnologias, para oferecer suporte a visões distintas em tempo real.

No MDD, o sistema evolui a partir de um modelo mais conceitual, independente da tecnologia envolvida, ou seja, está focado apenas no problema. Assim, devido à abstração dos detalhes computacionais e tecnológicos, o processo de desenvolvimento facilita a portabilidade das aplicações entre plataformas distintas.

Devido à especificação de um metamodelo e de uma metalinguagem dentro de um domínio específico, no caso o da saúde, a abordagem DSM favorece uma maior abstração do problema, sendo que os detalhes de computacionais são especificados posteriormente. Com o foco na solução do problema, o qual está bem definido dentro de um determinado domínio, a DSM permite que o desenvolvedor não se preocupe tanto com a tecnologia ou a plataforma a ser utilizada, podendo haver inclusive diferentes implementações para cada plataforma.

Dentro da MDA, a portabilidade é alcançada pelo foco no desenvolvimento de modelos PIM, que são, por definição, independentes de plataforma. Um PIM pode ser transformado automaticamente em múltiplos PSM para diferentes plataformas. Em princípio, tudo especificado no nível PIM é, portanto, totalmente portátil. O nível de portabilidade depende das ferramentas de transformação automatizadas que estão disponíveis.

4.5. Análise

Pelo fato de focarem num nível mais alto de abstração, lidando com questões relativas ao domínio do problema, o uso da orientação a modelos no desenvolvimento de HIS favorece uma concepção mais bem definida das funcionalidades do software, deixando às questões computacionais para um segundo momento, quando o escopo da solução já está especificado de forma satisfatória.

Entretanto, existem desafios que apenas o uso de abordagens orientadas a modelos não consegue permitir um alto nível de interoperabilidade, manutenibilidade e reuso no desenvolvimento de HIS, como:

- A dificuldade em lidar com as constantes mudanças na modelagem de dados clínicos, devido à complexidade de transcrever o conhecimento do profissional médico para dentro do sistema, a qual pode se traduzir em problemas de manutenibilidade e interoperabilidade a médio e longo prazo.
- Apesar do fato de a abordagem MDA ser capaz de facilitar o reuso de uma boa arquitetura, ela não auxilia o projeto de novas boas arquiteturas. Tipicamente, sistemas dentro de uma mesma família de HIS empregam estilos arquiteturais semelhantes. Esses estilos devem poder ser mais bem explorados pelas equipes de desenvolvimento visando projetos arquiteturais que propiciem manutenibilidade e reuso de longo prazo.

Assim, a utilização da orientação a modelos ainda depende de alguns fatores. Além dos descritos, como a existência de ferramentas adequadas para a automatização das atividades e a melhoria da qualidade dos processos de transformação, sobretudo aquelas de nível “horizontal” (modelos com metamodelos diferentes), além do processo de “ida e volta” (transformação em ambos os sentidos).

Porém, tais problemas têm sido amenizados com o andamento das pesquisas e o surgimento de ferramentas com mais funcionalidades, além de métodos e técnicas de transformação mais eficientes. Desta forma, mesmo que ainda diante de desafios, a própria evolução das abordagens

poderá proporcionar benefícios para os diferentes domínios onde estas são usadas para desenvolver aplicações, particularmente a área da saúde.

A especificação de padrões de interoperabilidade, como é feito pela Fundação OpenEHR, aliada aos conceitos propostos pelas abordagens orientadas a modelos, também contribui para a melhoria do processo de desenvolvimento de HIS através da reutilização de conhecimento clínico representado de forma portátil e independente de plataforma.

A Tabela 1 apresenta um resumo da análise, detalhando as diferentes formas como as abordagens orientadas a modelos podem lidar com os desafios do desenvolvimento de HIS.

Tabela 1. Soluções das abordagens orientadas a modelos para os desafios do desenvolvimento de HIS.

Desafio	MDA	MDD	DSM
Complexidade	Complexidade restrita ao PIM.	Abstração do domínio do problema nas fases iniciais.	Construção de um metamodelo específico para o domínio.
Produtividade e Qualidade	Separação entre o modelo conceitual (PIM) e o modelo físico (PSM).	Mais alto nível de abstração, menor sensibilidade a mudanças e maior automatização dos processos por meio de ferramentas.	Automação dos processos pelo atendimento de exigências de um domínio de aplicação restrito.
Integração e Interoperabilidade	Geração de pontes entre os PSM.	Conceitos mais próximos do domínio do problema.	Solução especificada em uma linguagem que usa diretamente conceitos e regras de um domínio de problema específico.
Portabilidade	Transformação de PIM para PSM.	Evolução do processo a partir de um modelo conceitual independente de plataforma e tecnologia.	Foco na solução do problema dentro de um domínio específico.

5. Trabalhos Relacionados

Gomes *et al.* (2011) propõem um método MDD para o desenvolvimento de HIS, visando proporcionar um alto grau de manutenibilidade e reuso empregando transversalidade na concepção de modelos de dados clínicos e modelos arquiteturais.

Menezes *et al.* (2010) propõem uma abordagem que utiliza o HL7, um padrão internacional para troca de mensagens entre HIS heterogêneos. Esta abordagem tem como objetivo a integração de arquétipos em mensagens HL7, usando uma DSL, o que permite modelar as mensagens HL7 com arquétipos e automatizar a maior parte do processo de construção de HIS pervasivos.

Balagtas-Fernandez e Hussmann (2009) propõem um *framework* conceitual e um processo que permitiria a simplificação do desenvolvimento de aplicações móveis para vigilância sanitária através da aplicação de MDD, especificamente a abordagem DSM.

Raghupathi e Umar (2008) apresentam um exemplo de aplicação da abordagem MDA em HIS. Além disso, eles destacam como potenciais benefícios dessa adoção: (i) uso de abstração para lidar com a complexidade; (ii) redução do *backlog* de aplicações; (iii) possibilidade de uso de padrões; (iv) diminuição da dependência de fornecedor; (v) possibilidade de portabilidade; e (vi) menores custos de desenvolvimento.

Essas pesquisas demonstram maneiras de aplicar as abordagens orientadas a modelos na área da saúde, descrevendo como resultados os avanços em relação aos métodos tradicionais de desenvolvimento de software.

No entanto, diferentemente deste trabalho, os trabalhos correlatos não discutem a aplicação de tais abordagens na área da saúde sob um ponto de vista crítico, discutindo questões como limitações, formas de aplicação, além dos benefícios e perspectivas.

6. Considerações Finais

Este trabalho realiza uma análise das abordagens orientadas a modelos usadas para o desenvolvimento de aplicações no domínio da saúde.

A partir da análise apresentada, é possível observar que, apesar da computação ubíqua ser um conceito novo e ainda pouco conhecido, já existe um relevante esforço no sentido de viabilizar o desenvolvimento de aplicações adequadas a sua realidade.

Com base na análise apresentada, pode-se concluir que as abordagens orientadas a modelos oferecem soluções para lidar com as dificuldades existentes no processo de desenvolvimento de HIS. Apesar de ainda haver a necessidade de lidar com alguns fatores, estas abordagens permitem que tal processo seja realizado de forma mais eficiente, já permitindo um apoio às suas atividades de maneira adequada.

Como trabalho futuro é possível sugerir um estudo mais amplo da aplicação de MDD, MDA e/ou DSM no desenvolvimento de HIS, como por exemplo, na produção de um conjunto de aplicações, a fim de que se possa avaliar a seus benefícios e identificar limitações e problemas a serem resolvidos. Também é possível sugerir a combinação das técnicas dessas abordagens com outras, como por exemplo, a Arquitetura Orientada a Serviços (*Service Oriented Architecture – SOA*) e Linhas de Produtos de Software (*Software Product Lines – LPS*).

Referências Bibliográficas

- Atkinson, C. e Kühne, T. (2003) “Model-Driven Development: A Metamodeling Foundation”, In: *IEEE Software*, v. 20, n. 5, p. 36-41.
- Balagtas-Fernandez, F. e Hussmann, H. (2009) “Applying Domain-Specific Modeling to Mobile Health Monitoring Applications”, In: *Proceedings of the 6th International Conference on Information Technology: New Generations (ITNG)*, p. 1682-1683.
- Beale, T. (2002). “Archetypes: Constraint-based domain models for future-proof information systems”, In: *Proceedings of the Workshop on Behavioural Semantics (OOPSLA)*, p. 1-18.
- Bergmann, N. W. (2010) “Better Design Methods for eHealth Software”, In: *International Journal of Engineering and Industries*, v. 1, n. 1, p. 1-9.
- Brown, A. (2004) “Model driven architecture: Principles and practice”, In: *Software and Systems Modeling*, v. 3, n. 4, p. 314-327.
- Ding, Y. e Klein, K. (2010) “Model-Driven Application-Level Encryption for the Privacy of E-health Data”, In: *Proceedings of the International Conference on Availability, Reliability and Security*, p. 341-346.
- Fernandes, J. E. M. (2010) “About Model-Based Approaches in Pervasive Information Systems Development”, Tese (Doutorado em Tecnologias e Sistemas de Informação) – Universidade do Minho, 183 f.
- Fernandes, J. E. M., Machado, R. J. e Carvalho, J. A. (2007) “Model-Driven Software Development for Pervasive Information Systems Implementation”, In: *Proceedings of the 6th International Conference on the Quality of Information and Communications Technology (QUATIC)*, p. 218-222.

- Garlan, D., Monroe, R. T. e Wile, D. (1997) “ACME: An Architecture Description Interchange Language”, In: Proceedings of the 7th Conference of the Centre for Advanced Studies on Collaborative Research (CASCON), p. 169–183.
- Gomes, A. T. A., Ziviani, A., Procópio, V. U., Moreira, V. M., Correa, B. S. P. M. e Teixeira, I. M. (2011) “Desenvolvimento Dirigido a Modelos para Aplicações em Saúde”, In: XXI Workshop De Informática Médica (WIM).
- Greenfield, J. e Short, K. (2004) “Software Factories: Assembling Applications with Patterns, Models, Frameworks and Tools”, John Wiley & Sons.
- Jones, V. (2006) “Model Driven Development of m-Health Systems (with a Touch of Formality)”, In: Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW), p. 580-585.
- Kelly, S. e Tolvanen, JP. (2008) “Domain-Specific Modeling: Enabling full code generation”, John Wiley & Sons.
- Khambati, A., Grundy, J., Warren, J. e Hosking, J. (2008) “Model-Driven Development of Mobile Personal Health Care Applications”, In: Proceedings of the 23rd IEEE/ACM International Conference on Automated Software Engineering, p. 467-470.
- Mellor, S., Clark, A. e Futagami, T. (2003) “Model-driven development - Guest editor's introduction”, In: *IEEE Software*. v. 20, n. 5, p. 14-18.
- Menezes, A. L., Cirilo, C. E., Moraes, J. L. C., Souza, W. L. e Prado, A. F. (2010) “Using Archetypes and Domain Specific Languages on Development of Ubiquitous Applications to Pervasive Healthcare”, In: Proceedings of the 23rd IEEE International Symposium on Computer-Based Medical Systems (CBMS), p. 395-400.
- Nardon, F. B., França, T. e Naves, H. (2011) “Construção de Aplicações em Saúde Baseadas em Arquétipos”, In: XXI Congresso Brasileiro de Informática em Saúde (CBIS).
- OMG (2003) “MDA Guide Version 1.0.1”, Technical Report.
- Raghupathi, W. e Umar, A. (2008) “Exploring a model-driven architecture (MDA) approach to health care information systems development”, In: *Int J Med Inform*, v. 77, p. 305–314.
- Santana, E. F. Z. (2010) “Uma Abordagem Orientada por Modelos para o Desenvolvimento de Software na Computação Ubíqua”, Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de São Carlos, 92 f.
- Selic, B. (2003) “The Pragmatics of Model-Driven Development”, In: *IEEE Software*, v. 20, n. 5, p. 19-25.
- Thomas, D. (2004) “MDA: revenge of the modelers or UML utopia?”, In: *IEEE Software*, v. 21, n. 3, p. 15- 17.