

Emprego de técnicas de representação do conhecimento como forma de apoio à engenharia de requisitos

Gilberto Dessbesell Jr¹, Eduardo Kroth¹

¹Departamento de informática – Universidade de Santa Cruz do Sul (UNISC)
Av. Independência, 2293 – 96.815-900 – Santa Cruz do Sul – RS – Brasil
gjunior@mx2.unisc.br, kroth@unisc.br

Abstract. The Requirements Engineering is fundamental part of the process of software development. Even if a program uses the most advanced technologies and has the best source code, it is not useful if it does not meet your requirements. The large number of failures occurring in software projects, attributed to deficiency of capturing and analyzing their requirements emphasize the importance of this area. In this work, we used knowledge representation in order to store the knowledge about the requirements of a project and then use it as a support to the capture and analysis of requirements. In addition, techniques of similarity were applied on the requirements, to ensure they are contextualized and mutually consistent.

Resumo. A Engenharia de Requisitos é parte fundamental do processo de desenvolvimento de software. Mesmo que um programa utilize as tecnologias mais avançadas e tenha o melhor código fonte, ele não será útil caso não atenda aos seus requisitos. O grande número de insucessos ocorridos em projetos de software, atribuídos à deficiência dos processos de captação e análise de requisitos enfatizam a importância desta área. Neste trabalho foi utilizada representação do conhecimento para armazenar o conhecimento sobre os requisitos de um projeto e utilizá-lo como suporte à captação e análise de requisitos. Também foram aplicadas técnicas de similaridade para garantir que os requisitos estejam contextualizados e consistentes entre si.

1. Introdução

Grande parte dos problemas existentes no processo de desenvolvimento de software está relacionado à má qualidade na captação e na análise dos requisitos [20]. Essa deficiência pode ser verificada através de produtos que atendem somente parcialmente às necessidades dos *stakeholders*, ou ainda pior, que sejam diferentes do que foi solicitado por eles.

A publicação Chaos Report 2009 [20], afirma que no ano de 2009, nos Estados Unidos, 24% dos projetos de software foram cancelados ou nunca utilizados. Outros 44% atrasaram, estouraram o orçamento ou tiveram seu escopo reduzido. O restante, apenas 32%, foi concluído com sucesso. Nos projetos falhos, a principal causa apontada (13%) foram os requisitos incompletos. Nos projetos com sucesso parcial, 24% dos problemas foram atribuídos aos requisitos e especificações incompletos. O estudo divulga ainda que a declaração de requisitos clara e limpa foi um dos fatores (13%) que mais influenciou nos projetos que obtiveram sucesso.

Várias técnicas têm sido desenvolvidas para melhorar a qualidade da captação e da análise dos requisitos, porém a maioria delas aborda apenas as formas como os requisitos são captados e analisados, deixando de considerar o fato de que a interpretação e o gerenciamento dos requisitos, por parte das pessoas envolvidas nestas tarefas, é uma questão muito importante neste processo e que depende da capacidade e do conhecimento de seres humanos.

A Representação do Conhecimento é uma subárea da inteligência artificial (IA), que busca formas de representar, de maneira computacional, o conhecimento criado por seres humanos, realizando a classificação e o armazenamento do conhecimento de maneira estruturada, em bases de conhecimento. Através de estruturas, como as ontologias, é possível organizar o conhecimento adquirido por um grupo de pessoas e depois torná-lo disponível para consultas computacionais. O objetivo desta abordagem é criar uma base de conhecimento em formato digital e não volátil, preservando e acumulando cada vez mais conhecimento a respeito de um determinado assunto.

Para consultar este conhecimento armazenado, a Recuperação de Informações dispõe de atividades que envolvem os aspectos intelectuais de descrição de informação e sua especificação para busca. Na área da computação, esta atividade é mais bem descrita como a recuperação de informações de bancos de dados em que predominam informações na forma textual, não tabuladas.

Com base nestes conceitos, este trabalho realizou consultas e apontamentos sobre bases de conhecimento de projetos de software, atingindo um resultado próximo, ou melhor, ao que um ser humano atingiria, dispondo das mesmas informações.

2. Engenharia de requisitos

[10] define um requisito de software como uma condição ou capacidade que um software deve realizar ou ter, para atender à necessidade de um *stakeholder*. [6] amplia estas definições, ao afirmar que requisitos são propriedades que um software deve possuir para atender às necessidades dos *stakeholders* no ambiente onde será utilizado. Abrangendo estas definições, o objetivo de um requisito de software é determinar os contratos, padrões, especificações e demais documentos, ou seja, as condições e capacidades impostas externamente ao contexto de desenvolvimento de software, que devem ser atendidas por um software [8].

[19] recomenda as características que uma especificação de requisitos de software deve ter. Ela deve ser, ao mesmo tempo: correta, não ambígua, completa, consistente, priorizável, verificável, flexível e rastreável. [8] contribui com esta definição reforçando a ideia de que é importante, principalmente quando se utiliza a linguagem natural, garantir que o conteúdo da Especificação de Requisito de Software (ERS) seja claro, preciso e não ambíguo.

Ao mesmo tempo em que a utilização da linguagem natural facilita o entendimento dos requisitos, por ser simples e de fácil compreensão, ela também dificulta alguns outros aspectos. Em casos em que é necessário ser muito específico ao descrever um requisito, o documento acaba se tornando muito detalhado e, com isso, difícil de ler. A utilização de linguagem natural também resulta em especificações confusas em alguns casos, pois requisitos funcionais e não-funcionais acabam sendo misturados em uma mesma especificação [16].

[18] definem que Engenharia de Requisitos (ER) fornece o mecanismo adequado para a compreensão do que os *stakeholders* querem, analisando a necessidade e a viabilidade das solicitações, negociando uma solução razoável, especificando uma solução de forma não-ambígua, validando esta especificação e gerenciando os requisitos para que eles sejam transformados em um software funcional.

[14] destaca que não existe uma forma incontestável de assegurar que a especificação de um sistema está propriamente de acordo com as necessidades dos *stakeholders*. Este é um desafio complexo enfrentado pelos engenheiros de requisitos, e o melhor modo de encará-lo é através de um processo consistente de ER.

3. Recuperação de informação

[12] define Recuperação de Informação (RI) como a atividade que envolve os aspectos intelectuais de descrição de informação (indexação, padronização) e sua especificação para busca, além de qualquer técnica, sistema ou máquina empregada para realizar ou auxiliar essas tarefas relevantes para o usuário. De modo a diferenciar os termos "Recuperação de Dados" (RD) e RI, [1] afirma que a RD consiste em determinar quais os documentos de uma coleção contém as palavras-chave contidas na pergunta do usuário. A principal diferença é que na RI os textos estão em linguagem natural e, na maioria das vezes, não estão bem estruturados, existindo a possibilidade de ser semanticamente ambíguos.

Segundo [9], um Sistema de Recuperação de Informação (SRI) é capaz de armazenar, recuperar e gerenciar informação. Apesar da forma de um objeto em um SRI ser diversa, o aspecto textual tem sido o único tipo de dado que permite um processamento funcional completo. O objetivo de um SRI é minimizar o custo de um usuário localizar a informação desejada, partindo do pressuposto que o usuário poderia localizar a informação analisando todos os documentos de uma coleção, manualmente. O custo pode ser entendido como o tempo gasto pelo usuário para completar a tarefa de recuperação da informação relevante.

Segundo [7], um dos métodos mais utilizados para o registro e posterior recuperação de informações em textos livres é a indexação. A ideia de indexar é produzir um conjunto de informações menor, porém mais eficiente, para representar o conteúdo e facilitar a recuperação do documento original. Já a técnica denominada vocabulário controlado [11] facilita muito a localização das informações, pois os usuários estão acostumados a utilizar os termos comumente utilizados na sua área de interesse.

O processo de indexação automática tem como objetivo identificar termos relevantes nos documentos de uma coleção e armazená-los em uma estrutura de índice. Normalmente são encontradas as seguintes fases nesse processo: identificação de termos, remoção de *stopwords*, normalização morfológica e identificação de termos compostos. Para cada uma dessas etapas, diversas técnicas podem ser adotadas. A ordem de aplicação de cada etapa também pode variar, ou ainda, algumas delas podem não ser utilizadas [15].

4. Ontologias

O termo "ontologia" começou a ser empregado na Ciência da Computação no início dos anos 90, em projetos de organização de grandes bases de conhecimento. Desde os anos 70, a subárea da IA já se preocupava com a organização e manipulação de bases de conhecimento, mas foi a partir da década de 90 que houve maior interesse no estudo de bases de conhecimento compartilháveis e reutilizáveis [13].

Para [4], uma ontologia é uma especificação formal e explícita de uma conceitualização compartilhada. Uma especificação claramente definida (explícita) e processável por uma máquina (formal), de um modelo abstrato compartilhado. Para [17], o termo "ontologia" foi adaptado e, para os profissionais da Ciência da computação, uma ontologia é um documento ou um arquivo que define formalmente as relações entre termos e conceitos, mantendo, desta forma, semelhanças com os tesouros utilizados para a definição de vocabulários controlados.

Ainda segundo [17], o objetivo da construção de uma ontologia é suprir a necessidade de um vocabulário compartilhado para a troca de informações entre os membros de uma comunidade, sejam eles seres humanos ou agente inteligentes. [2] concorda com esta definição, adicionando que as ontologias servem como base para garantir uma comunicação livre de ambiguidade. Ao que [3] complementa, ainda, dizendo que uma ontologia fornece um entendimento comum e compartilhado de um domínio, que pode ser comunicado através de pessoas e sistemas.

Segundo [5], as ontologias auxiliam no processo de comunicação e no trabalho colaborativo entre diferentes pessoas, com diferentes perspectivas de um domínio, pois criam um "esqueleto unificado do conhecimento". Ainda segundo os autores supracitados, as ontologias podem servir como fonte de aprendizado, publicação e referência. Isso ocorre devido ao fato de ontologias serem o resultado de um grande consenso sobre a estrutura de um domínio e serem capazes de fornecer informações confiáveis e objetivas sobre este domínio. Da mesma forma, especialistas de domínio podem usar ontologias para compartilhar seu entendimento da conceitualização e estrutura do domínio em questão.

Existem diversas linguagens propostas para a representação de ontologias, cada qual com suas particularidades, porém não existe um consenso sobre a padronização de uma linguagem para este fim. Uma opção que aponta para este consenso é a *Ontology Web Language* (OWL), uma linguagem para representação do conhecimento definida como linguagem padrão para este fim pelo [21].

5. Solução desenvolvida - OntoRE

A seção inicia com uma visão geral do "Ontology based Requirements Engineering" (OntoRE), com o objetivo de apresentar a solução como um todo, bem como o fluxo do processo e as principais funcionalidades desenvolvidas. Em seguida, cada uma das funcionalidades é detalhada, para que seja possível compreender onde e de que forma os conceitos apresentados no desenvolvimento do trabalho foram empregados.

O fluxo do processo pode ser acompanhado através da figura 1 e das letras (A, B, C, D e E) que acompanham a descrição da visão geral da solução proposta.

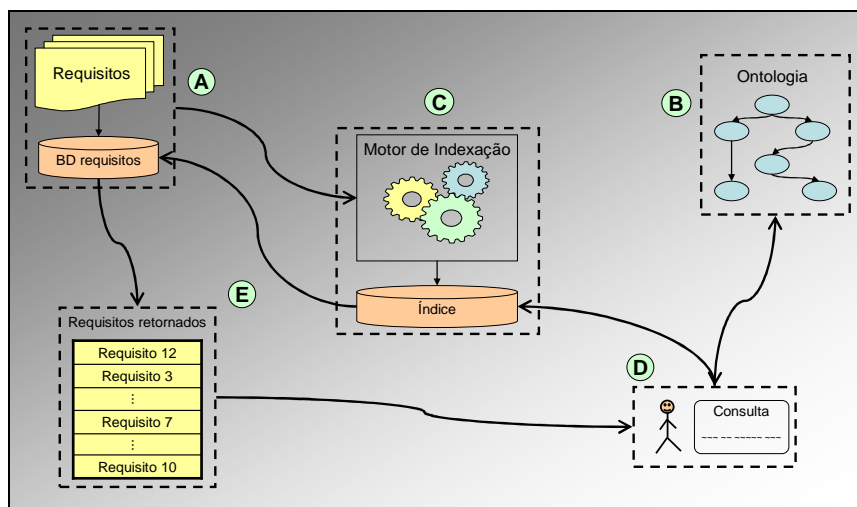


Figura 1 – Fluxo do processo do OntoRE.

A primeira funcionalidade desenvolvida foi a de manter a coleção de ERS (A) de um projeto. Optou-se por utilizar ERS textuais e em linguagem natural e armazená-las em um banco de dados relacional. O objetivo desta funcionalidade é criar um banco de dados com a coleção de ERS utilizada em um projeto.

Para o funcionamento deste trabalho, é necessária uma ontologia (B) que represente o conhecimento presente nas ERS utilizadas. Esta tarefa tem o objetivo de criar uma base de conhecimento com vocabulário controlado e livre de ambiguidades para o projeto. A ontologia utilizada foi construída em uma ferramenta externa, pois não foi objetivo deste trabalho criar um ambiente para manutenção de ontologias.

Após passar por um processo de tratamento dos dados, as ERS são submetidas a um motor de indexação (C), que resulta em uma base de requisitos indexados. Neste momento, as ERS podem ser utilizadas pelos usuários, através das funcionalidades descritas a seguir.

A segunda funcionalidade desenvolvida foi um ambiente de consulta (D), onde um usuário do sistema pode recuperar as ERS de seu interesse. Ele pode formular consultas textuais, sem necessariamente utilizar o vocabulário controlado do projeto, que após serem relacionadas com a base de conhecimento retornam as ERS relevantes à consulta realizada. As ERS selecionadas pelo mecanismo de busca são retornadas ao usuário classificadas por um ranking de relevância (E), para que as prováveis ERS que atendam às necessidades do usuário sejam verificadas primeiro, evitando o desperdício de esforço na tarefa de encontrar a(s) ERS desejada(s).

A terceira funcionalidade implementada foi a possibilidade de sugerir uma nova forma de escrever uma ERS, substituindo palavras que não façam parte do vocabulário controlado do projeto por termos que façam. O objetivo desta funcionalidade é consolidar e reduzir o vocabulário das ERS, de modo a facilitar a comunicação e simplificar a linguagem utilizada entre os *stakeholders* de um projeto, evitando possíveis problemas de comunicação.

A quarta funcionalidade criada foi a possibilidade de apresentar ao usuário ERS que possivelmente possuam descrição semelhante a uma ERS específica. O objetivo

desta funcionalidade é apresentar ao usuário uma lista de ERS que devem ser consideradas, pois possuem relação com a ERS que está sendo analisada.

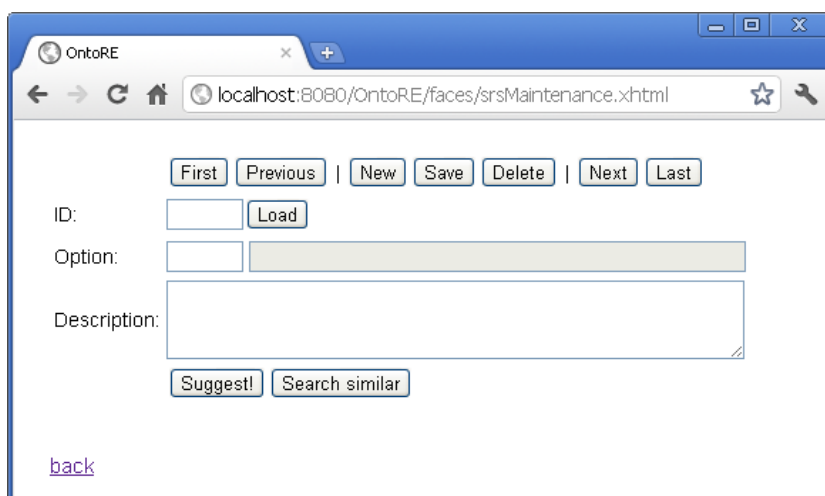
5.1 Funcionalidades

A tela inicial do OntoRE apresenta um menu que possibilita o acesso às funcionalidades desenvolvidas. O menu possui as seguintes opções: *Index*, *SRS maintenance* e *Search*. Ao acessar a opção *Index* o banco de dados de ERS será acessado para recuperar todas as ERS do projeto suportado pelo OntoRE e submetê-las ao processo de indexação.

Inicialmente é realizado um procedimento de limpeza, que remove caracteres indesejados das ERS. Em seguida, cada ERS passa pelo processo de remoção de *stopwords*, para então ser indexada. Ao final do processo, tem-se um índice referente as ERS do projeto, ao qual já é possível submeter consultas e recuperar ERS.

Através da opção *SRS maintenance* (figura 2), é possível adicionar novas ERS ao projeto, além de alterar e excluir ERS existentes. Através dos botões existentes na parte superior da tela, também é possível navegar entre as ERS do projeto.

Após informar a descrição de uma nova ERS ou de consultar uma ERS existente, é possível utilizar o botão "*Search similar*", visível na figura 2, para buscar ERS existentes no banco de dados que, de acordo com o conhecimento modelado na ontologia, tenham alguma semelhança com a ERS de referência.



The screenshot shows a web browser window titled 'OntoRE' with the address 'localhost:8080/OntoRE/faces/srsMaintenance.xhtml'. The page content includes a set of navigation buttons: 'First', 'Previous', 'New', 'Save', 'Delete', 'Next', and 'Last'. Below these are input fields for 'ID:' with a 'Load' button, 'Option:', and a larger 'Description:' text area. At the bottom of the form are buttons for 'Suggest!' and 'Search similar'. A 'back' link is located at the bottom left of the page.

Figura 2 – Manutenção de ERS.

O início do processo é mesmo da funcionalidade anterior: o texto da ERS passa pelo processo de limpeza e as palavras que fazem referência a algum termo da ontologia são identificadas. Para cada palavra identificada, é verificada na ontologia a existência de sinônimos. A pesquisa é então realizada substituindo cada palavra que faz referência a um termo da ontologia por todos seus sinônimos.

Pode ser atribuído ainda um peso de relevância aos conjuntos de palavras similares existentes na pesquisa, pois o fato deles existirem na ontologia pressupõe que são de maior relevância no contexto. Ao resultado da busca é aplicado ainda um ponto

de corte, referente à pontuação de relevância dada a cada resultado da pesquisa por ERS similares. Este ponto de corte tem como objetivo exibir apenas as ERS relevantes, desprezando as que tenham pouca similaridade com a ERS analisada.

Por fim, como pode ser visto na figura 3, é apresentada uma lista das ERS identificadas como similares. Estas ERS não descrevem, necessariamente, a mesma funcionalidade da ERS analisada. Elas devem ser analisadas, pois possuem relação com a ERS que está sendo analisada. Cabe ao usuário decidir se as ERS apresentadas têm, de fato, relevância no contexto em que a ERS está sendo analisada. Elas servem como uma espécie de lembrete, para que o usuário possa analisar as situações que elas descrevem e assim melhorar a qualidade da análise da ERS.



Figura 3 – Exemplo de consulta por ERS similares.

De forma muito semelhante à funcionalidade anterior, ao utilizar o botão "Suggest!", visível na figura 2, o sistema sugere uma nova maneira de escrever a ERS selecionada, utilizando os termos do vocabulário controlado definido na ontologia.

Quando o usuário utiliza a funcionalidade de sugestão de ERS similares, a descrição da ERS que será analisada é submetida ao mesmo processo de limpeza utilizado no momento da indexação. Em seguida, um algoritmo verifica quais palavras do texto fazem referência a algum termo existente na ontologia. Para cada palavra que exista na ontologia, verifica-se o termo do vocabulário controlado que deve ser utilizado em substituição àquela palavra, caso exista um. Por fim, como mostra a figura 4, é apresentada ao usuário uma sugestão de descrição da ERS, onde as alterações sugeridas pelo sistema aparecem destacadas.



Figura 4 – Alterações sugeridas na descrição da ERS.

A opção *Search*, apresentada na figura 5, é de grande importância para o sistema, pois ela é a base para as demais funcionalidades desenvolvidas. O tratamento do texto no processo de busca é bastante semelhante ao de indexação. Novamente, o processo inicia com a limpeza do texto que será submetido à busca, para que sejam removidos caracteres especiais e sinais de pontuação. Após identificar os termos que existem na ontologia do sistema, são recuperados e adicionados à busca os termos de um nível imediatamente acima do termo de pesquisa, ou seja, o termo pai. Cada grupo de termos, os da busca e os adicionados pelo sistema, recebem um grau de relevância. A ideia deste comportamento é adicionar uma espécie de contexto à pesquisa e, desta forma, torná-la mais precisa.

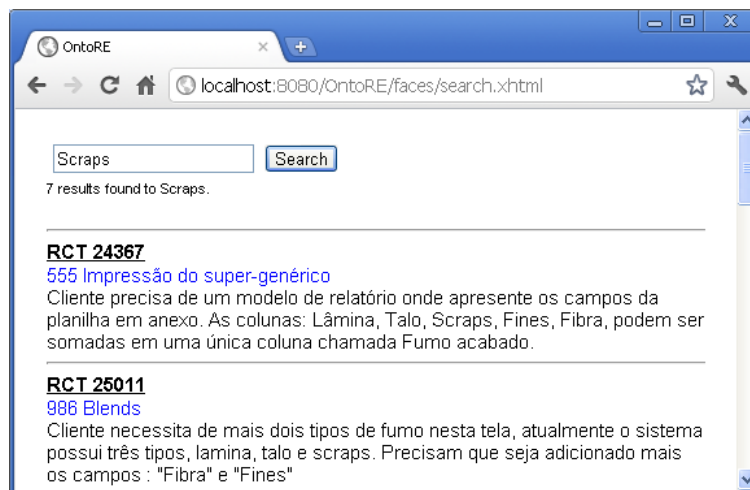


Figura 5 – Funcionalidade de consulta de ERS.

6. Estudo de caso

Com o objetivo de avaliar, de maneira prática e em um ambiente real, os resultados obtidos através do OntoRE, realizou-se um pequeno estudo de caso. Foram utilizadas cerca de 12.000 (doze mil) ERS da coleção de requisitos do projeto de um ERP que vem sendo desenvolvido há 5 anos.

O objetivo do estudo de caso foi verificar a efetividade da funcionalidade de sugestão de ERS similares. Para isso, o conjunto de ERS foi adicionado ao sistema e submetido à funcionalidade em ordem cronológica, simulando o crescimento natural do projeto. O resultado pode ser verificado na figura 6.

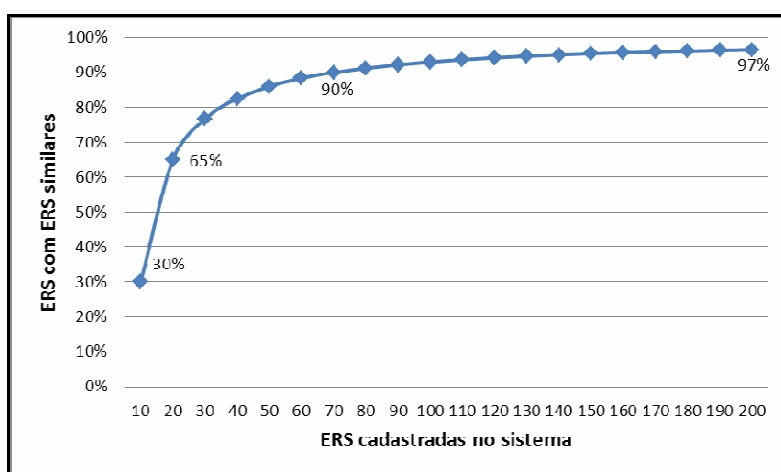


Figura 6 - Percentual de ERS com ERS similares.

Para a apresentação do resultado foram consideradas apenas as primeiras 200 ERS, pois já é possível apontar que praticamente todas as ERS do projeto possuem algum tipo de ERS similar. Considerando a totalidade de ERS utilizadas no estudo, concluiu-se que mais de 99% delas possuíam ao menos uma ERS similar.

7. Conclusão

A demanda crescente por informações que atendam aos mais diversos tipos de usuários impulsiona o surgimento de grandes projetos de desenvolvimento de software. Projetos, estes, que têm porte diretamente proporcional ao nível de importância que deve ser dado às tarefas desenvolvidas pela área de engenharia de software, em especial a subárea de ER, caso aspirem atingir êxito nas principais esferas de tempo, custo e qualidade.

Analisando os problemas responsáveis pelas falhas da maioria dos projetos de software nos últimos anos, é possível afirmar que os problemas enfrentados hoje não são relativos a como um software deve ser desenvolvido, mas a o quê deve ser desenvolvido. Esta análise faz-se necessária para que o resultado final do produto desenvolvido de fato agregue valor às atividades de seus usuários e não apenas se torne uma maneira digital de burocratizá-las.

Este trabalho introduziu tais problemas e apresentou maneiras de abordá-los, visando a melhoria da qualidade das tarefas de captação e análise de requisitos de software e, desta maneira, da qualidade do resultado produzido pelo processo de desenvolvimento de software. Foram desenvolvidas funcionalidades que auxiliam o analista de sistemas a ampliar seu campo de visão sobre os requisitos de um projeto, tendo assim maior qualidade na informação que ele considera ao tomar decisões a respeito das funcionalidades que devem ser desenvolvidas.

Referências

1. Baeza-Yates, R.; Ribeiro-Neto, B. (1999) "Modern Information Retrieval". ACM Press, New York.
2. Breitman, K. (2005) "Web Semântica: a Internet do future". LCT, Rio de Janeiro.
3. Cunha, L. M. S. (2002) "Web Semântica: estudo preliminar". Embrapa Informática Agropecuária, Campinas.
4. Fensel, D. (2001) "Ontologies: A Silver Bullet For Knowledge Management And Electronic Commerce". Springer, Berlin.
5. Gasevic, D.; Djuric, D.; Devedzic, V. (2006) "Model driven architecture and ontology development". Springer-Verlag, Heidelberg.
6. Goguen, J. (1996) "Formality and Informality in Requirements Engineering". International Conference on Requirements Engineering, Colorado Springs, EUA: IEEE Computer Society.
7. Hersh, W. R.; Detmer, W. M.; Frisse, M. E. (2001) "Information Retrieval Systems" In: SHORTLIFFE E. H. et al. Medical Informatics: Computer Applications in Health Care and Biomedicine. 2. ed. Springer p. 539-72, New York.
8. Kazmierczak, E. (2003) "Requirements Engineering". Course-work Masters - The University of Melbourne, Melbourne.
9. Kowalski, G. (1997) "Information Retrieval Systems: Theory and Implementation". Kluwer Academic Publishers, Boston.
10. Kruchten, P. (2003) "The Rational Unified Process: An Introduction". 3. ed. Addison-Wesley Professional, Boston.
11. Lancaster, F. W. (1968) "Information Retrieval Systems: Characteristics, Testing and Evaluation". John Wiley & Sons, New York.
12. Moores, C. N. (1951) "Data coding applied to Mechanical Organization of Knowledge". American Documentation, Apud (GUPTA e JAIN, 1997), v. 2, p. 20-32.
13. Moreira, A.; Alvarenga, L.; Oliveira, A de P. (2004) "O nível do conhecimento e os instrumentos de representação: tesouros e ontologias". DataGramaZero - Revista de Ciência da Informação, v. 5 n. 6.
14. Pressman, R. (2001) "Software Engineering: A Practitioner's Approach". 5. ed. McGraw-Hill, New York.
15. Riloff, E. (1995) "Little words can make big difference for text classification". In: Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval (SIGIR'95). Proceedings... ACM Press. p. 130-136, Seattle, USA.
16. Sommerville, I. (2004) "Software Engineering". 7. ed. Addison-Wesley, Boston.
17. Souza, R. R.; Alvarenga, L. (2004) "A Web Semântica e suas contribuições para a ciência da informação". Ci. Inf., v 33, n. 1, p. 132-141, Brasília.
18. Thayer, R. H.; Dorfman, M. (1997) "Software Requirements Engineering". 2. ed. IEEE Computer Society Press, Los Alamitos, CA.
19. The Institute of Electrical and Electronics Engineering. (1998) "IEEE Recommended Practice for Software Requirements Specifications: IEEE Std 830-1998". New York.
20. The Standish Group. (2009) "CHAOS Report 2009". Boston, MA.
21. World Wide Web Consortium, <http://www.w3.org>.