Pesquisando Soluções para Desafios em MDE com Kuaba

Adriana Pereira de Medeiros

Departamento de Computação, Instituto de Ciência e Tecnologia
Universidade Federal Fluminense (UFF) – Rio das Ostras, RJ – Brasil

adrianamedeiros@id.uff.br

Abstract. *Model-Driven Engineering (MDE) research has made substantial progress in recent years. However, there are still technical challenges, some related to the uncertainty during the design of software models, which is caused by many design alternatives, incomplete information and conflicting stakeholders opinions. The Kuaba approach for supporting design rationale addresses the causes of this uncertainty. This paper presents some research perspectives of an ongoing project that investigates how the Kuaba approach can be used in the development of solutions to these challenges in MDE.*

Resumo. *A pesquisa em Engenharia dirigida por modelos (MDE – Model-Driven Engineering) tem alcançado grande progresso nos últimos anos. No entanto, ainda há desafios técnicos, alguns relacionados à incerteza durante o design de modelos de software, que é causada por muitas alternativas de design, informações incompletas e opiniões conflitantes dos stakeholders. A abordagem Kuaba para design rationale trata as causas dessa incerteza. Este artigo apresenta algumas perspectivas de pesquisa de um projeto em andamento que investiga como Kuaba pode ser utilizada no desenvolvimento de soluções para esses desafios em MDE.*

1. Introdução

A pesquisa em Engenharia dirigida por Modelos (MDE - *Model Driven Engineering*) [Schmidt 2006] tem alcançado grande progresso nos últimos anos. A definição de padrões em modelagem forneceu as bases necessárias para pesquisas mais avançadas sobre ferramentas de design e gerenciamento de modelos. Pesquisas mais recentes possibilitam o uso de modelos em tempo de execução para gerenciar e entender os sistemas depois que eles foram implantados [Bucchiarone et al. 2020]. Na indústria a MDE tem sido utilizada de várias maneiras, desde esforços para definir modelos precisos para um domínio de aplicação, até usos muito restritos e limitados na geração de código para uma família de aplicações em uma única empresa [Whittle et al. 2014].

Apesar desse progresso, ainda há desafios técnicos a serem investigados para a melhoria das soluções oferecidas em MDE. Alguns deles foram identificados em dois eventos recentes, *Grand Challenges in MDE 2017 workshop* e *Winter Modeling Meeting 2018,* e apresentados em [Bucchiarone et al. 2020]. Entre esses desafios técnicos estão aqueles relacionados à incerteza durante o design de modelos de software, causada por muitas alternativas de design, informações incompletas e opiniões conflitantes dos *stakeholders*, comuns em sistemas que executam em ambiente aberto. Assim, como usar a MDE para (i) conectar modelos de discussão com artefatos de software, (ii) relacionar modelos diferentes a escolhas diferentes, (iii) detectar soluções

propostas para cada escolha, e (iv) alavancar/integrar ferramentas de modelagem flexíveis, são questões a serem levadas em consideração [Bucchiarone et al. 2020].

Este artigo apresenta algumas perspectivas de pesquisa relacionadas ao uso da abordagem Kuaba [Medeiros e Schwabe 2008] no desenvolvimento de soluções para essas questões. Kuaba apoia a captura, representação e processamento de *design rationale* em designs baseados em modelo. Para isto, ela usa uma representação baseada em argumentação que permite registrar toda a deliberação para a tomada de decisão sobre o design de um artefato. Essa representação inclui os problemas de design analisados, as alternativas de solução avaliadas para cada problema e o raciocínio usado para a escolha de uma alternativa em detrimento de outra.

O uso da abordagem Kuaba em soluções MDE pode contribuir para a compreensão dos modelos utilizados e para simulação de diferentes soluções de design antes da geração de código. Além disso, possibilita o registro de todo o processo de decisão durante a modelagem de software. Por tornar os principais elementos de decisão explícitos, pode facilitar a colaboração entre os desenvolvedores permitindo a análise das possíveis soluções de design avaliadas para um modelo, e a consulta aos prós e contras de cada solução discutidos durante a modelagem.

O restante deste artigo está organizado da seguinte forma: a seção 2 apresenta uma visão geral da abordagem Kuaba e sua implementação; a seção 3 aborda o uso da abordagem Kuaba no desenvolvimento de soluções para os desafios de pesquisa mencionados acima e a seção 4 apresenta algumas considerações finais sobre o projeto.

2. A abordagem Kuaba

A abordagem Kuaba permite a captura, representação e processamento de *design rationale* e tem como objetivo apoiar o reúso de designs baseados em modelo. A ontologia Kuaba, utilizada como modelo de representação, estrutura o *design rationale* em três elementos básicos: questões, que representam os problemas de design, ideias para solucionar esses problemas e argumentos que podem ser contra ou a favor dessas ideias. A ontologia também possui elementos específicos para registrar as decisões tomadas, as justificativas para aceitação ou rejeição das possíveis soluções de design, os artefatos gerados a partir das soluções aceitas e informações sobre a atividade de design, como o metamodelo utilizado e as pessoas envolvidas. A abordagem Kuaba usa a semântica fornecida pelo metamodelo utilizado no design para instanciar os elementos básicos dessa ontologia. A Figura 1 mostra uma visão geral da abordagem Kuaba.

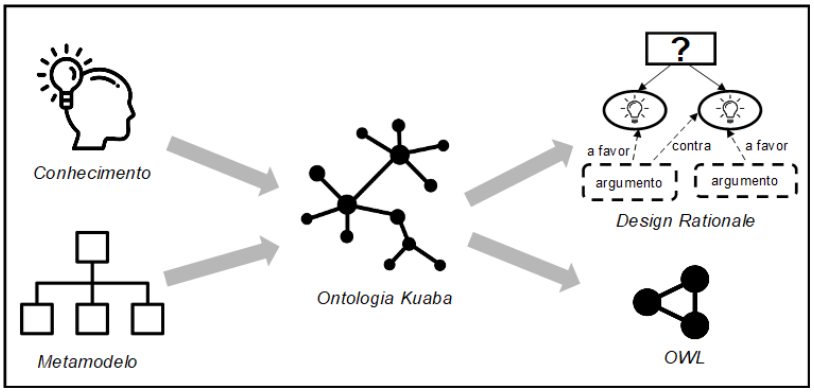

Figura 1 – Abordagem Kuaba

Em Kuaba, os itens de informação do domínio que está sendo modelado e os argumentos (conhecimento tácito) a favor ou contra as ideias consideradas durante a modelagem são fornecidos pelos especialistas de domínio ou desenvolvedores. As questões e as possíveis ideias de design sobre como modelar os itens do domínio são extraídas automaticamente do metamodelo utilizado. A representação final do *design rationale* é feita com a linguagem OWL (*OWL Web Language*) [W3C 2012], utilizada na Web Semântica. Isto permite o processamento computacional do *design rationale* para apoiar consultas sobre o design, a realização de inferências a partir do conhecimento registrado e o reúso dos modelos produzidos. A Figura 2 mostra um exemplo de representação de *design rationale* durante a produção de um modelo de análise extraindo informações do metamodelo da UML para diagrama de classes.

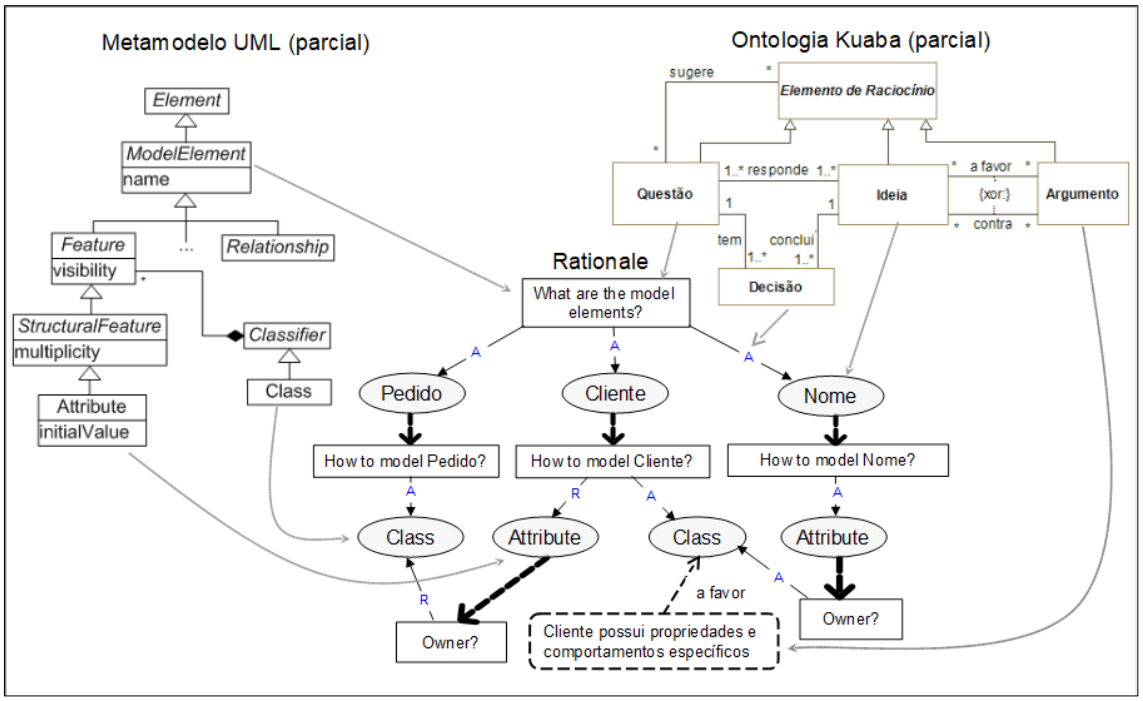

Figura 2. Exemplo de representação de *Design Rationale* com Kuaba

De acordo com o metamodelo da UML, o primeiro problema a ser resolvido no design de um diagrama de classes é a identificação dos elementos do modelo. Assim, é feita a instanciação do elemento *Questão* da ontologia com o valor "*What are the model elements?*" (representada com retângulo), que é respondida pelas instâncias do elemento *Ideia* (representadas com elipses) com os itens de informação "*Pedido*", "*Cliente*" e "*Nome*" fornecidos pelos desenvolvedores. Elas também poderiam ser obtidas de uma ontologia de domínio, ou extraídas a partir do *design rationale* de uma fase anterior do processo de software, elicitação de requisitos, que não é abordada neste artigo.

Uma vez que as ideias de domínio são fornecidas, é necessário definir como cada uma delas será modelada usando a linguagem UML. Isto é representado na Figura 2 pela sugestão das questões "*How to model ...?*" para cada ideia de domínio. As possíveis ideias de design que respondem essas questões são instanciadas a partir do metamodelo UML, ou seja, um elemento em um diagrama de classes pode ser, entre outras opções de design, uma classe, um atributo, ou uma associação. Por questões de simplicidade são mostradas no exemplo apenas as opções classe e atributo.

Esse exemplo mostra como a semântica do metamodelo UML "dirige" a instanciação da ontologia Kuaba, fornecendo as opções de design usadas na modelagem para registrar *design rationale*. Portanto, quando um metamodelo bem definido é utilizado, é possível automatizar a instanciação de grande parte dos elementos da ontologia Kuaba usados no registro de *design rationale* (questões e ideias de design). Assim, os desenvolvedores precisam fornecer apenas as ideias de domínio e os argumentos (representados como retângulos tracejados) contra ou a favor de cada ideia.

2.1. Implementação da abordagem Kuaba

A arquitetura para implementação da abordagem Kuaba inclui três componentes principais: uma ferramenta de design, o subsistema Kuaba, e um *framework* de implementação de metamodelos [Nunes e Medeiros 2009]. A ferramenta de design é responsável por permitir a edição de modelos e pela notificação das modificações realizadas nos mesmos. Essa notificação é feita ao subsistema Kuaba por meio de eventos. O subsistema Kuaba então identifica o tipo de evento ocorrido na ferramenta e instancia os elementos corretos da ontologia Kuaba com as opções de design do metamodelo usadas no modelo, solicitando argumentos quando necessário. O *framework* de implementação de metamodelos fornece toda a infraestrutura para gerenciar a criação, o armazenamento, o acesso, a descoberta e as consultas de metadados baseados na especificação MOF (*Meta-Object Facility*) [OMG 2016].

O subsistema Kuaba é o responsável pela implementação da abordagem Kuaba. Nele se encontram, entre outros, os módulos para a manipulação da ontologia Kuaba (*kuabaModel*), para captura e representação semiautomática de *design rationale (mofParser)* e para consulta e edição de *rationale* (*repository*). A arquitetura também inclui um módulo que deve ser implementado na ferramenta de design chamado *designToolAdapter*, que é responsável pelo recebimento dos eventos gerados em sua área de desenho. Isto torna o subsistema Kuaba independente de ferramenta de design. Essa arquitetura foi utilizada no desenvolvimento da ferramenta KSE (*Kuaba Software Engineering*) [Nunes e Medeiros 2009] na qual o subsistema Kuaba foi integrado à ferramenta de design ArgoUML (https://argouml.softonic.com.br/).

3. Desafios Técnicos em MDE e Kuaba

A abordagem Kuaba permite registrar as diversas alternativas de solução consideradas durante a modelagem e os argumentos (opiniões) contra ou a favor dessas alternativas. As decisões tomadas sobre a aceitação ou rejeição dessas alternativas também são registradas e utilizadas na geração de soluções de design. As soluções aceitas representam o modelo final produzido (artefato) que, pelo *design rationale*, é automaticamente vinculado ao registro da deliberação que lhe deu origem. Portanto, a abordagem Kuaba pode viabilizar o uso de modelos de discussão (deliberação) em ferramentas MDE, permitindo conectá-los aos artefatos de software produzidos, sendo uma possível solução para a questão (i) apresentada por Bucchiarone et al. (2020).

Na abordagem Kuaba as escolhas são feitas a partir do registro das decisões sobre a adoção ou não das diferentes opções de design consideradas. Como tanto as soluções aceitas como as rejeitadas ficam registradas no *design rationale* e estas incluem as opções de design previstas no metamodelo utilizado, é possível gerar e visualizar diferentes modelos a partir das soluções registradas. Assim, é viável (ii)

relacionar modelos diferentes a escolhas diferentes e (iii) detectar soluções propostas para cada escolha utilizando as representações de *design rationale*.

Por fim, está prevista neste projeto de pesquisa a integração do subsistema Kuaba com a ferramenta Papyrus (https://www.eclipse.org/papyrus/). O objetivo é utilizar *design rationale* na execução de modelos e contribuir para as pesquisas relacionadas à questão (iv) alavancar / integrar ferramentas de modelagem flexíveis.

4. Considerações finais

Embora a MDE tenha evoluído bastante, a falta de ferramentas tem sido uma dificuldade para sua utilização efetiva pelos desenvolvedores de software. Este projeto visa fornecer uma ferramenta integrada que implemente soluções para os desafios em MDE com Kuaba e apoie a simulação e o reúso de modelos utilizando *design rationale*.

O projeto também inclui o estudo de soluções utilizando *design rationale* para problemas de modelagem que segundo Whittle et al. (2017) ainda são relativamente inexplorados em abordagens MDE. Entre eles, o pouco suporte ao processo de resolução de problemas em ferramentas MDE; o suporte praticamente inexistente a estágios iniciais de projeto e à criatividade na modelagem; e a falta de suporte em ferramentas para apoiar modelagem colaborativa. O modelo de representação baseado em argumentação utilizado pela abordagem Kuaba pode ser adaptado para apoiar a resolução de problemas. A captura de *design rationale* pode ser aprimorada para dar suporte à interação entre os *stakeholders* em um projeto e a tomada de decisão sobre os requisitos do sistema a ser desenvolvido. A integração da ferramenta Papyrus com o subsistema Kuaba e o módulo *ModelBus* (https://www.modelbus.org/en/papyrus.html) está em estudo para fornecer suporte à modelagem de software colaborativa e ao registro de *design rationale* como resultado da colaboração entre os desenvolvedores.

Referências

Bucchiarone, A., Cabot, J., Paige, R. F., Pierantonio, A. (2020) “Grand Challenges in model-driven engineering: an analysis of the state of the research”, Software and Systems Modeling 19:5-13.

Medeiros, A. P. e Schwabe, D. (2008) “Kuaba approach: Integrating formal semantics and design rationale representation to support design reuse”, Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 22:399-419.

Nunes, T. R. e Medeiros, A. P. (2009) “KSE - ferramenta de apoio à captura e representação semi-automática de design rationale em Engenharia de Software”. XXIII Simpósio Brasileiro de Engenharia de Software (Ferramentas). Fortaleza, CE.

OMG (2016) “Meta Object Facility (MOF) Specification”, https://www.omg.org/mof/.

Schmidt, D. C. (2006) “Model-driven engineering”, IEEE Computer, 39 (2), p. 25-31.

Whittle, J., Hutchinson, J., Rouncefield, M. (2014) “The State of Practice in Model-Driven Engineering”, IEEE Software, May/June 2014, p.79-85.

Whittle, J.; Hutchinson, J.; Rouncefield, M.; Burden, H.; Heldal, R. (2017) “A taxonomy of tool-related issues affecting the adoption of model-driven engineering”, Software and System Modeling, 16:313–331.

W3C (2012), “Web Ontology Language (OWL)”, https://www.w3.org/OWL/.