

Usando Modelos de Requisitos em Tempo de Execução: Potencial e Desafios

Vítor E. Silva Souza and Renata Guizzardi

Núcleo de Estudos em Modelagem Conceitual e Ontologias (Nemo)
Departamento de Informática, Universidade Federal do Espírito Santo (Ufes)
Vitória, ES, Brasil
{vitorsouza,rguizzardi}@inf.ufes.br

Resumo Pesquisadores têm cada vez mais direcionado sua atenção ao uso de modelos em tempo de execução (*runtime*), provendo ferramentas e *frameworks* que auxiliam os desenvolvedores na tarefa de construir software alinhado a seus requisitos/arquitetura. Em particular, algumas pesquisas em Engenharia de Requisitos concentraram-se em desenvolver sistemas de software que possuam a capacidade de ler seus próprios modelos de requisitos e tomar decisões a partir de uma análise do mesmo. É uma tendência comum, por exemplo, na área de sistemas adaptativos, para a qual contribuímos recentemente. Neste artigo, propomos um novo projeto de pesquisa sobre o uso de modelos de requisitos em tempo de execução, discutindo seus potenciais benefícios e os desafios envolvidos.

Palavras-chave: requisitos, objetivos, modelos, tempo de execução, sistemas colaborativos de gestão de conhecimento

1 Introdução

Na área de modelagem conceitual, um assunto que vem atraindo a atenção de pesquisadores é o uso de modelos em tempo de execução. Esta tendência pode ser observada em artigos que estabelecem agendas de pesquisa e na criação de *workshops* específicos sobre o tema.¹ No contexto da Engenharia de Requisitos, por exemplo, a ideia é fazer com que o sistema de software leia seu próprio modelo de requisitos e tome decisões em tempo de execução baseado nele.

Um dos autores deste artigo propôs recentemente uma abordagem baseada em requisitos para o desenvolvimento de sistemas adaptativos [4]. A proposta, batizada de *Zanshin*, inclui também um *framework* que utiliza modelos em tempo de execução como explicado acima: em tempo de execução, indicadores específicos do sistema são monitorados e, caso algum destes apresente problemas, o sistema é instruído sobre o que deve ser feito para adaptar-se àquela situação. Em outras palavras, o *framework* utiliza os modelos em *runtime* para implementar um *feedback loop* que provê habilidades de auto-adaptação ao sistema.

Para representar requisitos, utilizamos modelos que incluem objetivos (*hard* e *soft*), tarefas, suposições de domínio (o que se assume ser verdadeiro) e restrições

¹ Veja, por exemplo, <http://www.comp.lancs.ac.uk/~bencomo/WorkshopMRT.html>.

de qualidade (critérios precisos para satisfação de objetivos *soft*). A abordagem *Zanshin* complementa estes modelos com requisitos do *feedback loop*, a saber: “Requisitos de Percepção” (*Awareness Requirements* ou *AwReqs*, indicam o que monitorar) [6] e “Requisitos de Evolução” (*Evolution Requirements* ou *EvoReqs*, especificam o que fazer para adaptar-se) [5].

Nosso objetivo é focar em domínios de aplicação que possam se beneficiar do uso de modelos em tempo de execução. Dentre eles, destacamos a área de Gestão Colaborativa do Conhecimento (GCC). Sistemas desenvolvidos para dar suporte a GCC geralmente são complexos, pois devem permitir que usuários de diferentes perfis e níveis de proficiência interajam, ao mesmo tempo gerenciando os itens de conhecimento pessoais de cada um destes usuários e/ou comunidades de usuários. Portanto, a capacidade de ajustar os requisitos em tempo de execução daria mais flexibilidade a sistemas deste tipo, tornando-os mais capazes de lidar com a dinâmica do ambiente de compartilhamento de conhecimento colaborativo.

Este projeto tem como meta o desenvolvimento de uma metodologia (possivelmente baseada em i^*), seguindo a linha de Souza [4] para desenvolver sistemas que utilizem modelos de requisitos em tempo de execução. Como campo de experimentação para as futuras propostas deste projeto, analisaremos trabalhos anteriores recentes na área de sistemas de GCC, procurando entender que tipo de suporte metodológico devemos desenvolver para novas versões do sistema.

Especificamente, com relação à metodologia existente, nosso objetivo é sanar algumas das limitações existentes na abordagem *Zanshin*; além de estudar de forma mais aprofundada a semântica dos diferentes elementos sintáticos utilizados em nossos modelos de objetivos nas diferentes fases do processo de software, com possível aplicação não só para sistemas adaptativos mas, em geral, para modelar sistemas que utilizam requisitos em tempo de execução.

Neste artigo, descrevemos nossa agenda de pesquisa neste tópico para o futuro próximo, apresentando potenciais benefícios do uso de modelos de requisitos em *runtime* e discutindo os desafios associados. A Seção 2 apresenta os objetivos da pesquisa. Em seguida, a Seção 3 discute as possíveis contribuições e desafios envolvidos nesta pesquisa. Finalmente, a Seção 4 apresenta as conclusões.

2 Objetivos de Pesquisa

Podemos definir o seguinte objetivo de pesquisa para nossa proposta: *Definir uma linguagem bem-fundamentada para a criação de modelos de requisitos para sistemas de software que possam ser utilizados em tempo de execução pelo próprio sistema, dando suporte à algum tipo de processo de tomada de decisão e propor um processo sistemático para a criação destes modelos em diferentes fases do processo de software*. A partir do objetivo acima, derivamos questões de pesquisa mais específicas:

Q1: *Qual é o estado-da-arte do uso de modelos de requisitos em runtime?*

Estamos interessados em conhecer os tipos de modelos utilizados para representar requisitos (sintaxe e semântica), em que passos do processo eles

são utilizados e que tipo de análise e tomada de decisão é feita em tempo de execução com base nestes modelos. Algumas destas propostas podem ser utilizadas como base para as propostas seguintes desta pesquisa.

Q2: *Qual o significado por trás dos elementos de um modelo de requisitos (orientado a objetivos)?*

Queremos utilizar análise ontológica para definir um meta-modelo que possa ser utilizado como base para construção de diferentes modelos de requisitos, ou seja, em diferentes pontos do processo (análise inicial dos requisitos, especificação detalhada, projeto, etc.) até seu uso em tempo de execução. Nosso foco é nas abordagens orientadas a objetivos.

Q3: *Que tipo de suporte metodológico pode ser oferecido para engenheiros de requisitos e desenvolvedores de software em geral usando estes modelos?*

Um dos objetivos desta pesquisa é também a proposta de alguma forma de suporte metodológico, o que pode variar de simples recomendações até um completo processo sistemático para a criação dos diferentes modelos.

Q4: *Que tipo de ferramenta poderia ser oferecida para dar suporte a engenheiros de requisitos e desenvolvedores de software em geral?*

Além de suporte metodológico, ferramentas que provejam suporte para criação, derivação, verificação, etc. de modelos (ferramentas CASE) são importantes para a adoção da proposta na prática. *Frameworks* que utilizam os modelos em tempo de execução para auxiliar sistemas-base na tomada de decisão baseada em modelos são também necessários para experimentar a proposta (vide **Q5**).

Possíveis domínios para o desenvolvimento de tais *frameworks* são sistemas adaptativos [4], interoperabilidade na Web Semântica [3] ou Sistemas Colaborativos de Gestão do Conhecimento [3].

Q5: *A proposta é factível e útil na prática?*

Por fim, porém não menos importante, todas as ideias propostas devem ser validadas por meio de experimentos e estudos de caso que consistam na elicitação de requisitos e no desenvolvimento de sistemas (ou simulação dos mesmos), executando em conjunto com os *frameworks* desenvolvidos para análise dos modelos em tempo de execução.

Experimentos já desenvolvidos anteriormente podem auxiliar nesta tarefa: o clássico exemplo do sistema de agendamento de reuniões [5], o sistema de despacho de ambulâncias [6,5], o caixa eletrônico [8], o sistema de informação acadêmico [3], etc.

3 Contribuições e Desafios

Dadas as questões apresentadas na Seção 2, nosso plano de trabalho começa com a revisão bibliográfica (fase na qual nos encontramos atualmente) sobre o uso

de modelos em tempo de execução (**Q1**), seguida pela análise ontológica destes modelos e proposta de um meta-modelo para modelos de requisitos (**Q2**). Em seguida, atividades relacionadas à elaboração da metodologia (**Q3**), desenvolvimento de ferramentas (**Q4**) e experimentos (**Q5**) serão conduzidas de forma iterativa, em paralelo. desta maneira, esperamos utilizar partes da metodologia que forem sendo definidas incrementalmente para descobrir, por meio dos experimentos, que tipo de ajuste se faz necessário.

O restante desta seção descreve, em quatro partes, as contribuições e desafios relacionadas às questões de pesquisa **Q2** a **Q5**.

3.1 Uma Linguagem Bem-Fundamentada para Uso de Requisitos em Tempo de Execução (**Q2**)

De acordo com Franch et al. [2], o uso do *framework i** na prática é afetado pela falta de uma semântica uniforme que descreva os conceitos da linguagem. Isto pode ser explicado por sua popularidade entre diferentes grupos de pesquisa e a diversidade de sua aplicação, o que acaba gerando dialetos distintos. Já há algum tempo, nosso grupo de pesquisa (Nemo) está envolvido em pesquisas que buscam um modelo semântico comum para os conceitos do núcleo da linguagem, utilizando ontologias de fundamentação para justificar as escolhas feitas.

Recentemente chegamos à conclusão que a linguagem utilizada em [4] também sofre de problemas similares. Isso abre, portanto, uma oportunidade de aproveitar o trabalho feito no Nemo para propor correções nesta linguagem, de modo que se torne uma linguagem de modelagem bem-fundamentada, possivelmente com impacto positivo em sua usabilidade, adoção na prática, etc.

Esperamos também investigar trabalhos relacionadas, como as ontologias existentes para engenharia de requisitos (ex.: a que foi utilizada em *Zanshin*), o que pode ser útil para a proposta de um meta-modelo bem-fundamentado desta linguagem. O desafio aqui é a proposta de um meta-modelo que seja universalmente aceito por diferentes pesquisadores e profissionais da área.

3.2 Desenvolvimento da Metodologia (**Q3**)

Desenvolver uma metodologia que dê suporte ao uso de requisitos em tempo de execução poderá auxiliar profissionais a construir modelos de boa qualidade. Automatizar certas partes do processo também os libera de tarefas repetitivas. Um dos desafios é dar suporte ao usuário sem, no entanto, restringi-lo em excesso.

Revisar a literatura de áreas relacionadas também pode ser útil nesta questão para auxiliar no desenvolvimento metodológico. Por exemplo, Engenharia Orientada a Modelos (*Model-Driven Engineering*, ou MDE) pode auxiliar no desenvolvimento de ferramentas que efetuem transformação de modelos (como foi feito, por exemplo, em [7]). Naturalmente, esta tarefa está também relacionada ao desenvolvimento de ferramentas (**Q4**).

Esperamos, também, utilizar os dois sistemas de GCC (apresentados na próxima subseção) como base de testes sobre a qual poderemos validar nossas

hipóteses e intuições, possivelmente promovendo novas ideias para a metodologia em desenvolvimento. A esse respeito, apesar de focarem na mesma área, tais sistemas são bem diferentes em estrutura e funcionalidade, tendo cada um deles seus desafios em particular. A próxima subseção discute estes desafios de forma mais aprofundada.

3.3 Desenvolvimento de Ferramentas (Q4)

O desenvolvimento de ferramentas pode ser dividido em duas partes: (a) desenvolver uma ferramenta CASE que auxilie desenvolvedores na tarefa de seguir a metodologia proposta de forma a criar modelos corretos e úteis; e (b) desenvolver um *framework* que use estes modelos em tempo de execução para efetuar algum tipo de tomada de decisão. Como mencionamos anteriormente, sistemas adaptativos [4] ou Sistemas GCC são domínios nos quais temos alguma experiência, o que nos motiva a construir ferramentas para estes domínios.

Sistemas GCC auxiliam usuários na obtenção e no compartilhamento de seus itens de conhecimento pessoais, enquanto provêem também funcionalidades de interação social. Suas capacidades são alinhadas com a Gestão Construtivista de Conhecimento [1], que defende que mais atenção e cuidado deve ser dada à fonte do conhecimento em práticas e sistemas GC. Tanto Biblioref [3] quanto Trama² são exemplos de sistema GCC.

Biblioref é um sistema GCC que dá suporte ao compartilhamento de documentos. Neste sistema, classificação e compartilhamento de conhecimento é baseada no desenvolvimento de taxonomias individuais de cada usuário, que são mapeadas em uma taxonomia de referência. Os principais benefícios deste sistema são: (a) promover colaboração entre usuários através do acesso mútuo aos seus documentos; (b) dar autonomia ao usuário na organização do conhecimento; e (c) prover mecanismos que relacionam os diferentes esquemas de classificação dos usuários, permitindo aos mesmos localizar potenciais colaboradores a partir de inferências em cima destas relações.

Trama foi desenvolvido para dar suporte à colaboração entre os membros do Laboratório de Tecnologias de Apoio a Redes de Colaboração. Ele é baseado em um sistema de gestão de conteúdo e provê ferramentas para armazenamento de artefatos de conhecimento (ex.: relatórios, atas de reunião, artigos de pesquisa, etc.), além de ferramentas de colaboração como fóruns, bate-papo e calendário.

Na área de sistemas adaptativos, iremos focar na melhoria do *framework Zanshin*, uma vez que modelos bem-fundamentados sejam desenvolvidos utilizando a abordagem homônima (vide Seção 3.1).

3.4 Validação por Meio de Experimentos (Q5)

Um aspecto importante de qualquer proposta é a validação. Métodos de validação abrangem desde descrições simples de cenários para motivação da pesquisa até estudos de caso completos conduzidos em empresas parceiras, pesquisas

² http://labtar.ufes.br/index.php?option=com_content&view=article&id=104.

com profissionais da área, etc. Conduzir experimentos em áreas que envolvem muita participação humana, como a Engenharia de Requisitos e a GCC, é em geral um grande desafio.

Como contribuições esperamos entregar resultados empíricos que falem sobre a utilidade, factibilidade e escalabilidade de nossas propostas.

4 Conclusões

Neste artigo, apresentamos os objetivos de pesquisa e rascunhamos um plano de trabalho para o desenvolvimento de uma linguagem bem-fundamentada para modelos de requisitos que possam ser utilizados em *runtime*, além de um processo para sua criação na prática. Esta pesquisa foca na Engenharia de Requisitos Orientada a Objetivos, com particular interesse em i^* e trabalhos relacionados.

Pretendemos seguir esta agenda de pesquisa nos próximos anos e esperamos contribuir com resultados às comunidades de pesquisa em Engenharia de Requisitos, Modelagem Conceitual e Gestão do Conhecimento.

Agradecimentos

Este trabalho foi parcialmente apoiado pela FAPES (<http://www.fapes.es.gov.br>) por meio da bolsa PRONEX #52272362.

Referências

1. Guizzardi, R.S.S.: Agent-oriented Constructivist Knowledge Management. Phd thesis, University of Twente, The Netherlands (2006)
2. Guizzardi, R.S.S., Franch, X., Guizzardi, G.: Applying a foundational ontology to analyze means-end links in the i^* framework. In: Proc. of the 6th International Conference on Research Challenges in Information Science. pp. 1–11. IEEE (2012)
3. Manola, R., Guizzardi, R.S.S., Gomes, R.L.: Biblioref: a semantic bibliographic reference management system. In: Companion Proc. of the 14th Brazilian Symposium on Multimedia and the Web. pp. 149–151. ACM (2008)
4. Souza, V.E.S.: Requirements-based Software System Adaptation. Phd thesis, University of Trento, Italy (2012)
5. Souza, V.E.S., Lapouchnian, A., Angelopoulos, K., Mylopoulos, J.: Requirements-driven software evolution. Computer Science - Research and Development pp. 1–19 (2012)
6. Souza, V.E.S., Lapouchnian, A., Robinson, W.N., Mylopoulos, J.: Awareness Requirements. In: Lemos, R., Giese, H., Müller, H.A., Shaw, M. (eds.) Software Engineering for Self-Adaptive Systems II, Lecture Notes in Computer Science, vol. 7475, pp. 133–161. Springer (2013)
7. Souza, V.E.S., Mazón, J.N., Garrigós, I., Trujillo, J., Mylopoulos, J.: Monitoring Strategic Goals in Data Warehouses with Awareness Requirements. In: Proc. of the 2012 ACM Symposium on Applied Computing. pp. 1075–1082. ACM (2012)
8. Tallabaci, G., Souza, V.E.S.: Engineering Adaptation with Zanshin: an Experience Report. In: Proc. of the 8th International Symposium on Software Engineering for Adaptive and Self-Managing Systems (to appear) (2013)