

A Utilização de Ontologias no Desenvolvimento Distribuído de Software: uma Revisão Sistemática da Literatura

Alex Nery Borges Júnior¹, Alexandre de Vasconcelos¹, Rodrigo Rocha¹, Silvio Meira¹

¹Centro de Informática – Universidade Federal de Pernambuco (UFPE)
Caixa Postal 7.851 – 50.732-970 – Recife – PE – Brasil

{anbj, almv, rgcr, srlm}@cin.ufpe.br

Abstract. *Motivated by factors such as business globalization and the increasing importance of software, arises the Distributed Software Development (DSD). This new approach to software development has brought several competitive, but also brought new challenges. This increases the demand for new technologies, models and techniques to assist the process of distributed software development. In this context, the use of ontologies in this environment can bring benefits as the representation of information with clarity and without ambiguity, also provides flexibility and portability of knowledge representation. This article presents a systematic review of the literature in DSD, whose goal is to identify the models, techniques, tools and best practices that use ontologies in DSD.*

Resumo. *Motivado por fatores como a globalização de negócios e o aumento da importância do software, surge o Desenvolvimento Distribuído de Software (DDS). Esta nova abordagem de desenvolvimento de software trouxe várias vantagens competitivas, mas também trouxe novos desafios. Por consequência, cresce a demanda por novas tecnologias, modelos e técnicas para auxiliar o processo de desenvolvimento distribuído de software. Neste contexto, a utilização de ontologias neste ambiente pode trazer benefícios como a representação de informações com clareza e sem ambigüidade, além de provê flexibilidade e portabilidade da representação de conhecimento. Este artigo apresenta uma revisão sistemática da literatura na área de DDS, cujo objetivo é identificar quais são os modelos, técnicas, ferramentas e boas práticas que utilizam ontologias no DDS.*

1. Introdução

Nas últimas décadas, pode ser observado um grande avanço do mercado global de desenvolvimento de software. Isso se deve ao fato do software ter assumido um papel de maior importância dentro das empresas, passando a ser componente estratégico para o sucesso das organizações [HERBSLEB 2001].

Como reflexo da globalização dos negócios, do crescimento da economia e dos avanços das tecnologias da informação e comunicação, ocorreu uma migração do mercado local para o mercado global. Esses fatores também atingiram o mercado de software. Neste contexto, surge o Desenvolvimento Distribuído de Software, onde os envolvidos em um determinado projeto estão dispersos.

Segundo [AUDY e PRIKLADNICKI], a distribuição do processo de desenvolvimento de software faz ampliar os desafios inerentes ao desenvolvimento

tradicional e gera novos desafios ao adicionar fatores como distância física e diferenças de fuso horário. Dessa maneira, é necessária a implantação de novas ferramentas, técnicas e modelos de colaboração para o suporte ao processo de desenvolvimento distribuído de software, assim, as empresas podem tratar da melhor maneira os aspectos do contexto distribuído.

Neste sentido, a utilização de ontologia como formalismo de representação do conhecimento nas técnicas, ferramentas e modelos que dão suporte ao desenvolvimento distribuído de software pode trazer muitos benefícios. Uma das vantagens da ontologia é a possibilidade de compartilhar um conhecimento comum sobre a estruturação da informação entre os envolvidos no processo. Além disso, o uso de ontologia permite modelar o conhecimento de determinado domínio, representar as informações com clareza e sem ambigüidade e separar a base de conhecimento de um sistema da sua implementação, fornecendo reusabilidade e portabilidade de conhecimento.

Assim, o objetivo deste trabalho é realizar uma revisão sistemática da literatura na área de DDS para identificar quais os modelos, técnicas, ferramentas e boas práticas que utilizam ontologias são utilizadas para desenvolver software no ambiente distribuído. Desta forma, espera-se uma contribuição para o aprimoramento das atividades de garantia da qualidade de software ao relatar quais melhores evidências provenientes da pesquisa podem ser integradas no processo de desenvolvimento de software.

Este trabalho está organizado da seguinte maneira: a Seção 2 descreve os principais conceitos de Desenvolvimento Distribuído de Software, Revisão Sistemática da Literatura e Ontologia; a Seção 3 relata a revisão sistemática a ser conduzida, apresentando detalhadamente o protocolo de revisão; a Seção 4 apresenta os resultados esperados com a execução da revisão sistemática; e por fim, a Seção 5 aborda as considerações finais.

2. Fundamentação Teórica

Esta seção apresenta os conceitos utilizados neste trabalho, como o Desenvolvimento Distribuído de Software, bem como os principais fatores que levam a esta nova abordagem da Engenharia de Software. Também são abordados os conceitos do processo de Revisão Sistemática da Literatura e as definições de Ontologias.

2.1. Desenvolvimento Distribuído de Software

O desenvolvimento de software de forma co-localizada (no mesmo espaço físico) tem se tornado cada vez mais custoso e menos competitivo para as empresas. Visando a redução de custos, a melhoria na qualidade de seus produtos, o aumento de produtividade e a competitividade global, várias empresas optam por distribuir seus processos de desenvolvimento em lugares diferentes, utilizando o desenvolvimento distribuído como uma alternativa para o desenvolvimento de software [AUDY E PRIKLADNICKI 2008].

O DDS é um modelo de desenvolvimento de software onde os envolvidos em um determinado projeto estão dispersos. Para [CARMEL 1999], as características fundamentais deste modelo de desenvolvimento são: a distância física, a diferença de fuso horário e as diferenças culturais entre os envolvidos no processo. Já [MEYER 2006] caracteriza o DDS pela colaboração e cooperação entre departamentos de

empresas e pela criação de equipes globais de desenvolvedores que trabalham em conjunto em um projeto comum, localizados em cidades ou países diferentes.

Outro conceito de DDS importante é apresentado por [KAROLAK 1998], que define Desenvolvimento Global de Software (GSD – Global Software Development) como uma modalidade de DDS que ocorre quando a distância física entre os elementos de um projeto distribuído envolve mais de um país.

Como citado anteriormente, várias empresas estão investindo na distribuição de seus processos de desenvolvimento de software em regiões diferentes visando obter maiores vantagens competitivas no mercado global de software. Entre os diversos benefícios que esse modelo de desenvolvimento oferece, [PRIKLADNICKI 2003] destaca:

- Possibilidade de desenvolvimento *follow-the-sun* (24 horas contínuas de trabalho), que permite o aumento de produtividade e a redução dos prazos de entrega dos produtos;
- Disponibilidade de recursos globais com custos mais baixos e a qualquer hora;
- Disponibilidade de recursos qualificados em áreas especializados;
- Oportunidade de realizar o desenvolvimento de software perto dos clientes;
- Possibilidade de formação de equipes virtuais para explorar as oportunidades de mercado;
- Escalabilidade, que permite a expansão das empresas para outras regiões;

2.2. Revisão Sistemática da Literatura

Em vista do quadro da relativa imaturidade da área de engenharia de software, muitas vezes a academia e a indústria de software se deparam com questões do tipo: “em qual tecnologia investir?” ou “qual linha de pesquisa devo direcionar meus esforços?”. Entretanto, as respostas para essas questões podem ser obtidas com uma utilização sistemática de abordagens baseadas em evidências. Um dos principais métodos do paradigma de práticas baseadas em evidências são as revisões sistemáticas, utilizadas para revelar evidências e construir um conhecimento mais amplo sobre um determinado domínio (TRAVASSOS E MAFRA, 2006).

A revisão sistemática da literatura (Systematic Literature Review – SLR) é uma prática de pesquisa que vem se destacando nos últimos anos. De acordo com (KITCHENHAM, 2004), a utilização de revisões sistemáticas fez a pesquisa na área de medicina avançar consideravelmente nas últimas décadas. Com o sucesso na medicina, as revisões sistemáticas estão se popularizando em outras áreas, como psiquiatria, ciências sociais e educação, porém, na área de Engenharia de Software, estão apenas iniciando. Mas ainda não estão bem estabelecidas na engenharia de software (OATES E CAPPER, 2009).

Segundo Travassos e Biolchini (2007) uma SRL permite elaborar uma revisão de literatura mais abrangente e não tendenciosa, produzindo resultados com valor científico. Utilizando padrões metodológicos rigorosos, esta forma de revisão permite sumarizar todos os estudos disponíveis e relevantes relacionados a um determinado

tópico de pesquisa. Com a análise dos dados coletados na revisão, é gerada uma base de conhecimento que responde algumas questões relacionadas ao tópico analisado.

No contexto da Engenharia de Software, a utilização de revisões sistemáticas pode gerar uma base de conhecimentos que permite caracterizar uma determinada tecnologia, ferramenta ou processo. Kitchenham (2004) acredita que a utilização dos resultados provenientes de uma SRL, em conjunto com experiência e valores humanos, pode ajudar o profissional no processo de tomada de decisão durante o desenvolvimento de software. Além disso, Travassos e Mafra (2006) afirma que a análise dos resultados de uma SRL pode ajudar a direcionar os esforços dentro de um projeto de pesquisa.

No início de uma revisão sistemática, alguns aspectos devem ser observados, como definir o processo utilizado para guiá-la, delimitar o objetivo da mesma e reconhecer a literatura. Travassos e Biolchini (2007) define o processo de revisão sistemática da literatura em três fases principais: Planejamento, Execução e Análise dos Resultados. Na figura 1 são detalhadas as fases de uma SRL.

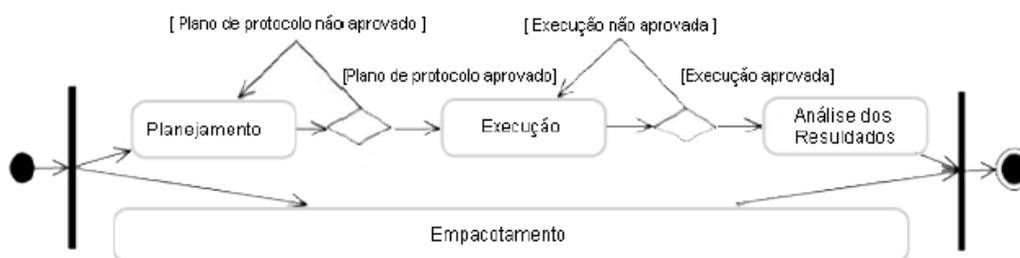


Figura 1. Processo de Revisão Sistemática

Fonte: Travassos e Biolchini (2007)

Na fase de Planejamento são definidos os objetivos da pesquisa, as questões de pesquisa, os métodos para executar a revisão e as estratégias de análise dos dados coletados, além de planejar as fontes e os critérios para a seleção dos trabalhos utilizados na revisão. Todas essas informações constituem o protocolo de revisão, que é fundamental para o início da execução da SRL. Ainda é sugerido que o protocolo seja revisado e avaliado por especialistas.

Na fase de Execução é realizada a busca dos estudos nas fontes pré-definidas e a avaliação dos mesmos, de acordo com os critérios definidos no protocolo de revisão. Posteriormente é feita a extração e a síntese dos dados relevantes para a questão de pesquisa.

Na última fase, a de Análise dos Resultados, é feita a formatação dos dados coletados durante a execução da revisão. Também é importante especificar o mecanismo de publicação dos resultados. No final, é elaborado um relatório de revisão sistemática, onde são mapeadas as informações levantadas durante a revisão, formando um conhecimento mais amplo sobre o tópico abordado na pesquisa.

2.3. Ontologias

O conceito de ontologia surgiu na área da Filosofia, como uma forma de descrever os objetos do mundo real. Nas últimas décadas, as ontologias também vêm sendo estudadas na área da Ciências da Computação, principalmente no campo de Inteligência Artificial (IA), onde é utilizada como uma forma de representar o

conhecimento de maneira legível para o computador. Na Engenharia de Software, a possibilidade de separar o conhecimento do domínio e a estratégia de solução da aplicação pode trazer benefícios no desenvolvimento de software (Tercio, 2003).

Ontologia é uma forma de representar e compartilhar conhecimento. Gruber (1993) define bem o que é ontologia: “Ontologia é uma especificação explícita de uma conceituação”. Segundo Tercio (2003), a utilização de ontologias permite estruturar de forma mais clara o conhecimento.

Uma ontologia é formada por um conjunto de classes, também denominadas entidades, que representam os conceitos do domínio e são organizadas de forma hierárquica. As classes possuem atributos e propriedades que as identificam. Cada classe possui um conjunto de indivíduos, que representam as instâncias dos conceitos.

Para Neto (2006), a vantagem de utilizar ontologias é a possibilidade de separar a modelagem do conhecimento da implementação do software. Neste sentido, a utilização de ontologias permite uma maior flexibilidade e portabilidade da representação do conhecimento. Tercio (2003) destaca que a utilização de um domínio pré-definido, através de ontologias, permite uma definição mais precisa da informação e provê consistência semântica à informação.

4. Metodologia e Estado Atual do Trabalho

3.1. Protocolo da Revisão

3.1.1. Questões de Pesquisa

3.1.2. Estratégia de Busca dos Estudos Primários

3.1.3. Estratégia de Seleção dos Estudos Primários

3.1.4. Critérios de Avaliação da Qualidade dos Estudos Primários

3.1.5. Estratégia de Extração dos Dados

3.1.6. Estratégia de Sumarização dos Dados

3.2. Execução da Revisão

4. Resultados Esperados

5. Considerações Finais

Referências

- Herbsleb, J. D. (2001) “Global Software Development”, IEEE Software, EUA, p. 16-20.
- Audy, J. L. N., Prikladnicki, R. (2008), Desenvolvimento Distribuído de Software, Editora Elsevier.
- Carmel, E. (1999) Global Software Teams – Collaborating Across Borders and Time-Zones. EUA, Prentice-Hall.
- Meyer, B. (2006). The Unspoken Revolution in Software Engineering. IEEE Computer.
- Karolak, D. W. (1998) Global Software Development – Managing Virtual Teams and Environments, EUA, IEEE Computer Society.
- 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.
- Travasso, G. H. and Mafra, S. N. (2006) Estudos Primários e Secundários Apoiando a Busca por Evidências em Engenharia de Software, PESC (Programa de Engenharia de Sistemas e Computação) / COPPE / UFRJ, Rio de Janeiro.
- Kitchenham, B. (2004) “Evidence-based Software Engineering”, Proceedings of the 26th International Conference on Software Engineering (ICSE’04) , IEEE Computer Society, EUA, p. 273-281.
- Oates, J. B. and Capper, G. (2009) “Using Systematic Reviews and Evidence-based Software Engineering with Master Students”, International Conference on Evaluation & Assessment in Software Engineering (EASE 2009).
- Travassos, G. and Biolchini, J. (2007) “Revisões Sistemáticas Aplicadas à Engenharia de Software”, XXI Brazilian Symposium on Software Engineering (SBES), Brasil.
- Silva, T. M. S. (2003) “Extração de Informações para Busca Semântica na Web Baseada em Ontologias”, Dissertação de Pós-graduação em Engenharia Elétrica, Universidade Federal de Santa Catarina, Florianópolis.
- Gruber, T. (1993). A Translation Approach to Portable Ontology Specifications. Relatório Técnico, Knowledge Systems Laboratory, Computer Science Department. Stanford University.
- Neto, R. F. B. (2006). Um Processo de Software e um Modelo Ontológico para Apoio ao Desenvolvimento de Aplicações Sensíveis a Contexto. *PHD Thesis*, ICMC-USP.