

Desenvolvimento Dirigido a Modelos para Aplicações em Saúde

A.T.A. Gomes, A. Ziviani, V.U. Procópio, V.M. Moreira, B.S.P.M. Correa, I.M. Teixeira

Laboratório Nacional de Computação Científica (LNCC) – Petrópolis, RJ – Brasil
Instituto Nacional de C&T em Medicina Assistida por Computação Científica (INCT-MACC)

{atagomes, ziviani, vmacedo, brunos, iuri}@lncc.br

Abstract. *We propose a new model-driven method for healthcare information systems that targets a higher degree of maintainability and reuse by employing crosscutting clinical data models and architecture models. To support this method on the automatic synthesis of such systems, we are progressing work on the development of a software workbench that weaves together clinical data models described according to the openEHR specifications and architecture models specified in the Acme language.*

Resumo. *Neste artigo propomos um novo método de desenvolvimento dirigido a modelos para sistemas de informação em saúde que objetiva um alto grau de manutenibilidade e reuso empregando transversalidade na concepção de modelos de dados clínicos e modelos arquiteturais. Para dar suporte a esse método na geração automática desses sistemas, estamos desenvolvendo um ferramental de software capaz de acoplar modelos de dados clínicos descritos de acordo com as especificações openEHR e modelos arquiteturais especificados na linguagem Acme.*

1. Introdução

Recentemente, tem havido considerável atenção sobre o potencial de Desenvolvimento Dirigido a Modelos – DDM no suporte ao desenvolvimento de Sistemas de Informação em Saúde – SIS. De forma geral, as iniciativas existentes advogam o uso do padrão MDA (*Model Driven Architecture*) [Miller e Mukerji 2003]. Conforme descrito em [Raghupathi e Umar 2008], a abordagem MDA viabiliza o desenvolvimento de SIS: (i) com baixo custo e *time-to-market*; (ii) interoperáveis; (iii) independentes de plataforma; e (iv) melhor manuteníveis. Contudo, em nossa visão, há dois desafios principais a serem ainda tratados para que DDM alcance um nível maior de interoperabilidade, manutenibilidade e reuso no desenvolvimento de SIS:

Complexidade na modelagem dos dados clínicos. Nossa experiência prévia no desenvolvimento de SIS [Teixeira et al. 2009] mostra que mudanças na modelagem de dados clínicos são constantes devido à complexidade de transcrever o conhecimento do profissional médico para dentro do sistema. Essa complexidade pode se traduzir em problemas de manutenibilidade e interoperabilidade a médio e longo prazo. Como consequência, a modelagem dos dados clínicos acaba exigindo atenção demasiada da equipe de desenvolvimento, sendo que constitui apenas uma parcela do esforço de desenvolvimento total de um SIS.

Modelagem arquitetural ad-hoc. Técnicas de projeto arquitetural são especialmente úteis no domínio de SIS. Nesse sentido, a abordagem MDA é capaz de facilitar o reuso de uma boa arquitetura, mas não auxilia o projeto de novas boas arquiteturas. Tipicamente, sistemas dentro de uma mesma família de SIS (p.ex., emergência pré-hospitalar, vigilância epidemiológica, monitoramento intra-hospitalar de pacientes) empregam estilos arquiteturais semelhantes (p.ex., cliente-servidor, *publisher-subscriber*, *peer-to-peer*). Defendemos que esses estilos devem poder ser melhor explorados pelas equipes de desenvolvimento visando projetos arquiteturais que propiciem manutenibilidade e reuso de longo prazo.

Esses dois desafios têm nos motivado a desenvolver um método de DDM para SIS mais abrangente que os baseados somente na abordagem MDA. Nosso método tem como objetivos:

Transversalizar modelos arquiteturais e de dados clínicos, de modo que especialistas em saúde não se importem em como os dados são processados, armazenados e transportados, e que arquitetos de software não se importem em como os dados são representados em SIS;

Dualizar modelos de dados clínicos, para separar a modelagem computacional da modelagem médica, o que permite reduzir o esforço da equipe de desenvolvimento na manutenção de SIS;

Estilizar modelos arquiteturais, para capturar características comuns dentro de famílias de SIS como artefatos reusáveis de projeto.

Para dar suporte ao método proposto, temos desenvolvido um ferramental de software inovador, que é apresentado na Seção 2. O estado atual de desenvolvimento do ferramental e as perspectivas futuras de aplicação desse ferramental são apresentados na Seção 3.

2. Ferramental de DDM Proposto

O ferramental proposto é apresentado na Figura 1. Todas as ferramentas ilustradas estão sendo prototipadas como *plugins* para o ambiente Eclipse¹ e são detalhadas adiante.

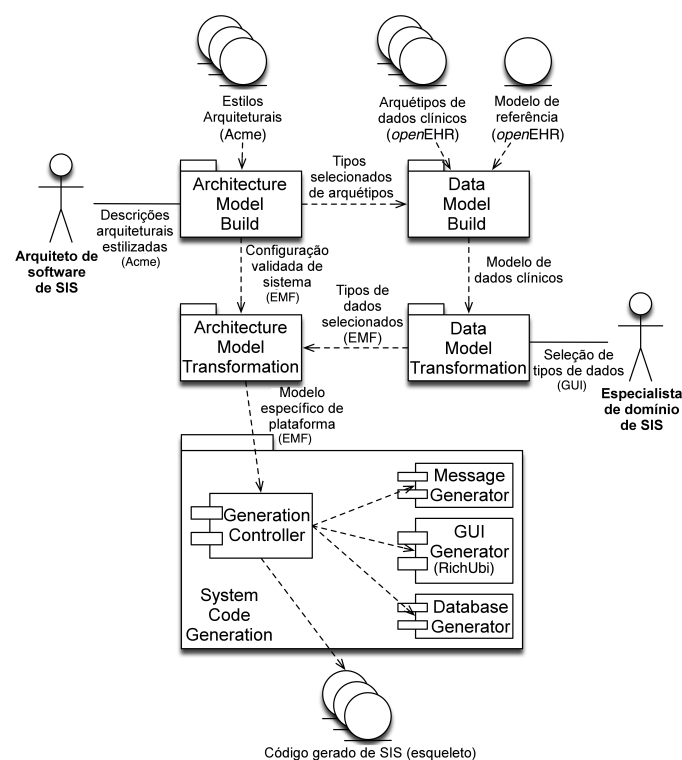


Figura 1. Ferramental de DDM proposto.

2.1. Transversalização de Modelos

Para transversalizar modelos arquiteturais e de dados clínicos, consideramos dois atores diferentes no método proposto: o *especialista de domínio* e o *arquiteto de software*. O primeiro

¹<http://www.eclipse.org>

especifica modelos de dados clínicos de SIS (vide Subseção 2.2) e o segundo modelos arquiteturais de SIS (vide Subseção 2.3). Esses dois modelos são transformados, conforme descrito nas subseções seguintes, em formatos compatíveis com o metamodelo EMF (*Eclipse Modeling Framework*)² e passados à ferramenta Architecture Model Transformation. Essa ferramenta implementa um motor de composição e transformação de modelos baseado em EMF que gera um modelo de plataforma específica – PSM (*Platform Specific Model*) também baseado em EMF, a partir do qual esqueletos de SIS são gerados (vide Subseção 2.4).

2.2. Modelagem de Dados Clínicos

Para dualizar modelos de dados clínicos, adotamos no método proposto o conjunto de especificações *openEHR* [Beale 2002]. No *openEHR*, a tarefa da modelagem da informação é mantida separada daquela da modelagem de dados clínicos. Modeladores de informação definem um *modelo de referência*, sobre o qual comitês multi-disciplinares de analistas de sistemas e médicos produzem os chamados *arquétipos de dados clínicos*, que restringem os tipos de dados definidos no modelo de referência. Por exemplo, no *openEHR* “Pressão sanguínea” é um arquétipo com tipos de dados como “Sistólica” e “Diastólica”, que definem restrições sobre os valores aceitos pelo tipo de dados mais geral “Quantidade” no modelo de referência.

No ferramental de software proposto, ambos o modelo de referência e os arquétipos de dados clínicos das especificações *openEHR* servem como entrada para a ferramenta *Data Model Build*. Essa ferramenta também recebe da ferramenta *Architecture Model Build* (vide Subseção 2.3) uma indicação sobre quais tipos de arquétipos de dados clínicos são de interesse – observação, instrução, avaliação ou ação – para a família específica de SIS da qual o sistema a ser gerado faz parte. A partir dessas entradas, a ferramenta *Data Model Build* gera um modelo completo de dados clínicos. Esse modelo é repassado para a ferramenta *Data Model Transformation*, que: (i) permite que o especialista selecione os tipos de dados no modelo de dados clínicos completo que são de interesse para o SIS específico a ser gerado – por exemplo, o tipo de dado “Muscle Tone” é crucial para obstetras (devido ao método Apgar), mas não faz sentido para serviços de emergência para pacientes de infarto do miocárdio; e (ii) traduz os tipos de dados selecionados em um formato compatível com EMF.

2.3. Modelagem Arquitetural

Para estilizar modelos arquiteturais, adotamos no método proposto a linguagem de descrição arquitetural Acme [Garlan et al. 1997]. Acme oferece construções de alto nível – como componentes, conectores e configurações – que incorporam informações sobre a estrutura de um sistema abstraindo detalhes da implementação do mesmo. Acme também provê suporte para *estilos arquiteturais* [Garlan et al. 1994], os quais prescrevem tipos de construções e restrições na composição dos mesmos. Por exemplo, um estilo “Pipeline” pode incluir um tipo de componente “Filter” e um tipo de conector “Pipe” além de restrições que impedem composições circulares de tais construções em uma configuração de sistema. Uma configuração seguindo um estilo específico em Acme pode assim ser vista como uma instância de tal estilo.

No ferramental de software proposto, estilos arquiteturais descritos em Acme servem como entrada para a ferramenta *Architecture Model Build*. Tais estilos definem regras para a configuração dos sistemas pertencentes a diferentes famílias de SIS – até o momento definimos estilos Acme para sistemas de emergência pré-hospitalar e vigilância síndrômica. A

²<http://www.eclipse.org/modeling/emf/>

ferramenta **Architecture Model Build**: (i) valida uma configuração de sistema especificada por um arquiteto de software de acordo com as regras de estilo da família SIS pretendida; (ii) indica à ferramenta **Data Model Build** os tipos de arquétipos de dados clínicos de interesse para essa família – por exemplo, sistemas de emergência manipulam a maior parte dos dados de observação e instrução, enquanto que sistemas de vigilância estão normalmente interessados em dados de observação e avaliação; e (iii) traduz a configuração validada de sistema em um formato compatível com EMF.

2.4. Geração de Código

A ferramenta **System Code Generation** implementa um motor de transformação baseado em EMF que gera esqueletos de código para SIS a partir de um PSM exportado pela ferramenta **Architecture Model Transformation**. A ferramenta **System Code Generation** é composta por quatro componentes. O componente **Generation Controller** analisa o PSM e invoca um dos três componentes geradores específicos dependendo da parte do PSM que está sendo analisada: (i) o componente **Message Generator** para a geração de entidades de protocolo e formatos de mensagens (p.ex. definições WSDL, *stanzas* XMPP) (ii) o componente **GUI Generator**, que é baseado no RichUbi, uma ferramenta baseada em EMF para geração de código para interfaces gráficas de usuário [Cirilo et al. 2010]; e (iii) o componente **Database Generator** para a geração de código específico de banco de dados (p.ex., esquemas, mapeamentos O/R).

3. Perspectivas e Trabalhos Futuros

Até o momento, todas as ferramentas apresentadas, à exceção da **Architecture Model Transformation**, já foram prototipadas – a composição dos modelos arquiteturais e de dados clínicos ainda é feita manualmente. Temos testado o ferramental em condições experimentais usando EJB, AJAX, Webservices e Hibernate como plataforma alvo. Como trabalho futuro, pretendemos validar a proposta completa através da modelagem de SIS existentes e da comparação dos esqueletos de código gerados pelo ferramental proposto com tais sistemas.

Referências

- Beale, T. (2002). Archetypes: Constraint-based domain models for future-proof information systems. *openEHR standard document*.
- Cirilo, C. E., do Prado, A. F., de Souza, W. L., e Zaina, L. A. M. (2010). Model driven RichUbi: A model driven process for building rich interfaces of context-sensitive ubiquitous applications. Em *Proc. of the 28th SIGDOC*, pág. 207–214, New York, NY, USA. ACM.
- Garlan, D., Allen, R., e Ockerbloom, J. (1994). Exploiting style in architectural design environments. *SIGSOFT Softw. Eng. Notes*, 19:175–188.
- Garlan, D., Monroe, R. T., e Wile, D. (1997). ACME: An Architecture Description Interchange Language. Em *CASCON'97: Proceedings*, pág. 169–183, Toronto, Ontario.
- Miller, J. e Mukerji, J. (2003). MDA guide version 1.0.1. Technical report, OMG.
- Raghupathi, W. e Umar, A. (2008). Exploring a model-driven architecture (MDA) approach to health care information systems development. *Int J Med Inform*, 77:305–14.
- Teixeira, I. M., Vicoso, R. P., Correa, B. S. M. P., Gomes, A. T. A., e Ziviani, A. (2009). Suporte remoto ao atendimento médico emergencial via dispositivos móveis. Em *Anais do IV Workshop de Informática Médica (WIM)*, Porto Alegre, RS, Brasil. SBC.