

Avaliação de Ontologias de Domínio para o Desenvolvimento Distribuído de Software

Ana Raquel M. Alves¹, Israel Felipe L. A. Silva¹, Renan L. Fernandes¹, Rodrigo C. Rocha¹, Thais A. Burity¹, Ryan R. de Azevedo¹

¹Unidade Acadêmica de Garanhuns, Universidade Federal Rural de Pernambuco (UFRPE)

Av. Bom Pastor, s/n – Boa Vista – 55.292-270 – Garanhuns – PE, Brasil

{anaraqueldemorais, israelita.felipe, rlfnan326}@gmail.com,
{rodrigo, thais.burity, ryan}@uag.ufrpe.br

Resumo. *O Desenvolvimento Distribuído de Software se tornou uma opção para empresas de software explorarem os benefícios desta abordagem. No entanto, esse conceito traz consigo novos desafios para o cenário de software, como a inexistência de um modelo público formal e normalizado das informações do projeto, o que dificulta a comunicação e o entendimento dos membros e dos artefatos. Dessa forma, este trabalho propõe uma avaliação parcial das ontologias de domínio para DDS, apresentando seus pontos em comuns, entendimento e campo de atuação. Servindo para concepção de novas ontologias para esse contexto com inferência de informações concisas, precisas e não ambíguas, de forma semiautomática dentro do ambiente.*

Abstract. *Distributed Software Development became an option for software companies to explore the benefits of this approach. However, that concept brings new challenges for the software scenario; the absence of a formal public model and standardized project information is one of these, which make it difficult the communication and understanding of members and artefacts. Thus, this paper proposes a partial evaluation of domain ontologies for DDS, showing their common points, knowledge and acting field. Give the basement for designing new ontologies for this new context with concise information, accurate and non-ambiguous, so semiautomatic within the environment.*

1. Introdução

Motivadas pelas vantagens, como disponibilidade de especialistas, redução de custos, incentivos de governos locais e baixas de rotatividade de pessoal, diversas empresas de software tem adotado equipes formadas por pessoas geograficamente distribuídas, configurando o Desenvolvimento Distribuído de Software (DDS).

Essa modalidade de trabalho traz novos desafios para o cenário de desenvolvimento de Software. De acordo com [Carmel 1999] e [Komi-Sirvo and Tihinen 2005] os seguintes fatores podem levar ao fracasso de um projeto distribuído: comunicação ineficiente, falta de gerência e diferenças culturais. Nesse contexto, é possível afirmar que a inexistência de um modelo público, formal e normalizado das informações do projeto dificulta a comunicação e o entendimento dos membros e dos artefatos, o que pode ser agravado quando a cultura e demais características dos membros das equipes, como o idioma, são distintas entre si.

Para reduzir essa complexidade, o uso de ontologias pode ser útil. [Wongthongtham et al. 2006] afirma que a utilização de ontologias é um novo paradigma para Engenharia de Software e que pode ser utilizado principalmente para conceder semântica para ferramentas auxiliares, prover forte comunicação baseado no conhecimento, centralização e disponibilidade de informação e também, prover uma base de conhecimento para que agentes de software capazes de formar um ambiente automatizado possam apoiar o DDS.

Dessa maneira, uma Ontologia voltada para o DDS permite que todos os membros da equipe tenham a mesma visão e compreensão das informações compartilhadas, além de estarem todos igualmente atualizados em qualquer mudança ou alteração feita no projeto. Como é uma ferramenta que, visa resolver problemas mais específicos que surgem nesse tipo de desenvolvimento, torna-se mais eficaz que a maioria das ferramentas de bate-papo utilizadas nesse contexto. Porém, para analisar a eficácia de uma ontologia é preciso analisar para qual área de desenvolvimento ela foi criada, e quais problemas ela resolve.

O presente artigo tem como objetivo analisar ontologias de domínio para DDS, mostrando seus pontos em comuns, disponibilidade e campo de atuação, a fim de possibilitar a produção de ontologias mais refinadas no futuro, com inferência de informações concisas, precisas e não-ambíguas, de forma semi-automática dentro do ambiente de desenvolvimento.

O restante do artigo está organizado como descrito a seguir. Na seção 2 é abordado o conceito de ontologias e suas estruturas, sendo também apresentado o ambiente de desenvolvimento distribuído de software. Na seção 3 são mostradas ontologias de domínio para DDS. A seção 4 apresenta a análise das ontologias para DDS, bem como a comparação de como essas ontologias agem no processo de desenvolvimento de software. Por fim, a seção 5 apresenta as considerações finais, evidenciando as contribuições do trabalho e a perspectiva de trabalhos futuros.

2. Referencial Teórico

2.1. Ontologias

Diversas definições são dadas a fim de descrever o significado computacional para ontologias. A mais popular é: “uma especificação formal e explícita de uma conceitualização compartilhada” definição dada por [Gruber 1995], onde ser formal implica em declarativamente definida e compreensível para agentes inteligentes de software; explícita significa que os elementos e suas restrições estão claramente definidos; conceitualização trata de um modelo abstrato de uma área de conhecimento ou de um universo limitado de discurso, compartilhada, indica um conhecimento consensual, seja uma terminologia comum da área modelada.

Assim, em um nível de abstração mais alto, ontologias estabelecem uma terminologia comum e não ambígua para os diversos domínios de conhecimento. Segundo [Guizzard 2000], uma ontologia é composta de conceito, relação, função, axiomas e instâncias. Em resumo, conceito pode ser “qualquer coisa” a respeito de “algo” que será explicado. Já o tipo de interação entre conceitos de um domínio e seus atributos é chamado de relação, cujo tipo denomina-se de função.

Os axiomas modelam sentenças que são sempre verdadeiras. Por fim, as instâncias representam elementos do domínio associados a conceitos específicos.

O uso de ontologias tem se popularizado através de diversas outras subáreas da Ciência da Computação, tais como: Engenharia de Software, Inteligência Artificial, Banco de Dados e Sistemas de Informação. Um dos principais responsáveis por esse fenômeno é o criador da Web Semântica, [Berners-Lee 2001].

São várias as motivações para o desenvolvimento de uma ontologia, [Noy e McGuinness 2001] e [Freitas 2003].

- Compartilhar entendimento comum da estrutura de informação entre pessoas ou entre agentes inteligentes de software;
- Reuso de conhecimento de um domínio. Caso exista uma ontologia que modele adequadamente certo conhecimento de um domínio, ela pode ser compartilhada e usada por engenheiros e desenvolvedores de ontologias, bem como, por equipes que desenvolvem aplicações semânticas e cognitivas;
- Tornar explícitas pressuposições de um domínio. As ontologias fornecem um vocabulário para representação de conhecimento e seu uso evita interpretações ambíguas;
- Possibilidade de tradução entre diversas linguagens e formalismos de representação do conhecimento. A tradução concretiza um ideal perseguido por gerações de pesquisadores de Inteligência Artificial. Ela facilita o reuso de conhecimento, e pode vir a permitir comunicação entre agentes em formalismos diferentes, uma vez que este serviço encontra-se disponível em um número cada vez maior de formalismos de representação do conhecimento. Outra forma de alcançar esse intento são os editores de ontologias em que se pode escolher em que linguagem de representação será escrito o código gerado.
- Mapeamento entre formalismos de representação de conhecimento que, inspirado no componente de conectividade para sistemas gerenciadores de bancos de dados ODBC (Open Database Connectivity), liga dois formalismos, criando uma interface interoperável de acesso comum para eles, permitindo a um agente acessar o conhecimento de outro agente.

2.2. Desenvolvimento Distribuído de Software

O desenvolvimento de software de forma co-localizada tem se tornado cada vez mais custoso e menos competitivo para as organizações. Visando a redução de custos, melhoria da qualidade, aumento de produtividade e competitividade global, várias empresas optam por distribuir seus processos de desenvolvimento em lugares diferentes [Audy e Prikladnicki 2008]. Sendo assim, surge o Desenvolvimento Distribuído de Software (DDS), onde os envolvidos em um determinado projeto estão dispersos geograficamente.

De acordo com [Prikladnicki 2003], as vantagens que o DDS oferece são:

- Possibilidade de desenvolvimento *follow-the-sun*, que permite o aumento de produtividade e a redução dos prazos de entrega dos produtos;
- Disponibilidade de recursos globais com baixos custos e a qualquer hora;
- Disponibilidade de recursos qualificados em áreas especializadas;
- Possibilidade de formação de equipes virtuais.

Entretanto, existe uma série de desafios inerentes a este ambiente de desenvolvimento. De acordo com [Lopes 2004], os principais desafios e questões envolvidas neste ambiente são: (1) diferenças culturais; (2) dispersão geográfica; (3) coordenação e controle; (4) comunicação; e (5) espírito de equipe.

3. Ontologias de Domínio para o DDS

Ontologias de Domínio tratam uma área específica de atuação, partindo de um ponto mais genérico [Semprebom, Camada and Mendonça 2007], e possuem grande conteúdo conceitual e relações de informação dentro do domínio. Uma vez que os usuários têm papéis bem definidos em um projeto de desenvolvimento de software, torna-se muito mais fácil colher informações sobre atividades e problemas propostos pelos próprios usuários, informações estas que enriquecem o conteúdo da ontologia. Utilizar-se desse ambiente para desenvolvimento distribuído de software traz grandes vantagens como, por exemplo, informações consistentes formuladas por especialistas, dentro de seu campo de atuação.

Nesse contexto, esse trabalho utiliza como base um mapeamento sistemático realizado por [Borges 2013], onde foram identificadas 4(quatro) ontologias de domínio para o contexto de DDS. Resumidas na Tabela 1 a seguir.

Tabela 1. Ontologias de Domínio para o DDS

Ontologias	Descrição
Componentes de Software	Ontologia que descreve conceitos de componentes de software, como suas descrições, operações, atributos, restrições e relacionamentos.
OFFFLOSC	Ontologia que descreve conceitos relacionados com atores, artefatos, atividades e recursos de comunidades de desenvolvimento Open Source.
Comunidades Open Source	Ontologia que descreve conceitos e relacionamentos de atores, regras, atividades, processos, artefatos e ferramentas de comunidades de projetos Open Source.
ONTODISEN	Ontologia que descreve conceitos e elementos que são representados e compartilhados em um ambiente de DDS, como usuários, ferramentas, ambientes, atividades e processos.

Tais ontologias foram desenvolvidas especificamente para equipes distribuídas, e portanto, possuem conceitos e características para esse contexto. É possível observar que muitos outros trabalhos utilizam ontologias para resolver ou mitigar desafios em ambientes distribuídos, mas que não são específicas para esse ambiente, estando voltadas para a Engenharia de Software em geral.

4. Avaliação de Ontologias para o contexto distribuído

Esta seção apresenta uma breve análise das ontologias de domínio para DDS resumidas na Tabela 1.

A primeira trata do OntoDiSEnv1 [Chaves, Leal and Steinmacher 2011], que na verdade é uma evolução do OntoDiSEN [Chaves, Huzita, Leal and Steinmacher 2010] para ambiente DiSEN. O OntoDiSEN é um modelo baseado em contexto para obtenção de informação em um domínio, que trata e seleciona as informações colhidas de acordo com o domínio da aplicação; a definição de aplicação, isto é, o ambiente ao qual se refere; ao seu objetivo, que tem como foco estruturar o contexto de interação dos membros; e quanto aos usuários, que podem ir desde a própria ontologia até usuários do ambiente. Para tais finalidades a ontologia apresenta então um classe principal de gerenciamento de contextos, denominada *Context*.

A evolução para o OntoDiSEnv1 deu-se na questão da obtenção e classificação de informações sobre os membros do ambiente, no qual apoiado no modelo DiSENCSE (*DiSEN-Context Sensitive Environment*) [Chaves, Huzita, Leal and Steinmacher 2010], foram introduzidas mais duas superclasses para inferência de informações, *action mapping* e *amended context registration*. Este modelo visa fazer com que todos os membros do domínio fiquem cientes da influência desta sobre ações de outros usuários e influências que tais ações têm o domínio, de forma semiautomática. Assim a ontologia ficou dividida em três superclasses, *Context*, classe principal do OntoDiSEN; *Actions* e *AmendedContext*.

Outra ontologia bem semelhante ao OntoDiSEnv1 é OFLOSSC, que na questão de desenvolvimento distribuído e com base no FLOSS [Mirbel 2009] foca em comunidades de desenvolvimento de código aberto. As duas se assemelham bastante uma vez que apresentam captura de informações e obtenção de conhecimento. Apesar disso, OFLOSSC é constituído de ontologias presentes no Dhruv [Mirbel 2009], OSDO [Mirbel 2009] e SIOC[Mirbel 2009], e além disso apresenta um modelo de cooperação (O'CoP [Mirbel 2009]), com foco para as comunidades *web*, sua classificação de informações dar-se pela interação entre usuários, ou atividades, enquanto o OntoDiSEnv1 trata de ações tomadas de acordo a problemas propostos, analisando a área de atuação o OFLOSSC captura as informações pela interação entre membros de acordo a seus papéis nos projetos.

A terceira ontologia analisada foi a TeamWeaver [Maalej and Happel 2008] que é voltada para desenvolvimento de software em geral. Possui ferramentas para organizar as informações e facilitar o acesso e entendimento às mesmas. No quesito de correção de *bugs*, fornece ajuda ao mostrar *bugs* (erros) relacionados e as soluções que foram tomadas nesses casos, auxiliando assim, na resolução dos *bugs* que eventualmente aparecerem. Utiliza a experiência e ações de cada membro para analisar as áreas em que ele tem conhecimento e selecionar membros mais aptos a solucionarem problemas em cada situação em que for necessário, a experiência de um membro é formada pelos projetos que ele já trabalhou nessa ontologia, sua área de atuação, problemas que foram resolvidos por ele e a área de cada um deles, e outras ações realizadas no ambiente da ontologia. Em cada solicitação seleciona o membro mais habilitado para realizar as atividades requeridas. A ontologia já foi desenvolvida e está disponível.

Finalizando com a ontologia OSDO [Simmons and Dillon 2008], esta é voltada para desenvolvimento de software *Open Source*. Seu principal objetivo é organizar a grande quantidade de informações geradas num projeto *open source*, possui um banco de dados para todos os artefatos que forem gerados no projeto. As informações são guardadas de forma categórica ajudando no entendimento e no acesso. Também há

classificação dos membros de uma equipe, organizando assim, o que cada um está habilitado a fazer ou não.

Na Tabela 2 são apresentadas algumas das características identificadas nas ontologias, como por exemplo, se ela é destinada ao desenvolvimento *open source*, a criação de perfil para membros através da experiência destes, se era uma ferramenta dedicada à correção de *bugs* e se as mesmas também se destinam a Engenharia de Software tradicional.

Tabela 2. Características das Ontologias

Características	OntoDisenv1	OFLLOSC	TeamWeaver	OSDO
Dedicada a Desenvolvimento Open Source	x	✓	x	✓
Cria perfil do membro através da experiência	✓	✓	✓	✓
Ferramenta dedicada a correção de bugs	x	✓	✓	x
Suporte ao ES tradicional	✓	✓	✓	✓

É importante ressaltar que a avaliação e análise das ontologias foi comprometida pelo fato de que apenas uma das ontologias, a TeamWeaver estava disponível, as outras não foram encontradas por buscas no google, por buscas em repositórios específicos de ontologias e por busca nos próprios links citados nos artigos das ontologias, sendo estes endereços inexistentes.

5. Considerações Finais

No contexto da globalização, a distribuição dos processos de desenvolvimento de software tornou-se um fato cada vez mais comum. O trabalho em ambientes de DDS é muito complexo e ainda não existem práticas maduras para esse contexto. Neste sentido, a utilização de ontologias pode trazer benefícios, como compreensão compartilhada das informações, facilidade de comunicação entre equipes distribuídas e eficácia no gerenciamento das informações.

Dessa forma, esse trabalho analisou as ontologias de domínio para DDS, comparando seus métodos de tratamento de informações obtidas em ambientes de produção de softwares, propiciando futuras evoluções das ontologias estudadas ou criação de novas mais especificamente para DDS, uma vez que as ontologias atuais não dão completo suporte a esse tipo de desenvolvimento. Os métodos utilizados foram a pesquisa de disponibilidade da ontologia, as características das ontologias e as funcionalidades do ambiente ao qual as ontologias se aplicam.

A pesquisa mostrou que as ontologias abordadas nos artigos analisados não suportam por completo o Desenvolvimento Distribuído de Software, no aspecto de inferências de informações, interferência e influência nas ações dos membros dos

ambientes nos quais as ontologias estão presentes, pois cada uma se mostra eficiente em uma determinada área do domínio, mas não como um todo. Por isso a evolução ou criação de ontologias para suporte complete a DDS é de fundamental importância para a comunidade de DDS bem como a Engenharia de Software no geral.

Como trabalhos futuros, realizar análise da ontologia DKDOnto [Azevêdo and Costa 2013] que está na fase final de desenvolvimento, e é um domínio específico de ontologia para projetos DDS. Utiliza uma ferramenta chamada DKDs, que auxilia na transmissão, geração e distribuição do conhecimento. É uma ferramenta que apoia a tomada de decisão em DDS. O sistema acessa sua base de conhecimentos, e, baseado nas experiências (projetos distribuídos, suas configurações, desafios enfrentados e soluções utilizadas) o sistema sugere possíveis soluções ao usuário para superar os desafios.

Por fim, também pretende-se tentar estabelecer o contato com os pesquisadores responsáveis pelas ontologias analisadas a fim de aprofundar a análise sendo possível detalhar melhor características e funcionalidades de ambiente e realizar uma análise através de alguma ferramenta própria para validação e avaliação de ontologias.

Referências

- Audy, J. and Prikladnicki, R. (2008) *Desenvolvimento Distribuído de Software*, Editora Elsevier.
- Azevêdo, Ryan; Costa, Catarina; Duarte, Marcos; Fachine, João Paulo; Freitas, Fred; Meira, Silvio; Rocha, Rodrigo G. C.. (2013). An Ontology-based System to Support Distributed Software Development. In *The Eighth International Conference on Software Engineering Advances (ICSEA) 2013*. October 27 - November 1, 2013 Venice, Italy
- Borges, A., Rocha, R., Costa, C., Tomaz H., Soares S., Meira S. (2013) “Ontologies Supporting the Distributed Software Development: a Systematic Mapping Study”. In *Proceedings of the International Conference on Evaluation & Assessment in Software Engineering (EASE)*. Porto de Galinhas, PE, Brasil.
- Carmel, E. (1999) *Global Software Teams: Collaboration Across Borders and Time Zones*. Prentice-Hall, EUA.
- Chaves, Ana Paula; Huzita, Elisa H. M.; Leal, Gislaine; Steinmacher, Igor. “OntoDiSEN: uma ontologia para apoiar o desenvolvimento distribuído de software” (2010). Faculdade Integrado de Campo Mourão; Universidade Estadual de Maringá; Universidade Tecnológica Federal do Paraná.
- Chaves, Ana Paula; Steinmacher, Igor. Leal, Gislaine Camila Lapasini. “OntoDiSEnv1: an Ontology to Support Global Software”, (2011). Federal Technological University of Paraná – Coordination of Technology Systems for Internet; State University of Maringá – Department of Production Engineering.
- Freitas, F. (2003). Ontologias e a web semântica. In: Renata Vieira; Fernando Osório. (Org.). *Anais do XXIII Congresso da Sociedade Brasileira de Computação* Campinas: SBC, 2003. v. 8, p. 1-52.

- G. Simmons & T.S Dillon. “Semantic Web Support For Open Source Software Development” (2008). Digital Ecosyst. & Bus. Intell. Inst., Curtin Univ. of Technol., WA, Australia.
- Gruber, T. R. (1995). “Toward Principles for the Design of Ontologies used for Knowledge Sharing”. In formal Ontology in Conceptual Analysis and Knowledge Representation. Kluwer Academic Publishers.
- Guizzardi, G. (2000). “Uma abordagem metodológica de desenvolvimento para e com reuso, baseada em ontologias formais de domínio.” Dissertação de Mestrado. Universidade Federal do Espírito Santo.
- J. H. Tim Berners-Lee and O. Lassila,. (2001). “The semantic web,” Scientific American Magazine.
- Komi-Sirvo, S; Tihinen M. (2005). Lessons Learned by Participants of Distributed Software Development. Journal Knowledge and Process Management, vol. 12 nº 2 p. 108–122.
- Lopes, L. T. (2004) “Um Modelo de Processo de Engenharia de Requisitos para Ambientes de Desenvolvimento Distribuído de Software”, Dissertação de Mestrado em Ciências da Computação, Pontifícia Universidade Católica do Rio Grande do Sul, Porto Alegre.
- Maalej, W. & Happel,Hans-Jörg “A Lightweight Approach for Knowledge Sharing in Distributed Software Teams” (2008). Technische Universität München, Munich, Germany. FZI Research Center for Information Technologies, Karlsruhe, Germany.
- Mirbel, Isabelle. “Oflossc, An Ontology For Supporting Open Source Development Communities” (2009). INRIA Sophia Antipolis, 2004 route des lucioles - BP 93, FR06902 Sophia Antipolis, Cedex France Laboratoire I3S, Route des Lucioles, BP 121, FR-06903 Sophia Antipolis, Cedex France.
- N. F. Noy and D. L. McGuinness. (2001.). “Ontology development 101: A guide to creating your first ontology.”. Online, 2001. [Online]. Available: <http://www.ksl.stanford.edu/people/dlm/papers/ontology101/ontology101-noymcguinness.html>
- Prikladnicki, R. (2003) “MuNDDoS: Um Modelo de Referência para Desenvolvimento Distribuído de Software”, Dissertação de Mestrado, Pontifícia Universidade Católica do Rio Grande do Sul, Brasil, 2003.
- Semprebom, Tiago; Camada, Marcos Yuzuru; Mendonça, Igor. “ONTOLOGIAS E PROTÉGÉ” (2007). Universidade Federal de Santa Catarina (UFSC) - Florianópolis, SC, Brazil.
- Wongthongtham, P., Chang, E., Dillon, T. S., Sommerville, I. (2006). Ontology-based Multi-site Software Development Methodology and Tools. J. of Systems Architecture. 640–653.