

# Síntese automática de Interfaces Gráficas de Usuário para Sistemas de Informação em Saúde

Iuri M. Teixeira<sup>1</sup>, Regina M. Braga<sup>1</sup>, Antônio Tadeu A. Gomes<sup>2</sup>

<sup>1</sup>Universidade Federal de Juiz de Fora (UFJF) – Juiz de Fora, MG – Brasil

<sup>2</sup>Laboratório Nacional de Computação Científica (LNCC)  
Instituto Nacional de Ciência e Tecnologia em Medicina Assistida por Computação Científica (INCT-MACC) – Petrópolis, RJ – Brasil

malinoski.iuri@gmail.com, regina.braga@ufjf.edu.br, atagomes@lncc.br

**Resumo.** *A modelagem de dados clínicos para Sistemas de Informação em Saúde (SIS) demanda expertise de domínio. Técnicas de Desenvolvimento Dirigido por Modelos (DDM) permitem uma melhor articulação entre especialistas de domínio e desenvolvedores e possibilitam reduzir o custo de desenvolvimento de sistemas de software. Modelos de dados clínicos baseados em especificações padronizadas e abertas como a do openEHR facilitam sobremaneira a aplicação de técnicas de DDM para SIS. Contudo, o uso de modelos de dados clínicos não resolve sozinho o problema fundamental do alto custo de desenvolvimento de SIS. Um desses problemas é a falta de informações arquiteturais nestes modelos, que são deslocados para os processos de transformação de modelos. A partir deste contexto, apresentamos um framework para geração de código baseado em modelos de dados clínicos considerando as especificações openEHR e informações arquiteturais baseado em algumas construções presentes em linguagens de descrição arquitetural (ADL), como Acme. Aplicamos neste framework técnicas de DDM na síntese automática de interfaces gráficas de usuário (GUI – Graphic User Interface) para SIS, e discutimos sobre a hipótese de que a mesma pode melhor articular especialistas de domínio e desenvolvedores de SIS, ao mesmo tempo em que simplifica a especificação dos processos de transformação de modelos.*

## Informações Gerais

O ingresso do aluno Iuri Malinoski Teixeira no programa de pós-graduação da Universidade Federal de Juiz de Fora (UFJF) na área de Engenharia de Software foi iniciado em março de 2011 e tem previsão de término em fevereiro de 2013. A Prof<sup>a</sup> Regina Braga do Programa da Pós-Graduação em Ciência da Computação da Universidade Federal de Juiz de Fora (PGCC) e o Prof. Antônio Tadeu A. Gomes do Laboratório Nacional de Computação Científica (LNCC) são responsáveis, respectivamente, pela orientação e co-orientação do aluno.

## Palavras-chave

Sistema de Informação em saúde, Modelos de dados clínicos, Desenvolvimento Dirigido por Modelos, Transformações de modelos.

## 1. Caracterização do Problema

O emprego de técnicas de Desenvolvimento Dirigido por Modelos (DDM) [Stahl e Völter, 2012] para Sistemas de Informação em Saúde (SIS) ainda é um desafio importante a ser tratado para que se alcance um maior nível de interoperabilidade, manutenibilidade e reuso [Haux, 2010]. Modelos de dados clínicos baseados em especificações padronizadas e abertas, como o *openEHR* [Beale, 2002], podem facilitar sobremaneira a aplicação de técnicas de DDM para SIS. Este tipo de especificação permite uma melhor articulação entre especialistas de domínio e desenvolvedores, além de facilitar a geração de código para SIS baseado nestes modelos. No entanto, modelos de dados clínicos não resolvem sozinho o problema fundamental do alto custo de desenvolvimento de SIS. O processo de transformação de modelos [Czarnecki e Helsen, 2006] é uma das técnicas de DDM que pode minimizar este problema. Esses processos são cruciais em DDM e, tipicamente, variam entre diferentes famílias de SIS, como sistemas de atendimento emergencial e de vigilância epidemiológica [Braga et al 2012, Gomes et al 2012]. Diante deste contexto, modelos de dados clínicos não são suficientes para a síntese automática de código para SIS, uma vez que informações importantes nesse processo, em particular informações arquiteturais, não são encontradas nesses modelos.

O objetivo deste trabalho é apresentar um *framework* para a síntese automática de interfaces gráficas de usuário (GUI – *Graphic User Interface*) para SIS através de modelos de dados clínicos associados a informações arquiteturais. Foi desenvolvida para esse *framework* uma técnica de DDM baseada em um repositório de regras reusáveis de transformações de modelos baseadas em padrões de projetos [Gamma et al. 1995]. Neste contexto, apresentamos uma estratégia para geração de código para diferentes famílias de SIS inspirada em algumas construções para definição de estilos arquiteturais oferecidas por linguagens de descrição arquitetural (ADL) como Acme [Garlan, 1997]. A estratégia consiste na definição de um estilo arquitetural que descreve propriedades de GUI. Essas propriedades permitem enriquecer componentes definidos em famílias de SIS – estas também descritas como estilos arquiteturais – com informações de GUI. Nessa estratégia, não só a implementação de SIS, mas também a especificação de regras de transformação pode ser simplificada, permitindo reduzir o custo para o desenvolvimento de SIS.

## 2. Fundamentação Teórica

Técnicas de DDM baseiam-se no fato de que mudanças realizadas em um nível mais abstrato de modelagem possam ser propagadas para níveis mais concretos de projeto, mitigando redundâncias e inconsistências no processo geral de desenvolvimento de software. Dois conceitos centrais de DDM que viabilizam o princípio de especialização de modelos abordado acima são *metamodelos* e *transformação de modelos*.

Um metamodelo é um modelo que representa os metadados dos elementos de modelagem, e das associações entre esses metadados, que podem ser usados na construção de modelos que instanciam esse metamodelo. Esse processo de construção também vale para o metamodelo em si, ou seja, um metamodelo possui um ou mais metamodelos correspondentes. Neste trabalho adotamos um subconjunto do metamodelo MOF (*Meta Object Facility*) do OMG (*Object Management Group*)

[OMG 2012] implementado na plataforma Eclipse [Eclipse Foundation, 2012], denominado Ecore, como metamodelo que servirá de base das técnicas de DDM desenvolvida neste trabalho.

Os processos de transformação de modelos são divididos em duas categorias [Czarnecki e Helsen, 2006]: modelo para modelo (M2M); e modelo para código (M2C). Há diferentes abordagens para transformações de modelos, sendo a especificação da OMG uma das mais utilizadas. Nessa especificação, um ou mais modelos de destino são gerados a partir de um modelo de origem com base em um conjunto de regras de transformação baseadas nos metamodelos de origem e de destino. A linguagem de transformação ATL (*Atlas Transformation Language*) [Jouault et al, 2008], desenvolvida para a plataforma Eclipse, segue as especificações da OMG. Neste trabalho adotamos a linguagem ATL devido à sua maturidade de definições, documentação abrangente, e integração com o metamodelo Ecore para a plataforma Eclipse.

### 3. Caracterização da Contribuição

Esta seção apresenta o *framework* desenvolvido no contexto deste trabalho, denominado IMT (Interface Model Transformation Framework), para a geração de código de GUI para SIS. Este *framework* tem como entrada os modelos de dados clínicos desejados e o modelo arquitetural do SIS pretendido e gera como saída código de GUI. Como ilustrado na Figura 1, o *framework* proposto possui três elementos principais para a sua funcionalidade: “M2M Transformation”, “M2C Transformation” e “Transformation Rules Repository”. Os dois primeiros elementos são os componentes do *framework* IMT que executam transformações, respectivamente, para processos de M2M e M2C. O terceiro elemento é um repositório de regras de transformação de modelos.

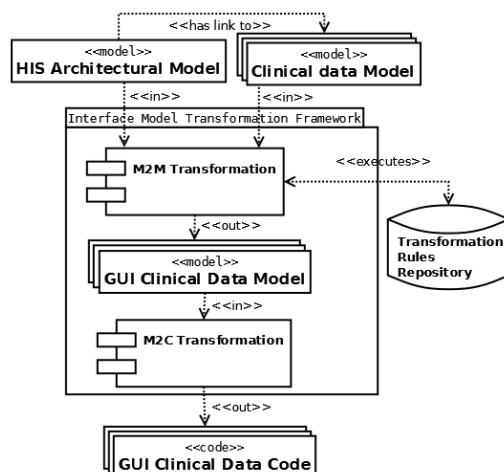


Figura 1. Gerador de código para Sistemas de Informação em Saúde

Na Figura 1, o componente de transformações M2M realiza a síntese entre modelos de dados clínicos (“Clinical Data Model”) relacionados ao modelo arquitetural de SIS pretendido (“HIS Architectural Model”). A partir destes modelos, são executadas as regras de transformações localizadas no repositório de regras reutilizáveis (“Transformation Rules Repository”). O resultado do

processo de transformações M2M são modelos abstratos com as principais informações de domínio para GUI independentes de plataforma (“GUI Clinical Data Model”). O processo de transformações M2C realiza transformações dos modelos recebidos do componente de transformações M2M para modelos concretos de GUI (“GUI Clinical Data Code”). Essa transformação utiliza a ferramenta RichUbi [Cirilo et al, 2010] para gerar código em JSP (*JavaServer Pages*) e HTML (*HyperText Markup Language*), entre outros. Entretanto, é importante frisar que outras ferramentas para processos de transformações M2C podem ser utilizadas, uma vez que as principais informações de domínio, especializadas em GUI, estão contidas nos modelos gerados pelo componente “M2M Transformation”.

### **3.1. Uma Estratégia para Transformações de Modelos de Dados Clínicos com Informações Arquiteturais**

Uma hipótese importante levantada neste trabalho é a de que transformações de modelos de dados clínicos sem considerar informações arquiteturais pode levar a uma explosão combinatória de regras e, conseqüentemente, a um esforço cognitivo adicional para os desenvolvedores de regras de transformações. Como exemplos, podemos mencionar famílias de SIS como emergência pré-hospitalar e vigilância epidemiológica, discutidas em [Braga et al 2012, Gomes et al 2012], onde diferentes combinações de tipos de dados clínicos ocorrem devido a presença de distintas estruturas arquiteturais:

- Em uma família de SIS de suporte remoto a emergência pré-hospitalar, identificam-se dois componentes arquiteturais fundamentais relacionados a GUI: o “emergencista”, que representa o primeiro atendimento ao paciente; e o “especialista”, que auxilia remotamente o emergencista na tomada de decisão quanto ao procedimento clínico a ser adotado no atendimento. Um exemplo de SIS pertencente a essa família, focado no atendimento emergencial de vítimas de infarto, é descrito em [Correa et al, 2011]. Nessa família, dados de observação (como “pressão sanguínea”) são tipicamente associados a “elementos de formulário” no componente emergencista e a “elementos de visualização” no componente especialista, enquanto o inverso acontece para dados de instrução (como “intervenção a ser adotada”).
- Em um sistema de vigilância epidemiológica, diferentes combinações ocorrem devido à presença de diferentes estruturas arquiteturais (um *survey* desses sistemas é apresentado em [Job et al, 2011]). Contudo, de forma geral, há uma maior relevância de dados clínicos de observação e avaliação (como “clusters de casos de cólera”), em contraposição aos dados de observação e instrução dos sistemas de atendimento emergencial, além de uma maior tendência a unidirecionalidade dos fluxos de dados envolvidos. Na família de sistemas de vigilância epidemiológica são identificados dois componentes arquiteturais fundamentais relacionados a GUI: “coletor” e “epidemiologista”. O primeiro componente realiza a coleta de dados clínicos de observação em elementos de formulários. O segundo componente permite a um epidemiologista analisar os dados coletados, portanto, este componente está relacionado a elementos de visualização.

A estratégia proposta neste trabalho para geração de código GUI considera a hipótese discutida acima. A representação das famílias de SIS neste trabalho é baseada em metamodelos ECore e foi inspirada em algumas construções presentes em ADLs como Acme. Essas construções seriam *propriedades*, *componentes* e *estilos arquiteturais*. A estratégia consiste na definição de um estilo arquitetural base para descrever uma família de propriedades relacionadas a GUI. Essas propriedades enriquecem componentes de famílias específicas de SIS, famílias estas também definidas como estilos arquiteturais que derivam do estilo arquitetural base acima. As propriedades do estilo arquitetural base permitem associar tipos de dados clínicos (como observação e instrução) a tipos de elementos de GUI (como formulário e visualização), sendo essas associações definidas de forma particular para cada família de SIS.

Como ilustrado na Figura 2, todos os metamodelos necessários para o processo de transformações M2M foram definidos como instâncias do metamodelo “Ecore”. O relacionamento proposto entre modelos e metamodelos são necessários para o desenvolvimento e execução das regras de transformações “ATL Rules”. A partir deste relacionamento, diferentes famílias de SIS (“HIS Architecture Family”) podem utilizar a mesma família de propriedades para enriquecer componentes arquiteturais com propriedades de GUI (“GUI Clinical Property Family”). As regras ATL têm como entrada os modelos de dados clínicos desejados (“Clinical Data Model”) e o modelo arquitetural de SIS pretendido (“HIS Architectural Model”). Quando as regras são executadas, o componente para transformações M2M do *framework* proposto gera modelos abstratos de GUI para dados clínicos (“GUI Clinical Data Model”).

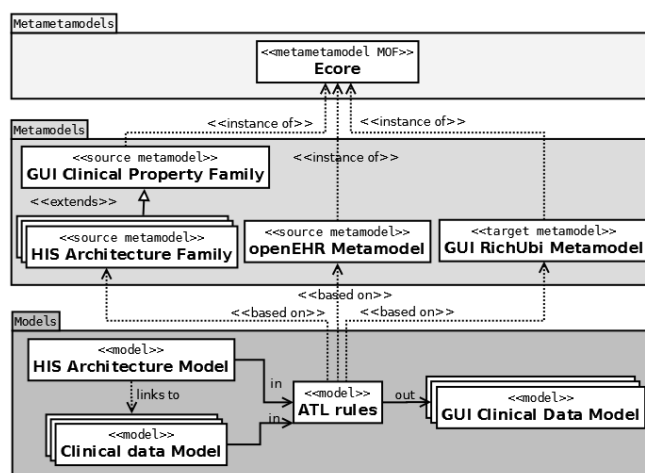
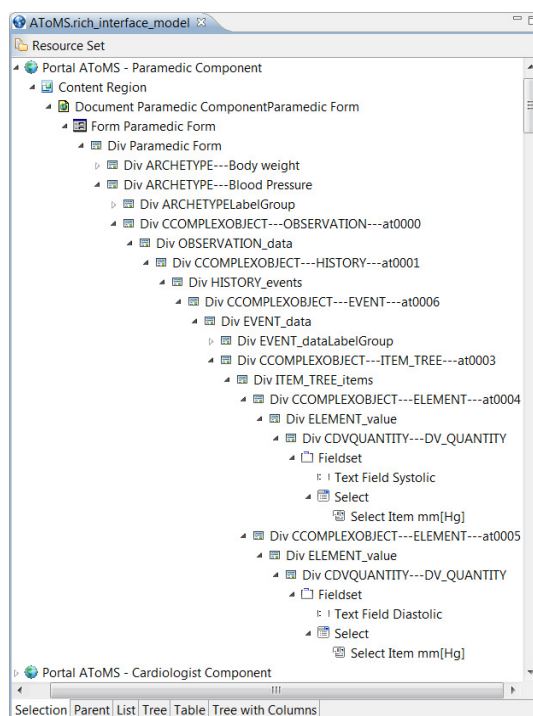


Figura 2. Relacionamento entre modelos e metamodelos para as transformações de modelos proposta.

### 3.3. Cenário de Uso

A fim de testar as idéias, considerando o *framework* proposto, apresentamos um cenário de uso para a geração automática de código GUI para SIS a partir da síntese entre modelos de dados clínicos (arquétipos) desejados e do modelo de SIS pretendido. Nesta aplicação prática foi utilizada a família de sistemas de emergência pré-hospitalar e a família de sistemas de vigilância na saúde, discutidos anteriormente. A família de propriedade GUI foi desenvolvida para enriquecer estas duas famílias. A família de

propriedade GUI, que é uma família abstrata para um estilo arquitetural base, foi especificada em formato Ecore. Bem como as duas famílias SIS, que são as famílias de domínios, considerou-se apenas os componentes de arquitetura destas famílias. Através do trabalho proposto foi possível gerar distintos códigos de GUI a partir de arquétipos (do modelo *openEHR*) considerando informações arquiteturais.



**Figura 3. Modelos de dados clínicos para formulário de GUI.**

Por questão de espaço será demonstrada a geração de código para a família de sistemas de emergência pré-hospitalar que consiste na geração de código GUI para modelos de dados clínicos (arquétipos) considerando um sistema de emergência, discutido anteriormente, para as vítimas de infarto do miocárdio. Este sistema pode ser definido a partir de uma família de sistemas de emergência pré-hospitalar considerando o metamodelo Ecore baseado em ADL. Essa família pode ser enriquecida com propriedades de GUI em seus componentes arquiteturais “emergencista” e “especialista”. Os arquétipos de “pressão sanguínea” e de “peso corporal”, que são do tipo de dado clínico “observação”, são necessários para o sistema supracitado. Considerando uma família de sistemas de emergência pré-hospitalar com os metadados que representam os componentes “emergencista” e “especialista”, o tipo de dado clínico “observação” é restrito aos metadados que definem esses componentes, respectivamente, com propriedades de GUI para “formulários para observação” e “visualizações para observação”. Por razões de espaço, a Figura 3 mostra apenas uma síntese do arquétipo “pressão sanguínea” como uma consequência das regras de transformações realizada pelo processo de transformações de M2M do *framework* proposto.

O repositório de regras do *framework* proposto é constituído por três conjuntos de regras para realizar as devidas transformações. Para cada componente arquitetural encontrado pelo primeiro grupo de regras, esses componentes são transformados em

portais *web*. O segundo grupo de regras realiza a ligação para o terceiro grupo de regras através das identificações de arquétipos (como arquétipos de observação "pressão sanguínea" e "peso corporal") definidas nas propriedades de GUI ("formulário para observação" e "visualização para observação"). Através dos modelos arquiteturais (família de domínio e propriedades de GUI), o terceiro grupo de regras pode realizar as devidas transformações dos arquétipos para "elementos de formulários" ou para "elementos de visualização". A Figura 4 ilustra, de maneira simplificada, o código resultante destes modelos abstratos de GUI (HTML).

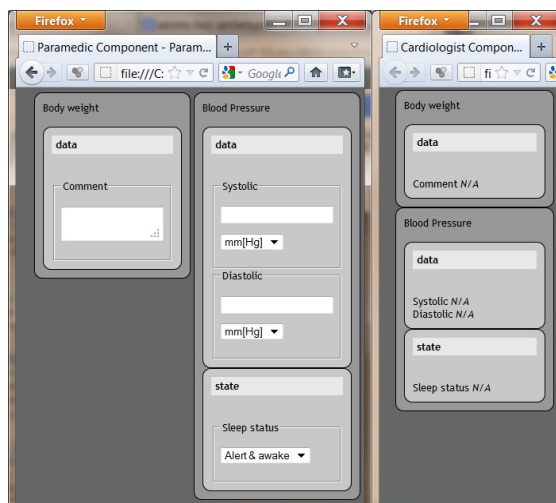


Figura 4. Código resultante dos modelos abstrato de GUI

## 5. Comparação com Trabalhos Relacionados

Trabalhos como os encontrados em Schuler et al (2006), Nardon et al (2007), Linden et al (2009) e Atalag e Yang (2010), exploram, segundo diferentes perspectivas e com diferentes resultados, a geração de código de GUI para SIS a partir de dados clínicos baseados nas especificações *openEHR*. Entretanto, essas abordagens não empregam metamodelagem e o uso de regras de transformação formalmente especificadas, o que compromete o reuso durante a fase de projeto do software. Costa (2011) propõe uma abordagem para construção de interfaces de usuário para sistemas de informação em geral (não restrita a SIS) por meio da interação com o usuário. Este trabalho utiliza metamodelagem e regras de transformação, mas não se baseia em um metamodelo comum (como MOF ou Ecore) tanto aos metamodelos como às regras, o que compromete tanto o reuso de projeto como a interoperabilidade do software resultante. Além disso, a ausência de tratamento específico para modelos de dados clínicos torna inviável a geração de código sem perder conceitos de domínio.

Os trabalhos acima se caracterizam por gerar apenas interfaces gráficas. Entretanto, a proposta de Menárguez-Tortosa (2011) aborda a geração automática de código para SIS completos através de técnicas de DDM para informações médicas. Os SIS gerados são formulários *web* para preenchimento de dados clínicos que são enviados por mensagens padronizadas, tornando possível a alimentação de bases de dados que seguem esse padrão, garantindo interoperabilidade por construção. Contudo, não é explorada nesse artigo a questão de que diferentes famílias de SIS demandam diferentes tratamentos para os dados clínicos em diferentes componentes de um SIS, o

que só é possível de ser feito quando os dados clínicos são enriquecidos com informações arquiteturais, como proposto neste trabalho.

## 6. Conclusão

Técnicas de DDM oferecem suporte para a resolução de problemas inerentes ao desenvolvimento de SIS e processos de transformações de modelos são uma das peças fundamentais nessas técnicas. Modelos de dados clínicos possuem um grande potencial para o estabelecimento de processos de DDM, entretanto não são suficientes para gerar SIS completos. Diante deste contexto, este trabalho propõe uma arquitetura para a síntese automática de código de SIS baseada na associação entre modelos de dados clínicos e informações arquiteturais.

O trabalho apresentado emprega metamodelagem e regras de transformação baseada em padrões de projetos formalmente especificadas no metamodelo Ecore, o que permite o reuso de projeto e a interoperabilidade por construção do software resultante. Foi apresentada uma estratégia que modelos de dados clínicos são utilizados para a geração de código para SIS sem perder conceitos de domínio. Esse artigo explora o fato de que diferentes famílias de SIS demandam diferentes tratamentos para os dados clínicos relevantes em diferentes componentes de SIS, o que só é possível de ser feito quando os dados clínicos são enriquecidos com informações arquiteturais, como proposto em neste trabalho. Foi apresentada uma estratégia de transformações de modelos para geração de código de GUI considerando distintas famílias de SIS. Essa estratégia foi inspirada em algumas construções para definição de estilos arquiteturais oferecidas por linguagens de descrição arquitetural ADL como Acme. Como trabalho futuro, está sendo investigada a extensão do *framework* proposto para facilitar a geração de código utilizando outros construtores para definição de estilos arquiteturais como *connectors*, *ports*, *constraints*, *rules*, *attachments*, *bindings*, entre outros.

## Referências

- A. T. A. Gomes, A. Ziviani, B. S. P. M. Correa, I. M. Teixeira, e V. M. Moreira. SPLiCE: a software product line for healthcare. IHI '12, pages 721-726. 2012. ACM.
- Braga, R. M.; Cavalini, L. T.; Cirilo, C. E.; Cook, T. W.; Correia, B. S. P. M.; Freire, S. M. ; Gomes, A. T. A. ; Moreira, V. M.; Menezes, A.; Moraes, J. L. C.; Prado, A. F.; Souza, W. L.; Teixeira, I. M. ; Ziviani, A.. Scientific Computing Applied to Medicine and Healthcare. 1ed. Petrópolis, RJ - Brasil: 2012, v. , p. 315-354.
- B. S. P. M. Correa, B. Gonçalves, I. M. Teixeira, A. T. A. Gomes, e A. Ziviani. AToMS: a ubiquitous teleconsultation system for supporting AMI patients with prehospital thrombolysis. *Int. J. Telemedicine Appl.*, 2011:2:1-2:12, Jan. 2011.
- C. E. Cirilo, A. F. Prado, W. L. D. Souza, e L. A. M. Zaina. Model driven RichUbi: a model driven process for building rich interfaces of context-sensitive ubiquitous applications. *Proceedings of the 28th*, pages 207-214, 2010.
- D. H. Job, A. T. A. Gomes, e A. Ziviani. Health Systems for Syndromic and Epidemiological Surveillance. IGI Global, 2011, p. 245-262.
- Eclipse Foundation (2012). Eclipse Foundation. Disponível em: <http://www.eclipse.org> (Acesso: novembro de 2012).



- Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. *Design Patterns: Elements of Reusable Object-Oriented Software*. November 1994.
- F. B. Nardon, T. França, e H. Naves. Construção de aplicações em saúde baseadas em arquétipos. *Acta Scientiarum Biological Sciences*, 28(4), 2007.
- F. Jouault, F. Allilaire, J. Bézivin, e I. Kurtev. ATL: A model transformation tool. *Sci. Comput. Program.*, 72(1-2):31-39, June 2008.
- Gamma, E.; Helm, R.; Johnson, R. ; Vlissides, J. *Design Patterns*. Boston, MA: Addison-Wesley, January 1995.
- Garlan, D., Monroe, R.T., Wile, D.: Acme: An architecture description interchange language. In: *Proceedings of CASCON'97*. pp. 169–183. Toronto, Ontario (1997)
- H. Van Der Linden, T. Austin, e J. Talmon. Generic screen representations for future-proof systems, is it possible? there is more to a gui than meets the eye. *Computer Methods and Programs in Biomedicine*, 95(3):213-226, 2009.
- K. Atalag e H. Y. Yang. From openEHR Domain Models to Advanced User Interfaces: A Case Study in Endoscopy. *Health Informatics New Zealand Conference*, 2010.
- K. Czarnecki e S. Helsen. Feature-based survey of model transformation approaches. *IBM Systems Journal*, 45(3):621-646, 2006.
- M. Eichelberg, T. Aden, J. Riesmeier, A. Dogac, e G. B. Laleci. A survey and analysis of electronic healthcare record standards. *ACM Comput. Surv.*.
- M. Menárguez-Tortosa, C. Martínez-Costa, e J. T. Fernández-Breis. A generative tool for building health applications driven by iso 13606 archetypes. *J Med Syst*, 2011.
- OMG (2012). Object Management Group (OMG). Disponível em: <http://www.omg.org> (Acesso: novembro de 2012).
- S. L. da Costa. Uma Abordagem Baseada em Modelos para Construção Automática de Interfaces de Usuário para Sistemas de Informação. Dissertação de Mestrado. Goiânia, UFG, 2011.
- R. Haux. Medical informatics: Past, present, future. *I. J. Medical Informatics*, 79(9):599-610, 2010.
- T. Beale. Archetypes: Constraint-based domain models for future-proof information systems. *openEHR Standard document*, 2002.
- T. Schuler, S. Garde, S. Heard, e T. Beale. Towards automatically generating graphical user interfaces from openehr archetypes. *Studies in Health Technology and Informatics*, pages 221-226. IOS Press, 2006.
- T. Stahl e M. Völter. *Model-Driven Software Development: Technology, Engineering, Management*. Wiley, Chichester, UK, 2006.