

Subprojeto de Iniciação Científica

Edital:	Edital PIIC 2020/2021
Título do Projeto:	Evolução de Software baseada em Requisitos e Ontologias
Título do Subprojeto:	Desenvolvimento de Ferramenta de Conversão de Modelos de Requisitos Baseados em Objetivos
Candidato a Orientador:	Vítor Estêvão Silva Souza
Candidato a Bolsista:	(A definir)
Membros Equipe do Projeto:	César Henrique Bernabé (Doutorado)

1 Resumo

No contexto de GORE (*Goal-Oriented Requirements Engineering* — Engenharia de Requisitos Orientada a Objetivos), muitas abordagens e ferramentas já foram propostas, no entanto faltam um arcabouço semântico para lidar com problemas conceituais, como, por exemplo, sobrecarga, excesso, redundância e incompletude de construtos, problemas esses que podem levar à não compreensão e mal uso das linguagem GORE, além de comprometer sua expressividade. Assim, o projeto de pesquisa no qual este subprojeto está inserido tem por objetivo utilizar ontologias como ferramentas conceituais no domínio de GORE. Um de seus resultados em particular foi a *Goal-Oriented Requirements Ontology* (GORO), uma ontologia sobre o domínio de GORE, que serviu de base para a construção de um protótipo de ferramenta de conversão de modelos GORE chamado *GORO Conversion Tool* (GCT). Este subprojeto tem por objetivo a evolução desta ferramenta, por meio da utilização de técnicas de Desenvolvimento Orientado a Modelos (*Model-Driven Development* ou MDD).

Palavras-chave: Requisitos. Objetivos. Ontologia. Ferramenta. MDD.

2 Introdução

No contexto da Engenharia de Software, há décadas existem esforços voltados especificamente para a área de Engenharia de Requisitos — *Requirements Engineering* ou RE (Ross & Schoman Jr., 1977; Mylopoulos et al., 1999). Também há décadas, as pesquisas em RE reconhecem o importante papel que objetivos (*goals*) desempenham no processo de RE (van Lamsweerde, 2001). Objetivos proveem uma série de vantagens quando comparados com os conceitos utilizados na Engenharia de Requisitos tradicional, como, por exemplo: proveem critérios de completude para

especificações de requisitos (relativos aos objetivos dos *stakeholders*¹); proveem justificativa de pertinência (*rationale*) para cada requisito do sistema; possuem maior legibilidade ao se estruturar documentos de requisitos complexos; dão assistência à exploração de alternativas, detecção e resolução de conflitos, etc. (van Lamsweerde, 2001).

Ao longo dos anos, muitas abordagens de Engenharia de Requisitos Orientada a Objetivos (*Goal-Oriented Requirements Engineering* ou GORE) foram propostas. O NFR Framework (Chung et al., 1999) promove a análise de diferentes soluções de acordo com requisitos não-funcionais. KAOS (Dardenne et al., 1993) provê um modelo conceitual, uma linguagem de modelagem associada e um conjunto de estratégias para aquisição de requisitos baseada em objetivos. GBRAM (Antón, 1996) oferece técnicas para análise, elaboração e refinamento de objetivos. *i** (Yu et al., 2011) — e sua versão mais recente, iStar 2.0 (Dalpiaz et al., 2016) — introduz aspectos de modelagem e raciocínio social aos modelos de requisitos, por meio de conceitos como agentes e dependências. Baseados em *i**, a metodologia Tropos (Bresciani et al., 2004) expande os conceitos para outras fases do processo de desenvolvimento de software, com foco em agentes de software, enquanto a linguagem GRL,² parte da recomendação ITU-T *User Requirements Notation* (URN),³ combina a modelagem social e intencional de *i** com a abordagem orientada a cenários Use Case Maps (UCM) (Liu & Yu, 2004). Por fim, Techne (Jureta et al., 2010) se propõe a ser uma linguagem GORE abstrata para servir de base formal para novas linguagens de modelagem voltadas às fases iniciais da Engenharia de Requisitos.

Dada a existência de inúmeras abordagens para representação de modelos de requisitos e o fato de representarmos requisitos desde as fases iniciais do levantamento até a execução do software, incorremos, inevitavelmente, em problemas de modelagem conceitual como sobrecarga, excesso, redundância e incompletude de construtos (Guizzardi, 2005). Como exemplo de sobrecarga, o construto *requisito* pode significar uma intenção de um *stakeholder* em uma fase inicial de levantamento, um artefato documental que representa uma possível solução em uma fase subsequente, ou até mesmo um artefato de código em tempo de execução que permita a um sistema raciocinar em cima de seu modelo de requisitos, como é o caso, e.g., das propostas para sistemas adaptativos baseadas em requisitos (Souza, 2012).

No projeto de pesquisa no qual este subprojeto está inserido, utilizamos ontologias como ferramentas conceituais para tratar estes e outros problemas de modelagem conceitual no domínio de GORE. Uma ontologia como artefato computacional, i.e., uma especificação explícita e formal de uma conceituação compartilhada (Studer et al., 1998), pode auxiliar na mitigação destes problemas e já vem atraindo interesse na comunidade de RE (Dermeval et al., 2016). Ao explicitar e definir formalmente os conceitos, relações e restrições em um domínio de interesse,

¹*Stakeholder*, traduzido em algumas obras como “envolvido”, é qualquer um que tenha interesse direto ou que se beneficie de um software a ser desenvolvido (Pressman & Maxim, 2016).

²<http://www.cs.toronto.edu/km/GRL/>.

³<http://www.itu.int/rec/T-REC-Z.151/en>.

uma ontologia pode ser usada para facilitar a integração de dados, modelos e sistemas desenvolvidos de forma independente, para apoiar a comunicação entre pessoas trabalhando nesse universo de discurso e como uma especificação reutilizável para a construção de sistemas no domínio (Jasper et al., 1999). Em outras palavras, ontologias são amplamente aceitas na literatura como ferramentas para prover semântica e interoperabilidade entre entidades com naturezas distintas (Guarino et al., 2009; Guizzardi, 2007).

Neste contexto, um dos resultados deste projeto é a *Goal-Oriented Requirements Ontology*, ou GORO (Bernabé et al., 2019; Bernabé, 2020), uma ontologia sobre o domínio de GORE. A partir da ontologia, e como prova de conceito e aplicação prática da mesma, foi implementado um protótipo de ferramenta de conversão de modelos GORE, chamado *GORO Conversion Tool* (GCT).⁴ Utilizando a ontologia como interlingua, o GCT converte modelos iStar produzidos na ferramenta *piStar*⁵ para modelos KAOS que podem ser carregados na ferramenta *Objectiver*.⁶

Este subprojeto tem por objetivo evoluir o protótipo da GCT de modo que se torne uma ferramenta mais ampla — i.e., dê suporte a outras possibilidades de conversão de modelos além de iStar → KAOS — e amigável — i.e., seja fácil de instalar e usar, facilitando sua adoção pela academia e pela indústria.

3 Objetivos

O objetivo geral desse trabalho é evoluir o protótipo da GCT (*GORO Conversion Tool*) de modo a dar suporte a uma gama maior de possibilidades de conversão de modelos, bem como torná-la uma ferramenta fácil de instalar e usar, tanto para acadêmicos quanto para profissionais da área de Engenharia de Requisitos. São objetivos específicos deste trabalho:

- Aprender sobre GORE — *Goal-Oriented Requirements Engineering* ou Engenharia de Requisitos Orientada a Objetivos — entendendo seus princípios (van Lamsweerde, 2001), conhecendo as principais linguagens (Chung et al., 1999; Dardenne et al., 1993; Dalpiaz et al., 2016; Jureta et al., 2010) e verificando como a GORO (Bernabé et al., 2019; Bernabé, 2020) representa este domínio do conhecimento;
- Aprender sobre MDD — *Model-Driven Development* ou Desenvolvimento Orientado a Modelos (Brambilla et al., 2017; Pastor et al., 2008) — e as ferramentas associadas a esta abordagem como, por exemplo, a plataforma Eclipse RCP (*Rich Client Platform*) e seus *plug-ins* relevantes como, e.g., EMF,⁷ XText,⁸ Sirius,⁹ etc.;

⁴<https://github.com/nemo-ufes/gorotool>.

⁵<https://www.cin.ufpe.br/~jhcp/pistar/>

⁶<http://www.objectiver.com/>

⁷<https://eclipse.org/modeling/emf/>.

⁸<https://eclipse.org/Xtext/documentation/>.

⁹<https://eclipse.org/sirius/overview.html>.

- Conhecer a *GORO Conversion Tool* (GCT) e utilizar o conhecimento adquirido para torná-la uma ferramenta amigável e com uma gama maior de possibilidades de conversão;
- Propor uma linguagem simples, textual, que utilize a GORO como base para representar modelos GORE e que possa ser convertida para modelos das principais linguagens de modelagem existentes, de modo a permitir aos usuários da ferramenta escrever modelos com base na GORO e representá-los em suas linguagens GORE de preferência.

No contexto do projeto de pesquisa “Evolução de Software baseada em Requisitos e Ontologias”, este subprojeto visa contribuir com o objetivo “O1. Realizar análise ontológica das principais linguagens de modelagem de requisitos orientadas a objetivos, visando propor uma linguagem unificada, ontologicamente bem fundamentada, para Engenharia de Requisitos Orientada a Objetivos”. Em particular, contribui especificamente para os resultados “R1.3. Proposta de linguagem GORE unificada, com base no arcabouço semântico provido pelos resultados anteriores” e “R1.4. Ferramenta de interoperabilidade entre diferentes linguagens GORE, com base nos resultados anteriores”.

4 Metodologia

Este subprojeto será feito no contexto de um trabalho de pós-graduação, envolvendo, portanto, além do(a) bolsista de Iniciação Científica e do professor orientador, também aluno de Doutorado César Henrique Bernabé que, além de autor da GORO (Bernabé et al., 2019; Bernabé, 2020) e da *GORO Conversion Tool* (GCT), já cursou disciplina no programa de pós-graduação sobre Desenvolvimento Orientado a Modelos (MDD) e poderia auxiliar o(a) bolsista com a plataforma Eclipse e seu sistema de *plug-ins*.

O trabalho será iniciado com a revisão bibliográfica, dividida em três partes. A primeira consistirá em leituras introdutórias sobre ontologias como ferramentas de modelagem conceitual. O(A) bolsista deve estudar o suficiente dos conceitos e técnicas que foram utilizadas na construção da GORO. A segunda inclui leitura de artigos sobre GORE, suas principais linguagens de modelagem e experimentação das ferramentas existentes. O(A) bolsista deve familiarizar-se o suficiente com este domínio de modo a, em seguida, analisar e compreender a GORO. A terceira visa estudar sobre Desenvolvimento Orientado a Modelos (MDD) para conhecer técnicas e ferramentas que possam ser utilizadas para realizar o trabalho de evolução da GCT.

Em seguida, o(a) bolsista deverá estudar o código da GCT e, tendo conhecido as possibilidades que as ferramentas MDD oferecem, utilizá-las para a evolução da GCT, de modo a satisfazer os objetivos do trabalho, ou seja: (i) implementar meios de se instalar e utilizar facilmente a ferramenta; (ii) adicionar novas possibilidades de conversão entre modelos de ferramentas associadas a outras linguagens GORE, além da conversão iStar → KAOS existente; e (iii) incluir na GCT um editor que permita que um modelo GORE seja construído dentro da mesma por meio de

uma linguagem textual simples, baseada na GORO, e seja convertido para modelos das outras linguagens já incluídas na GCT.

Além das atividades de desenvolvimento descritas acima, prevê-se também a participação do(a) bolsista na escrita de um artigo científico junto aos demais membros do projeto sobre a ferramenta desenvolvida.

5 Plano de trabalho/Cronograma

Esta seção descreve as atividades que serão desenvolvidas pelo(a) bolsista e seu cronograma de execução para que o objetivo deste subprojeto possa ser alcançado.

ATIVIDADES

Lista de atividades
1 - Estudo sobre ontologias: leitura de artigos sobre ontologias em geral e, em particular, sobre UFO (Guizzardi, 2005), ontologia de fundamentação utilizada pela GORO, por meio de estudo dirigido com o professor orientador e apoio do estudante de Doutorado para ter a base suficiente para entender bem a GORO posteriormente;
2 - Estudo de GORE: leitura de artigos sobre GORE e suas linguagens de modelagem, estudo dirigido nos mesmos moldes, para obter um entendimento do domínio e experimentação de ferramentas de modelagem ligadas às diversas linguagens GORE;
3 - Estudo sobre desenvolvimento orientado a modelos: leitura de artigos e documentação técnica sobre esta abordagem de desenvolvimento de ferramentas, com apoio do estudante de Doutorado e do professor orientador;
4 - Desenvolvimento de uma versão da <i>GORO Conversion Tool</i> (GCT) que seja fácil de instalar e utilizar para execução da conversão iStar → KAOS já implementada em seu protótipo, bem como sua atualização para a última versão da GORO (Bernabé, 2020);
5 - Inclusão na GCT de outras possibilidades de conversão entre modelos GORE construídos utilizando ferramentas associadas às principais linguagens de modelagem GORE (exploradas na Atividade 1);
6 - Proposta e implementação de uma linguagem de modelagem simples, baseada na GORO, que permita representar modelos GORE de forma textual e convertê-los para as diversas representações gráficas suportadas pelas conversões da GCT;
7 - Elaboração de relatórios e artigos científicos: escrita dos relatórios técnicos obrigatórios do PIIC e possivelmente um artigo científico relatando os resultados alcançados durante a Iniciação Científica.

CRONOGRAMA (Ago/2020 a Jul/2021)

Atividade	ago	set	out	nov	dez	jan	fev	mar	abr	mai	jun	jul
1	X	X										
2		X	X	X								
3			X	X	X							
4				X	X	X	X					
5						X	X	X	X	X		
6									X	X	X	X
7						X					X	X

Referências

-
- Antón, A. I. (1996). Goal-Based Requirements Analysis. In *Proc. of the 2nd International Conference on Requirements Engineering*, pages 136–144. IEEE.
- Bernabé, C. H. (2020). GORO: uma ontologia sobre requisitos orientados a objetivos. Technical report, Dissertação de Mestrado, Programa de Pós-Graduação em Informática. Universidade Federal do Espírito Santo, Vitória, ES, Brasil.
- Bernabé, C. H., Souza, V. E. S., Falbo, R. d. A., Guizzardi, R. S. S., & Silva, C. (2019). GORO 2.0: Evolving an Ontology for Goal-Oriented Requirements Engineering. In *Proc. of the 2019 International Workshop on Ontologies and Conceptual Modelling (Onto.com 2019)*, pages 169–179, Salvador, BA, Brazil. Springer.
- Brambilla, M., Cabot, J., & Wimmer, M. (2017). *Model-Driven Software Engineering in Practice: Second Edition*. Morgan & Claypool Publishers.
- Bresciani, P., Perini, A., Giorgini, P., Giunchiglia, F., & Mylopoulos, J. (2004). Tropos: An Agent-Oriented Software Development Methodology. *Autonomous Agents and Multi-Agent Systems*, 8(3):203–236.
- Chung, L., Nixon, B. A., Yu, E., & Mylopoulos, J. (1999). *Non-Functional Requirements in Software Engineering*. Springer.
- Dalpiaz, F., Franch, X., & Horkoff, J. (2016). iStar 2.0 Language Guide. *CoRR*.
- Dardenne, A., van Lamsweerde, A., & Fickas, S. (1993). Goal-directed Requirements Acquisition. *Science of Computer Programming*, 20(1-2):3–50.
- Dermeval, D., Vilela, J., Bittencourt, I. I., Castro, J., Isotani, S., Brito, P., & Silva, A. (2016). Applications of ontologies in requirements engineering: a systematic review of the literature. *Requirements Engineering*, 21(4):405–437.
- Guarino, N., Oberle, D., & Staab, S. (2009). What is an Ontology? In Staab, S. & Studer, R., editors, *Handbook on Ontologies*, International Handbooks on Information Systems, pages 1–17. Springer, 2 edition.

- Guizzardi, G. (2005). *Ontological Foundations for Structural Conceptual Models*. Phd thesis, University of Twente, The Netherlands.
- Guizzardi, G. (2007). On Ontology, ontologies, Conceptualizations, Modeling Languages, and (Meta)Models. In *Proc. of the 2007 Conference on Databases and Information Systems*. IOS Press.
- Jasper, R., Uschold, M., & Others (1999). A framework for understanding and classifying ontology applications. In *Proc. of the IJCAI-99 Workshop on Ontologies and Problem-Solving Methods (KRR5)*, volume 99, pages 1–12.
- Jureta, I. J., Borgida, A., Ernst, N. A., & Mylopoulos, J. (2010). Techne: Towards a New Generation of Requirements Modeling Languages with Goals, Preferences, and Inconsistency Handling. In *Proc. of the 18th IEEE International Requirements Engineering Conference*, pages 115–124. IEEE.
- Liu, L. & Yu, E. (2004). Designing information systems in social context: a goal and scenario modelling approach. *Information Systems*, 29(2):187–203.
- Mylopoulos, J., Chung, L., & Yu, E. S. K. (1999). From Object-Oriented to Goal-Oriented Requirements Analysis. *Communications of the ACM*, 42(1):31–37.
- Pastor, O., España, S., Panach, J. I., & Aquino, N. (2008). Model-driven development. *Informatik-Spektrum*, 31(5):394–407.
- Pressman, R. S. & Maxim, B. R. (2016). *Engenharia de Software: Uma Abordagem Profissional*. Bookman, 8 edition.
- Ross, D. T. & Schoman Jr., K. E. (1977). Structured Analysis for Requirements Definition. *IEEE Transactions on Software Engineering*, SE-3(1):6–15.
- Souza, V. E. S. (2012). *Requirements-based Software System Adaptation*. PhD thesis, University of Trento, Italy.
- Studer, R., Benjamins, V., & Fensel, D. (1998). Knowledge engineering: Principles and methods. *Data & Knowledge Engineering*, 25(1-2):161–197.
- van Lamsweerde, A. (2001). Goal-Oriented Requirements Engineering: A Guided Tour. In *Proc. of the 5th IEEE International Symposium on Requirements Engineering*, pages 249–262. IEEE.
- Yu, E. S. K., Giorgini, P., Maiden, N., & Mylopoulos, J. (2011). *Social Modeling for Requirements Engineering*. MIT Press, 1st edition.