



Transinformação

ISSN: 0103-3786

ISSN: 2318-0889

Pontifícia Universidade Católica de Campinas

Zacarias, Rodrigo Oliveira; Jacyntho, Mark Douglas
de Azevedo; Vasconcelos, Aline Pires Vieira de
Um método para criação de tesouros semânticos de requisitos usando a ontologia SKOS
Transinformação, vol. 32, e190086, 2020
Pontifícia Universidade Católica de Campinas

DOI: 10.1590/2318-0889202032e190086

Disponível em: <http://www.redalyc.org/articulo.oa?id=384365244016>

- Como citar este artigo
- Número completo
- Mais informações do artigo
- Site da revista em redalyc.org

redalyc.org
UAEM

Sistema de Informação Científica Redalyc

Rede de Revistas Científicas da América Latina e do Caribe, Espanha e Portugal

Sem fins lucrativos acadêmica projeto, desenvolvido no âmbito da iniciativa
acesso aberto

Um método para criação de tesouros semânticos de requisitos usando a ontologia SKOS

A method for creating semantic thesauri of requirements using the SKOS ontology

Rodrigo Oliveira ZACARIAS¹  0000-0003-0005-4669

Mark Douglas de Azevedo JACYNTHO¹  0000-0003-3910-1442

Aline Pires Vieira de VASCONCELOS¹  0000-0002-2683-595X

Resumo

Problemas relacionados à especificação de requisitos, tais como ambiguidade e incompletude, ainda são recorrentes nos processos de desenvolvimento de sistemas de informação. O reúso de requisitos é um dos mecanismos que podem auxiliar na redução desses contratemplos. Nesse sentido, o objetivo deste trabalho é propor um método para a criação e para a publicação de tesouros semânticos de requisitos para reúso, utilizando tecnologias e padrões da *Web Semântica* e de acordo com os princípios *Linked Data*. Para descrição formal desses tesouros, a ontologia central utilizada é a *Simple Knowledge Organization System*. Esse modelo ontológico fornece um conjunto de axiomas e de propriedades voltados para criação de tesouros, permitindo documentar de forma precisa e fidedigna, em um grafo de conhecimento, a definição, a hierarquia e outros inter-relacionamentos entre os requisitos de um sistema. Também é apresentado um protótipo de *Web service* que funciona como repositório para reúso e demonstra o método na prática. É descrito um estudo sobre a viabilidade da implementação da proposta realizado com profissionais, em que foram promovidos uma discussão em grupo e um posterior preenchimento individual de um questionário de avaliação. O estudo obteve resultados favoráveis, em sua maioria, e algumas sugestões de melhoria foram apontadas. Os participantes consideraram a proposta relevante para a Engenharia de Requisitos e com potencial de expansão, uma vez que as diretrizes apresentadas permitem a criação de novos tipos de inferência e naveabilidade sobre os requisitos armazenados.

Palavras-chave: Ontologia SKOS. RDF. Requisitos. Reúso. Tesouros semânticos.

Abstract

Problems related to requirements specification, such as ambiguity and incompleteness, are still recurrent within information system development processes. Reusing requirements is one of the mechanisms that can help reduce these setbacks. In this sense, this paper aims to propose a method for creating and publishing reuse-oriented semantic thesauri of requirements, us-

¹ Instituto Federal de Educação, Ciência e Tecnologia Fluminense, Diretoria de Extensão, Pesquisa e Pós-Graduação, Programa de Pós-Graduação em Sistemas Aplicados à Engenharia e Gestão. R. Dr. Siqueira, 273, Parque Dom Bosco, 28030-130, Campos dos Goytacazes, RJ, Brasil. Correspondência para/Correspondence to: M. D. A. JACYNTHO. E-mail: <mjacyntho@iff.edu.br>.

Artigo elaborado a partir da dissertação de R. O. Zacarias, intitulada "Uma proposta de documentação e reúso de requisitos baseada em tesouros semânticos". Instituto Federal Fluminense, Campos dos Goytacazes, 2020.

Apresentado em 11 de novembro de 2019, reapresentado em 19 de maio de 2020, aprovado em 8 de julho de 2020.

Como citar este artigo/How to cite this article

Zacarias, R. O.; Jacyntho, M. D. A.; Vasconcelos, A. P. V. Um método para criação de tesouros semânticos de requisitos usando a ontologia SKOS. *Transinformação*, v. 32, e190086, 2020. <https://doi.org/10.1590/2318-0889202032e190086>



ing the Semantic Web's technologies and standards, and following the principles of Linked Data. The central ontology used for a formal description of these thesauri is the Simple Knowledge Organization System. This ontological model provides a set of thesaurus-creating axioms and properties, allowing accurate and reliable documentation of the definition, hierarchy, and other interrelationships among a system's requirements, in a knowledge graph. We present a prototypical Web service that functions as a repository for reuse and demonstrates the method in practice. We also describe a study about the feasibility of the proposal, which was carried out with professionals and included a group discussion and later, the completion of individual evaluation questionnaires. The study's results were mostly favorable, and some suggestions for improvement were appointed. The participants considered the proposal relevant to Requirements Engineering and with potential for expansion, since the guidelines presented allow the creation of new kinds of inference and navigability over the stored requirements.

Keywords: SKOS ontology. RDF. Requirements. Reuse. Semantic thesauri.

Introdução

A estruturação e a implantação de um sistema (e.g. um software), em uma organização, nem sempre é um processo trivial, exigindo pleno entendimento do domínio de conhecimento, de requisitos e de regras de negócio. Segundo Xavier (2001), um domínio pode ser definido como uma coleção de artefatos e de componentes de sistema, especializados em uma tarefa, com o intuito de fornecer aos usuários um conjunto de funções de uma área de conhecimento em particular.

O processo de desenvolvimento de um sistema é composto pelas etapas de elicitação e análise de requisitos, projeto, desenvolvimento, teste e implantação. A etapa dos requisitos é uma das mais importantes do ciclo de vida do sistema e deve ser realizada cuidadosamente, pois, caso contrário, pode levar o projeto ao fracasso (Pressman, 2011). A identificação de problemas nessa fase inicial é crucial, uma vez que permite adotar uma solução de baixo custo e com menor impacto no escopo do projeto, tendo em vista o menor número de mudanças a serem realizadas (Montequin *et al.*, 2014).

Dentro de um domínio, os requisitos são descrições de ações ou de serviços que um sistema deve executar, bem como de restrições ao seu funcionamento. Tais requisitos são artefatos que refletem as necessidades quantificadas e documentadas, desejos e expectativas de patrocinadores, clientes e outros *stakeholders* (Sommerville, 2011).

A obtenção de requisitos pode ser uma tarefa complexa, visto que é necessário profundo conhecimento sobre o domínio de atuação do sistema. Em muitos casos, essa complexidade está relacionada à dificuldade dos *stakeholders* em expressar suas necessidades e dos profissionais, sem experiência naquele domínio, em entendê-las completamente (Kaiya *et al.*, 2010). Como normalmente os requisitos são expressos em linguagem natural, é comum surgirem problemas relacionados à ambiguidade e à completude. Por isso, Palomares *et al.* (2017) destacam que o reúso auxilia na minimização desse problema.

Diamantopoulos e Symeonidis (2018) destacam reúso de requisitos e demais artefatos de sistema como um conceito importante que agrupa diversos benefícios ao projeto, tais como redução de tempo e de esforços empregados em cada uma de suas fases, e, consequentemente, proporciona aumento de produtividade e de qualidade no desenvolvimento do sistema. Para que essa atividade ocorra de forma eficiente, é necessária a implementação de mecanismos que permitam a disseminação e a recuperação dessas informações.

Nesse âmbito, o uso de Sistemas de Organização do Conhecimento (SOC) pode auxiliar na execução dessas atividades. Os SOC podem ser definidos como estruturas ou esquemas com o intuito de auxiliar na representação, organização, recuperação e gestão do conhecimento. Alguns exemplos são vocabulários controlados, taxonomias, classificações e tesouros (Altounian; Gomes, 2016).

Segundo Fagundes *et al.* (2020), tesouros podem atuar nas etapas de elicitação e de especificação de requisitos auxiliando na compreensão de termos específicos do domínio, assim como na sua documentação. Um tesauro é um SOC que classifica e estrutura conceitos em formas hierárquica, equivalente ou associativa. A delimitação das relações entre os termos indexados (requisitos, neste caso) reduz ambiguidades, melhora a completude e a consistência das informações e, portanto, aumenta a precisão dos mecanismos de recuperação (Altounian; Gomes, 2016).

Isotani *et al.* (2015) afirmam que o uso extensivo de tecnologias da *Web Semântica*, como ontologias, pode realizar interseções com essa área da Ciência da Informação, visto que também permitem a aquisição, a especificação e a representação formal de um determinado conhecimento.

A *Web Semântica*, apresentada por Berners-Lee, Hendler e Lassila (2001), destaca-se por utilizar a *Web* não apenas para compartilhar informação, mas, sobretudo, por atrelar significado a ela. Ao associar aos documentos publicados um conjunto de declarações estruturadas e inteligíveis por máquinas (chamadas de metadados), seus conteúdos podem ser processados, e agentes de *software* podem auxiliar nos processos de tomada de decisão.

Estendendo esse contexto de forma mais pragmática, Berners-Lee *et al.* (2006) introduzem o conceito de *Linked Data* (Dados Ligados) para permitir que, além da publicação dos metadados, seja possível interligar dados por meio de *links* semânticos (relacionamentos formais), construindo a chamada *Web* de Dados, uma evolução da *Web* de documentos original. Nessa nova *Web*, toda a informação pode ser compreendida, tanto por humanos quanto por agentes de *software*, com estes contribuindo para a sua recuperação e para a descoberta de conhecimento (Heath; Bizer, 2011).

Nesse contexto, a união de SOC com tecnologias da *Web Semântica* pode beneficiar as atividades de documentação de requisitos para reúso em projetos. Uma possibilidade que surge é a criação de tesauros de requisitos com anotação semântica. Para isso, essa estrutura deve então ser representada por meio do modelo de dados padrão da *Web Semântica*, que é semiestruturado, baseado em grafos e chamado *Resource Description Framework* (RDF). Ele reusaria ontologias consagradas como a *Simple Knowledge Organization System* (SKOS), um padrão *de facto* para construção de tesauros semânticos (Manola; Miller, 2004; Isaac; Summers, 2009).

Na *Web Semântica*, graças ao caráter estruturado do modelo RDF, os mecanismos de busca, de recuperação e de compartilhamento de informação se tornam bem mais precisos e inteligentes, sendo que a informação pode ser interpretada tanto por máquinas quanto por humanos, trazendo mais refinamento e eficácia para os SOC (Gabriel Junior; Laipelt, 2017). Assim, este trabalho procura responder à seguinte questão de pesquisa: como criar tesauros semânticos de requisitos, reusando ontologias consagradas da *Web Semântica*, de modo que esses tesauros possam ser publicados para reúso, de acordo com os princípios *Linked Data*?

A partir disso, esta pesquisa tem o objetivo de propor um método para representação sistemática de requisitos de um sistema por meio de conceitos em tesauros SKOS que, por sua vez, são definidos e interconectados com as propriedades dessa ontologia, trazendo maior compreensão, graças à semântica explícita. Com isso, procura-se trazer benefícios para a atividade de recuperação de requisitos para reúso.

Para tanto, foi empregada uma metodologia de pesquisa qualitativa, aplicada e experimental, culminando na avaliação/validação do método proposto por meio do desenvolvimento de um protótipo completo (base de dados semântica, *Web service* semântico e aplicação cliente) e de um estudo de viabilidade, baseado em questionário aplicado a profissionais da área de tecnologia.

Este artigo está estruturado da seguinte forma: primeiro são abordados fundamentos relacionados à Engenharia de Requisitos, à *Web Semântica*, ao conceito de *Linked Data*, a tesauros e à ontologia SKOS; a seguir, é descrito o método proposto para a construção de tesauros de requisitos; depois, são discutidos os resultados do estudo de viabilidade realizado com o protótipo; e, por fim, são comentadas as considerações finais e os trabalhos futuros.

Fundamentos

Engenharia de Requisitos

A Engenharia de Requisitos (ER) comprehende um conjunto de atividades, técnicas, métodos e práticas que auxiliam em atividades relativas ao desenvolvimento, à documentação, ao gerenciamento e à manutenção do conjunto de requisitos de um sistema (Sommerville, 2011). De acordo com Pressman (2011), a ER fornece mecanismos apropriados para entender as necessidades do cliente, verificando a viabilidade do projeto e negociando a melhor solução para o que o cliente deseja.

Segundo Sommerville (2011), os requisitos são frequentemente classificados em funcionais e em não funcionais. Os Requisitos Funcionais (RF) descrevem as funcionalidades ou os serviços que o sistema deve oferecer. São descritos de acordo com o ponto de vista do cliente em relação às regras de seu negócio. Exemplos: cadastrar produtos, listar clientes, emitir relatórios de compras *etc.* Já os Requisitos Não Funcionais (RNF) referem-se às especificações técnicas do sistema para qualificar os requisitos funcionais em resposta ao cliente. Exemplos: usabilidade, desempenho, confiabilidade, segurança, entre outros.

Como já mencionado, a descrição e a especificação de requisitos não são tarefas triviais, tanto para os RF como para os RNF. Em relação aos RF, observa-se que eles podem ser descritos em linguagem natural e suplementados com diagramas no documento de requisitos (Sommerville, 2011). No caso dos RNF, são exigidos conhecimentos específicos em diversas áreas, como confiabilidade, desempenho, eficiência, portabilidade e usabilidade, e podem ser utilizados catálogos, *checklists* ou *templates*.

O uso de padrões, de *templates* ou de formulários para descrever requisitos representa a chamada especificação estruturada de requisitos (Sommerville, 2011). Esses documentos guiam o engenheiro ou o analista de requisitos durante o detalhamento das informações sobre os objetos manipulados, as funções desempenhadas e os eventos processados pelo sistema. Alguns exemplos dessa prática são apresentados por Juristo *et al.* (2007), Barcelos (2016) e Silva *et al.* (2016).

Web Semântica e Linked Data

Berners-Lee, Hendler e Lassila (2001) definiram a *Web Semântica* como uma extensão da *Web* original, na qual (meta)dados estruturados são disponibilizados de forma que possam ser processados automaticamente por máquinas, sem interposição de seres humanos. Dessa forma, esses dados e essas informações podem ser acessados e interpretados tanto por humanos quanto por máquinas.

Para possibilitar essa distribuição de dados na *Web*, é necessário um padrão de estruturação, especificação e interligação entre as instâncias, por meio da infraestrutura já disponibilizada pela *Web* convencional. Esse padrão é o modelo de dados *Resource Description Framework*, o qual se trata de um grafo que permite realizar declarações e estabelecer relacionamentos entre os dados disponíveis na *Web* (Jacyntho; Schwabe, 2016).

Manola e Miller (2004) relatam que um grafo RDF é um conjunto de triplas (recurso-propriedade-valor). Em uma tripla RDF, recurso é o nó de origem e representa qualquer “coisa” identificada por um endereço *Web* (*Uniform Resource Identifier*, URI); propriedade é a aresta, sendo uma propriedade de alguma ontologia, também identificada por um URI; e valor é um literal ou outro recurso (URI). Por exemplo, suponha que João e Maria são amigos. Logo, poder-se-ia afirmar que “João conhece Maria”, criando a seguinte tripla RDF:

```
<http://www.example.com/pessoas/joao>
<http://xmlns.com/foaf/0.1/knows>
<http://www.example.com/pessoas/maria>.
```

Nesse exemplo, o nó de origem é o URI que identifica “João”, a aresta é o URI da propriedade “conhece” da ontologia para descrição de pessoas *Friend of a Friend* (FOAF) (Brickley; Miller, 2014) e o nó de destino é o URI que identifica “Maria”.

Assim como no modelo de dados relacional tradicional, associado ao modelo de dados RDF, existe uma linguagem de consulta padrão chamada SPARQL (acrônimo recursivo para *SPARQL Protocol and RDF Query Language*) (Harris; Seaborne, 2013), que nos permite obter, de forma precisa, os dados em um grafo RDF.

A prática de publicar e de estabelecer relacionamentos entre os recursos disponíveis na *Web*, com o objetivo de promover reúso, é conhecida como *Linked Data*. Marcondes (2012) ressalta que a proposta de interligar dados abertos por meio de *links* semânticos oferece grande potencial. Ao contrário dos *links* convencionais, os quais funcionam como simples elos de navegação para humanos, os *links* semânticos, que são propriedades de ontologias,

estabelecem relacionamentos com significado formal explícito, legível, pois, por computadores, o que enriquece o processamento de busca, interpretação e reutilização de conhecimento.

Berners-Lee *et al.* (2006) descrevem quatro princípios para a publicação de dados ligados na *Web*, conhecido como princípios *Linked Data*, a saber: (1) Utilização de URI para nomear recursos; (2) Utilização de URI *Hypertext Transfer Protocol* (HTTP) para que pessoas e máquinas possam acessar esses recursos na *Web*; (3) Uso de URLs dereferenciáveis, ou seja, assim que um URI for acessado, deve-se retornar informação útil em RDF; (4) Inclusão de *links* para outros URI, para que pessoas e agentes de *software* possam navegar para outros recursos e obter mais informações.

Para orientar e encorajar as pessoas a produzir e a publicar dados abertos interligados, Berners-Lee (2009) elaborou um modelo para classificação da maturidade de dados abertos em cinco níveis (estrelas): ★ Disponível na *Web* (em qualquer formato, *e.g.*: PDF), mas com uma licença aberta, para serem dados abertos; ★★ O nível anterior, com dados estruturados inteligíveis por máquina (*e.g.*: uma planilha *Excel* em vez de uma imagem digitalizada de uma tabela); ★★★ Todos os níveis anteriores aliados a um formato não proprietário (*e.g.*: CSV², em vez de *Excel*); ★★★★ Todos os níveis anteriores somados ao uso dos padrões do W3C, *World Wide Web Consortium* (RDF e SPARQL) para descrever os recursos (entidades), de forma que seja possível referenciá-los para reúso; ★★★★★ Todos os outros níveis, além da ligação dos dados publicados a dados de outras pessoas para prover contexto (*Linked Data mashup*³).

Tesauros

Para Mendes *et al.* (2015), um tesauro pode ser definido como uma lista estruturada de termos associados entre si de forma sintática e semântica, baseada na análise de conceitos, por meio de aspectos hierárquicos de generalização e de especialização do significado dos termos. Basicamente, um tesauro é composto por uma taxonomia (hierarquia) de conceitos (termos), em que cada um deles é detalhadamente descrito. Essa hierarquia de conceitos é utilizada para classificar (indexar) artefatos de interesse, como livros, produtos, páginas *Web*, entre outros.

Os termos de um tesauro podem ser interconectados por meio de relações de hierarquia (supertipo e subtipo), de equivalência (sinônímia) ou de associação (Mendes *et al.*, 2015; Altounian; Gomes, 2016). Para exemplificar, a seguir é apresentada a hierarquia de um pequeno tesauro sobre esportes (RT: *Related Term*; BT: *Broader Term*; NT: *Narrower Term*).

- "Voleibol"
 - RT "Quadra"
 - RT "Rede"
 - BT "Esporte"
- "Esporte"
 - BT "Atividade"
 - NT "Tênis"
 - NT "Futebol"
 - NT "Voleibol"

A construção de tesauros é um procedimento sistematizado e bem definido. Segundo Maculan (2015), as metodologias para a construção de tesauros apresentam um esquema composto por três etapas: (I) inicial: planejamento geral do tesauro, incluindo delimitação do domínio a ser modelado e definição dos objetivos, tema, público-alvo e principais fontes de terminologia; (II) desenvolvimento: estrutura conceitual, critérios de modelagem, organização de descritores e estabelecimento de relações semânticas; e (III) edição: montagem da estrutura de relações, escolha de forma de apresentação e de simbologia das relações, e seleção de *software* para gerenciamento das etapas de construção do tesauro.

²Comma Separated Values (CSV): uma implementação de arquivos de texto tabulados que são separados por delimitadores que, neste caso, são vírgulas.

³Interligação (reúso) de dados de terceiros a fim de enriquecer os próprios dados.

Ontologia SKOS

No contexto da informática e da Ciência da Informação, uma ontologia é definida como um modelo constituído de termos organizados de forma taxonômica, com definições e axiomas utilizados para representação formal de uma área de conhecimento específica. Segundo Dermeval *et al.* (2016), esse conjunto de conceitos é dotado de classes, propriedades (atributos e relacionamentos), incluindo os significados e as restrições. Na *Web Semântica*, ontologias são especificadas em linguagens padrão de criação de ontologias (metaontologias), como *RDF Schema* (*RDFS*) (Bickley *et al.*, 2014) e *Web Ontology Language* (*OWL*) (Horrocks *et al.*, 2012).

Segundo Ramalho (2015), *Simple Knowledge Organization System* (*SKOS*) é uma ontologia escrita em *OWL* que permite representar taxonomias, tesouros, esquemas de classificação, glossários, folksonomias e outros esquemas de organização de conhecimento. Em *SKOS*, os conceitos (termos) do tesouro podem ser organizados de forma simples e/ou hierárquica em triplas *RDF*, a partir dos seguintes elementos: Conceitos: podem ser definidos como uma invenção da mente humana para “encapsular” fragmentos da realidade. São os elementos-chave da estrutura e são identificados por *URI*; Propriedades: têm a função de descrever os atributos dos conceitos, a partir do objetivo do esquema conceitual. As propriedades podem ser de etiquetagem ou de documentação; Relacionamentos: têm a função de imputar a semântica aos esquemas a partir da categorização e da hierarquia dos conceitos. As relações podem ser de hierarquia, de associação ou de equivalência.

A ontologia *SKOS* é uma das mais utilizadas no movimento *Linked Open Data*. Além disso, quando houve a publicação da ISO 25964 (padrão internacional para tesouros e interoperabilidade com outros vocabulários) pela *National Information Standards Organization* (Organização Nacional de Normas de Informação), um dos principais esforços foi alcançar a meta de interoperabilidade com o *SKOS* e outros esquemas (Blumauer, 2014).

Método proposto para criação de tesouros semânticos de requisitos

Nesta seção, é apresentado o método proposto para a criação de tesouros semânticos de requisitos, incorporando os princípios da *Web Semântica* e do *Linked Data* para organizar e para documentar, formalmente, o conhecimento durante a especificação de requisitos, aprimorando, pois, a sua disponibilização, a sua recuperação e o seu reúso. Para cada um dos passos, a trajetória metodológica percorrida para sua definição utilizou como ponto de partida o plano de construção de tesouros de Maculan (2015). Complementando a explanação do método, a seção subsequente a esta é reservada à descrição da aplicação do método proposto em um estudo de viabilidade com profissionais da área de Tecnologia da Informação (TI), visando a corroborar sua eficácia.

Planejamento das informações do tesouro e definição do template de descrição de requisito

Definir as informações relacionadas a requisitos que serão publicadas

O primeiro passo é definir, com base na literatura e na experiência dos engenheiros de requisitos adquirida em projetos passados, quais informações são pertinentes à publicação de requisitos na forma de tesouros, visando ao registro perene e ao reúso por outros projetos.

Elaborar um template (modelo) de descrição de requisito para utilizar, de forma sistemática, na definição do tesouro

Todo tesouro é estruturado por meio de uma hierarquia ou de uma taxonomia de seus conceitos (termos). No método proposto, como se trata de um tesouro de requisitos, os conceitos do tesouro são os próprios requisitos em si. Em outras palavras, o método preconiza que cada requisito seja definido como um conceito no tesouro.

Além da hierarquia, em um tesouro, cada conceito, em separado, deve ser descrito, detalhadamente, de forma estruturada. Portanto, das informações pertinentes anteriormente levantadas, é necessário definir um *template*

(modelo) estruturado para descrição de cada requisito que compõe a hierarquia de requisitos do tesouro. Deve-se ser considerado que quanto maior o nível de detalhamento das informações relacionadas a um requisito, maior a garantia de seu entendimento, seja por parte do usuário, seja por parte da equipe de desenvolvimento.

Ao optar por um modo de especificação estruturada através do uso de *templates*, é facilitada a identificação de quais dados/informações devem ser representados e organizados, assim como seu mapeamento ontológico para a construção dos tesauros. Para o *template* de descrição de cada requisito, o método sugere uma estrutura híbrida e adaptada ao contexto da *Web Semântica*, tendo como bases principais os modelos de especificação de RF e de RNF propostos por Barcelos (2016) e Silva *et al.* (2016), contendo os seguintes campos:

Identificador: URI do requisito. O requisito é representado por um recurso RDF e, por definição, deve ser identificado por um endereço *Web* (URI). Usar um identificador sugestivo para o requisito que remeta a seu propósito.

Exemplo: <http://example.com/requirements/comprarProdutoOnline>

Nome preferencial: nome preferencial do requisito.

Exemplo: *Comprar Produto Online*

Nome(s) alternativo(s): outro(s) nome(s) que podem ser utilizados para se referir ao mesmo requisito.

Exemplo: *Comprar Mercadoria Online*

Idioma: definir o idioma em que o requisito está escrito.

Exemplo: pt-br.

Problema: descrever a situação ou a necessidade do cliente que deu origem ao requisito.

Exemplo: Permitir que o cliente escolha um ou mais produtos e compre-os.

Contexto: descrever o ambiente em que o problema foi constatado.

Exemplo: Compras on-line pelo *site* de comércio eletrônico da empresa.

Definição: definir o requisito usando algum modelo padrão de escrita de requisito, incluindo os parâmetros obrigatórios (delimitados pelos símbolos <>) e opcionais (delimitados pelos símbolos []).

Exemplo de modelo para RF: O <tipo de usuário ou nome do ator> deve ser capaz de <realizar algo> <ao objeto> [sob alguma condição].

Exemplo de definição: O <cliente> deve ser capaz de <comprar on-line> <produto> [em estoque].

Exemplo de modelo para RNF: O sistema deve ser capaz de garantir <atributo de qualidade> [sob alguma condição].

Exemplo de definição: O sistema deve ser capaz de garantir <disponibilidade 24 horas, 7 dias por semana>.

Exemplo: demonstração de uso do requisito.

Exemplo RF: <João> deve ser capaz de <comprar on-line> <o livro "O Pequeno Príncipe"> [em estoque].

Domínio(s): indicar o(s) domínio(s) de conhecimento (ou de negócio) ao(s) qual(is) o requisito está vinculado.

Exemplo: <http://example.com/requirements/vendas>

Tipo(s) de Sistema: indicar a qual(is) tipos de sistema o requisito está relacionado.

Exemplo: <http://example.com/domain/comercioEletronico>

Tipo(s) de Requisito: indicar se o requisito é classificado como funcional ou como uma (ou mais) das subcategorias de não funcional (conforme ISO/IEC-25010).

Exemplo: <http://example.com/requirementsType/funcional>

Requisito(s) mais geral(is): indicar o(s) requisito(s) pai, ou seja, aquele(s) que está(ão) diretamente acima deste na hierarquia de requisitos do tesauro.

Exemplo: <http://example.com/requirement/comprarProduto>

Requisito(s) mais específico(s): indicar o(s) requisito(s) filho, ou seja, aquele(s) que está(ão) diretamente abaixo deste na hierarquia de requisitos do tesauro.

Exemplo: <http://example.com/requirement/comprarProdutoPerecívelOnline>

Requisito(s) relacionado(s): indicar o(s) requisito(s) relacionado(s) a este de alguma outra forma que não seja hierárquica.

Exemplo: <http://example.com/requirement/devolverProdutoOnline>

Observação: observação em geral sobre o requisito.

Exemplo: *Somente produtos em estoque podem ser adquiridos online. Não é permitido fazer encomendas online.*

A escolha pelo padrão de especificação estruturado de Barcelos (2016) é feita com base no fato de ser uma proposta diretamente voltada para documentação de RF de sistemas de informação para reúso. Além disso, a proposta foi validada por meio da execução de estudos de casos com estudantes de diferentes áreas, como Ciência da Computação, Engenharia Elétrica e Sistema de Informação. Desse modelo, foram reutilizados no *template* desta pesquisa, os campos: nome (aqui dividido em nome preferencial e nome(s) alternativo(s)), problema, definição, exemplo, domínio(s) e tipo(s) de requisito.

Em relação ao modelo de especificação estruturada para RNF proposto por Silva *et al.* (2016), a escolha se deve ao fato de sua criação ter passado por um processo bem sistematizado e definido, com uma extensa pesquisa bibliográfica e com o preenchimento de questionários sobre necessidades com especialistas na área de requisitos, os quais também participaram da sua validação. Desse modelo, foram reutilizados no *template* desta pesquisa os campos: identificador, contexto, definição, exemplo e tipo(s) de requisito.

Como contribuição, esta pesquisa sugere a utilização dos campos: idioma, tipo(s) de sistema, observação, requisito(s) mais geral(is), requisito(s) mais específico(s) e requisito(s) relacionado(s). Esses três últimos têm o objetivo de agregar à especificação dados/informações que representam características para construção de tesouros.

É importante ressaltar que essa estrutura apresentada é uma sugestão de *template* para inserção das informações nos tesouros. Não obstante, nada impede que este *template* seja adaptado, estendido ou simplificado, de acordo com as necessidades de detalhamento de informações de projeto ou com o formato de entrada dos dados.

Estruturação do tesauro: mapeamento do template de descrição de requisito em ontologias Linked Data

Mapear cada campo do template em propriedades de ontologias Linked Data consagradas, sobretudo a ontologia SKOS, voltada justamente para a criação de tesouros

Como dito anteriormente, cada requisito é um conceito do tesauro, logo, torna-se um conceito SKOS, ou seja, cada requisito é uma instância da classe *skos:Concept* e deve ser relacionado a esta classe pela propriedade *rdf:type* do vocabulário padrão RDF. A seguir, devem ser estabelecidas as relações hierárquicas e associativas entre esses conceitos. O Quadro 1 apresenta as relações e as propriedades do vocabulário SKOS e os respectivos mapeamentos dos campos do *template* de especificação de requisitos. Como o vocabulário SKOS não prevê propriedades para o campo Idioma do *template*, no Quadro 1 são apresentadas duas propriedades extras para esse propósito, uma da ontologia *Dublin Core* (prefixo *dcterms*) e outra da ontologia *Schema.org* (prefixo *schema*) (*Schema.org Community Group, 2011; Dublin Core Metadata Initiative, 2012*). Além dessas duas propriedades, o idioma também pode ser especificado, *in loco*, no valor literal de cada propriedade com "@" e com o código do idioma (e.g.: "text@en" ou "texto@pt") (Quadro 1).

Quadro 1. Relações e propriedades de ontologias utilizadas para o mapeamento das informações do *template*.

Tipo de Propriedade	Propriedade	Definição na ontologia	Mapeamento do Template
Relações Hierárquicas	<i>skos:broader</i>	Relaciona o conceito com um conceito mais geral.	Domínio(s) Tipo(s) de Sistema Tipo(s) de Requisito Requisito(s) mais Geral(is)
	<i>skos:narrower</i>	Relaciona o conceito com um conceito mais específico.	Requisito(s) mais específico(s)
Relações Associativas	<i>skos:related</i>	Define qualquer relação associativa não-hierárquica entre dois conceitos.	Requisito(s) relacionado(s)
Propriedades de Etiquetagem	<i>skos:prefLabel</i>	Subpropriedade de <i>rdfs:label</i> que define o termo (nome) preferencial de um conceito, em um determinado idioma.	Nome Preferencial
	<i>skos:altLabel</i>	Subpropriedade de <i>rdfs:label</i> que define termos (nomes) alternativos de um conceito, como sinônimos, acrônimos, abreviações, variações de ortografia, e formas de plural/singular.	Nome(s) Alternativo(s)
Propriedades de Documentação	<i>skos:note</i>	Nota geral sobre o conceito	Observação
	<i>skos:definition</i>	Subpropriedade de <i>skos:note</i> que apresenta uma explicação formal completa do significado de um conceito.	Definição
	<i>skos:example</i>	Subpropriedade de <i>skos:note</i> que fornece exemplo de uso de um conceito.	Exemplo
	<i>skos:scopeNote</i>	Subpropriedade de <i>skos:note</i> que indica o escopo (contexto) onde o conceito pode ser usado.	Problema Contexto
Propriedades de Idioma	<i>dcterms:language</i>	Idioma ou língua do conteúdo.	Idioma
	<i>schema:inLanguage</i>		

Fonte: Elaborado pelos autores (2020).

O uso dessas relações e propriedades SKOS deve ser planejado de acordo com semântica que se deseja estabelecer entre dois requisitos, a partir do pensamento em triplas RDF (recurso-propriedade-valor). Nesse planejamento, devem ser considerados também os princípios *Linked Data*, gerando informações úteis por meio das relações, assim como a criação de URI únicos para cada requisito, sendo estes diretamente ligados a URI de outros requisitos, permitindo a naveabilidade na estrutura do tesauro.

Caso seja utilizado outro *template* de especificação de requisitos, o raciocínio continua o mesmo, bastando apenas fazer uma simples adaptação no mapeamento entre os elementos do *template* e as correspondentes propriedades da ontologia SKOS.

Cadastro e publicação do tesauro

Criar e publicar o tesauro de requisitos por meio de uma fonte de dados RDF que permita realizar consultas SPARQL ad hoc para precisamente acessar os requisitos e reusá-los como quiser

As triplas RDF que descrevem os requisitos devem ser cadastradas (ou geradas automaticamente de alguma forma) e armazenadas em uma base de conhecimento semântica de requisitos, um banco de dados RDF (*triple store*). Por padrão, esse banco oferece um *SPARQL endpoint*, ou seja, um *Web service* que permite executar consultas SPARQL *ad hoc* sobre o tesauro, obtendo precisamente a informação que se queira, como a rastreabilidade entre requisitos ou entre qualquer outro artefato e um ou mais requisitos.

Além disso, seguindo o terceiro princípio *Linked Data*, os URI dos requisitos devem ser dereferenciáveis, ou seja, ao serem acessados, esses URI devem retornar um arquivo descrevendo o requisito. Por meio de negociação de conteúdo HTTP (Fielding *et al.*, 1999), é retornado ou um arquivo RDF para máquinas ou um arquivo HTML para humanos, ao acessar o mesmo URI. Com relação ao quarto princípio *Linked Data*, sempre que possível, é preciso enriquecer a descrição dos requisitos, relacionando-os com recursos (URI) de outras fontes de dados consagradas da Web Semântica (e.g.: DBpedia), perfazendo o *Linked Data mashup*, com o objetivo de alcançar as cinco estrelas propostas pelo modelo de Berners-Lee (2009).

A base de conhecimento deve ser extensível para permitir que novos conceitos, relações ou propriedades sejam inseridos ou modificados a qualquer momento. Conforme a essência da Web, para referenciar um requisito, a partir de qualquer lugar do mundo, seja para estendê-lo ou para (re)usá-lo, basta estabelecer um *link* com seu respectivo URI, formando uma tripla RDF.

Uma aplicação com interface de usuário amigável (ou um *plug-in* para uma ferramenta existente) deve ser criada para dar suporte ao cadastro e uma definição das relações entre os requisitos, de modo que o grafo RDF subjacente seja automaticamente criado e armazenado no banco de dados. Ao inserir um novo requisito, com base nos axiomas da ontologia SKOS, o módulo raciocinador (*reasoner*) do banco RDF infere automaticamente mais informações, contribuindo para que o tesouro se torne cada vez mais robusto e completo.

Aplicação do método em um estudo de viabilidade

Esta seção descreve a aplicação do método proposto em um estudo de viabilidade, com a especificação de requisitos utilizando o *template* sugerido, e o desenvolvimento de um protótipo de *Web service* – uma camada de serviços sobre a fonte de dados RDF para facilitar o acesso ao tesouro.

Em seguida, é criada uma aplicação cliente para interagir com os recursos do *Web service* e, assim, submeter o método à apreciação de profissionais de TI para atestar sua viabilidade. Trata-se de uma explanação em alto nível de abstração, com as triplas RDF mostradas de forma gráfica para facilitar a compreensão.

Para os exemplos de estruturas semânticas apresentadas ao longo desta seção, são considerados os seguintes prefixos (*namespaces*) para abreviar os respectivos URI:

```
prefixo rt: <http://localhost:8080/requirementsThesauri/> (host local)
prefixo rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
prefixo rdfs: <http://www.w3.org/2000/01/rdf-schema#>
prefixo skos: <http://www.w3.org/2004/02/skos/core#>
prefixo owl: <http://www.w3.org/2002/07/owl#>
prefixo dbpedia: <http://dbpedia.org/resource/>
prefixo schema: <http://schema.org/>
prefixo dcterms: <http://purl.org/dc/terms/>
```

Para este exemplo de uso do método, é considerado o seguinte requisito não funcional realista: “Confidencialidade de Dados de Servidores”. O vocabulário SKOS é a ontologia principal utilizada na construção das taxonomias, por isso, há destaque nos grafos RDF para os relacionamentos que utilizam suas propriedades, baseadas no Quadro 1 do método. Cabe ressaltar que a construção dos grafos não está restrita à ontologia SKOS, pois, por definição, na Web Semântica, quaisquer outras ontologias complementares também podem ser usadas, agora ou futuramente, como neste exemplo, em que foram usadas propriedades das ontologias RDF Schema (Brickley *et al.*, 2014), e OWL.

A Figura 1 apresenta a composição do grafo RDF do requisito em questão utilizando as propriedades SKOS, seguindo a etapa (2) do método. Cada tripla descreve o requisito cujo URL é [rt:requirements/confidencialidadeDeDadosDeServidores](#).

Como mencionado no *template* do método, um requisito pode pertencer a mais de uma hierarquia, surgindo, assim, tesauros ou subtesauros de apoio ao tesauro principal de requisitos. A Figura 2 apresenta o tesauro de apoio “Tipos de Domínio de Conhecimento/Negócio” dos sistemas (outros domínios podem ser facilmente cadastrados na base de conhecimento RDF). Assim como os requisitos, o conceito *rt:domains* e todos os outros conceitos são instâncias (propriedade *rdf:type*) da classe *skos:Concept*, mas essas triplas foram omitidas, na Figura 2, por razões de espaço. As estruturas dos outros tesauros de apoio (“Tipos de Requisito” e “Tipos de Sistema”) possuem formato similar.

A propriedade *skos:narrower* indica um conceito mais específico, e a propriedade inversa *skos:broader* indica um conceito mais geral. Com a propriedade *owl:sameAs* é estabelecida uma ligação com o recurso externo “educação” da DBpedia, perfazendo o *Linked Data mashup*, permitindo que o usuário possa consultar mais informações sobre esse domínio (reúso de informações).

Sob o ponto de vista hierárquico, na Figura 3, o requisito “Confidencialidade de Dados de Servidores” é um subconceito do conceito de domínio “Gestão de Servidores”, bem como do tipo de requisito “Confidencialidade”. Por exemplo, na ontologia SKOS, a propriedade *skos:broader* é definida como subpropriedade da propriedade *skos:broaderTransitive*, que, por sua vez, é uma propriedade transitiva. Com isso, ao relacionar, por meio da propriedade *skos:broader*, esse requisito ao tipo de requisito “Confidencialidade”, a máquina infere automaticamente que o requisito também é do tipo “Segurança” e “Não Funcional”, relacionando-o diretamente com esses supertipos, por meio da propriedade *skos:broaderTransitive*. Assim, configura-se uma rastreabilidade via inferência ontológica (esta vale para todos os relacionamentos *skos:narrower* e *skos:broader*).

Para obter, precisamente, os requisitos no tesauro, basta executar consultas SPARQL sobre a base de conhecimento de requisitos, utilizando, para isso, bibliotecas RDF, e.g.: Jena (Apache Software Foundation, 2011). A consulta retorna os URI dos requisitos de interesse. De posse desses URI, esses requisitos podem ser facilmente referenciados, acessados e reusados.

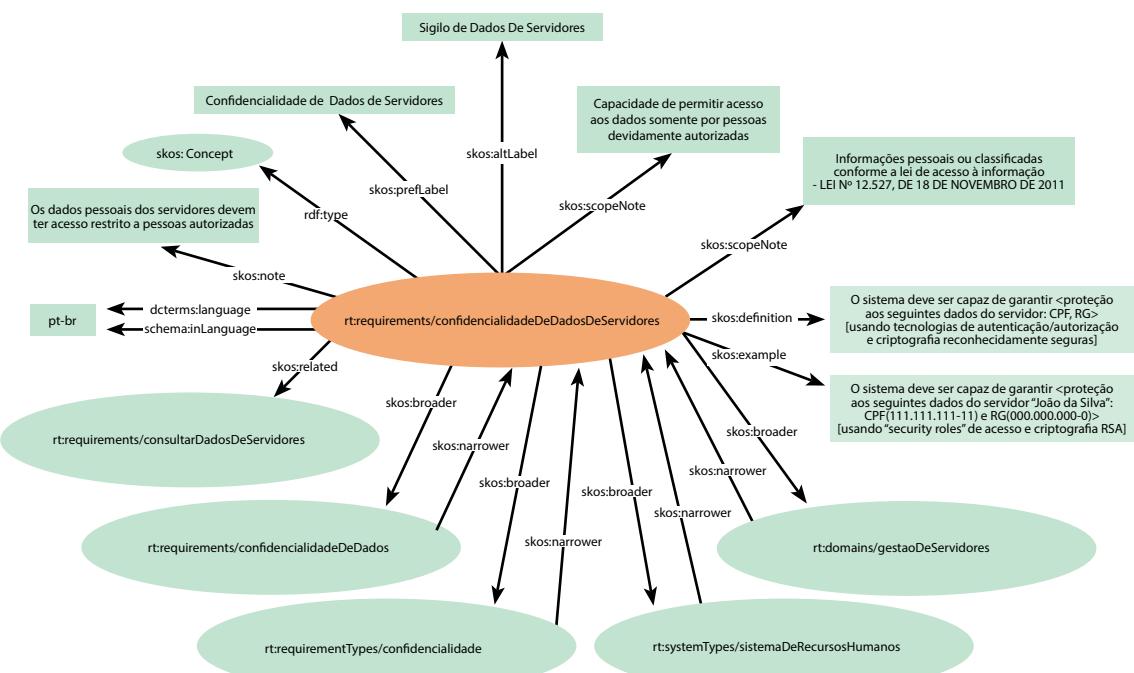


Figura 1. Grafo RDF com as propriedades de etiquetagem e de documentação do requisito “Confidencialidade de Dados de Servidores”.

Fonte: Elaborado pelos autores (2020).

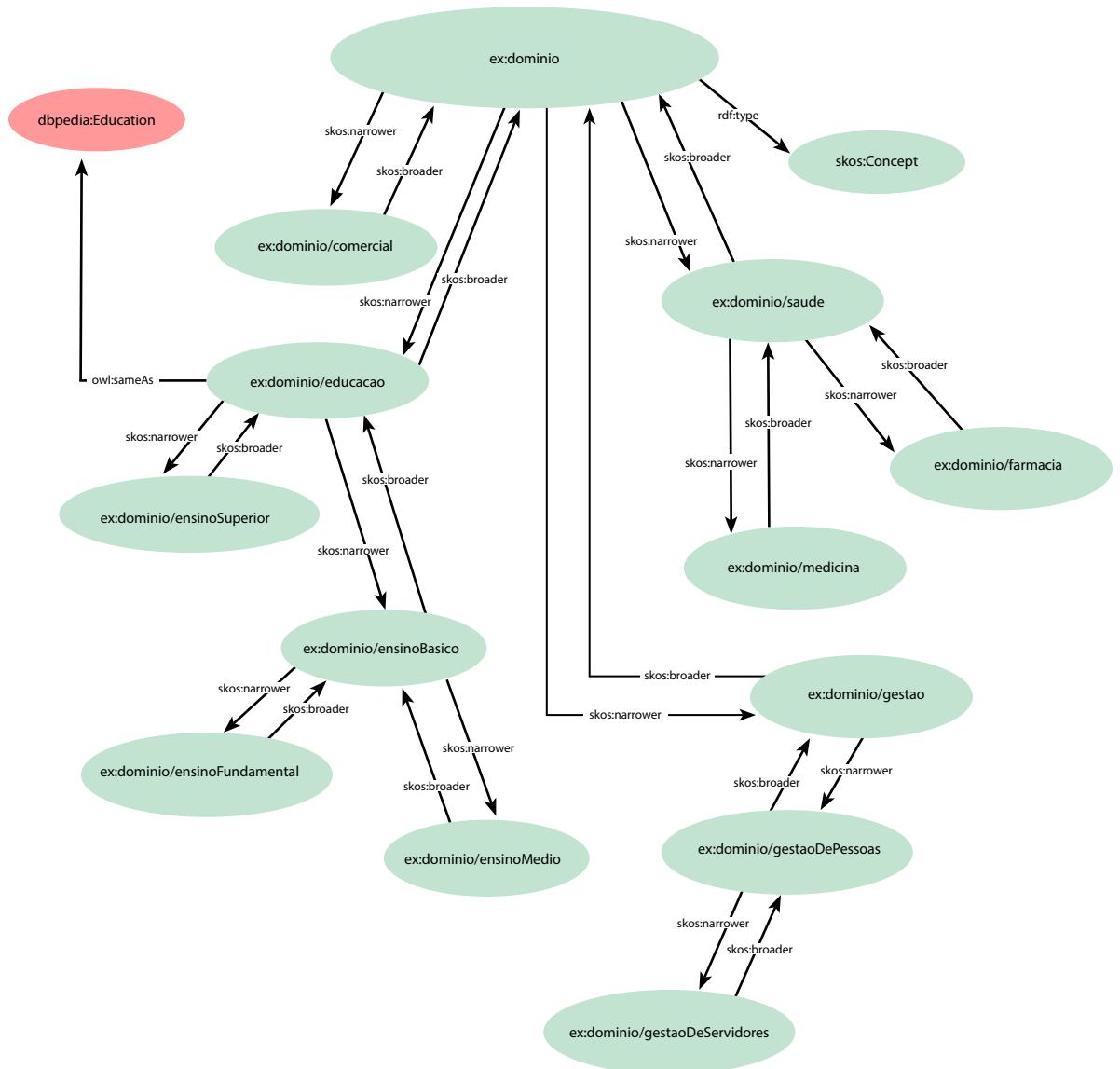


Figura 2. Grafo RDF do tesouro de Domínios de Conhecimento/Negócio.

Fonte: Elaborado pelos autores (2020).

Visando ao cadastro e à publicação do tesouro previsto na etapa (3) do método, foi criado um protótipo de *Web service* RESTful conectado a um banco de dados nativo RDF. A decisão por um *Web service* se deve à possibilidade de compartilhamento de acesso com diversos dispositivos clientes (computadores, celulares, tablets etc.) e à simplicidade de implementação dos métodos para manipulação dos recursos com a biblioteca *Jena*.

Os dispositivos clientes acessam o *Web service* por meio de uma requisição HTTP, endereçada a um URI que identifica um recurso na base de dados RDF do tesouro de requisitos. Essa requisição representa uma ação a ser executada sobre o referido recurso, de acordo com o método HTTP empregado (*POST*: *criação*; *GET*: *leitura*; *PUT*: *atualização*; *DELETE*: *remoção*). As requisições são processadas pela biblioteca Jena, que acessa a base de dados RDF por meio de consultas SPARQL, gerenciando as inferências com as ontologias e retornando as triplas RDF na sintaxe requisitada pelo cliente, entre as quais: JSON-LD (Sporny *et al.*, 2014), RDF/XML (Gandon; Schreiber, 2014) e Turtle (Beckett *et al.*, 2014).

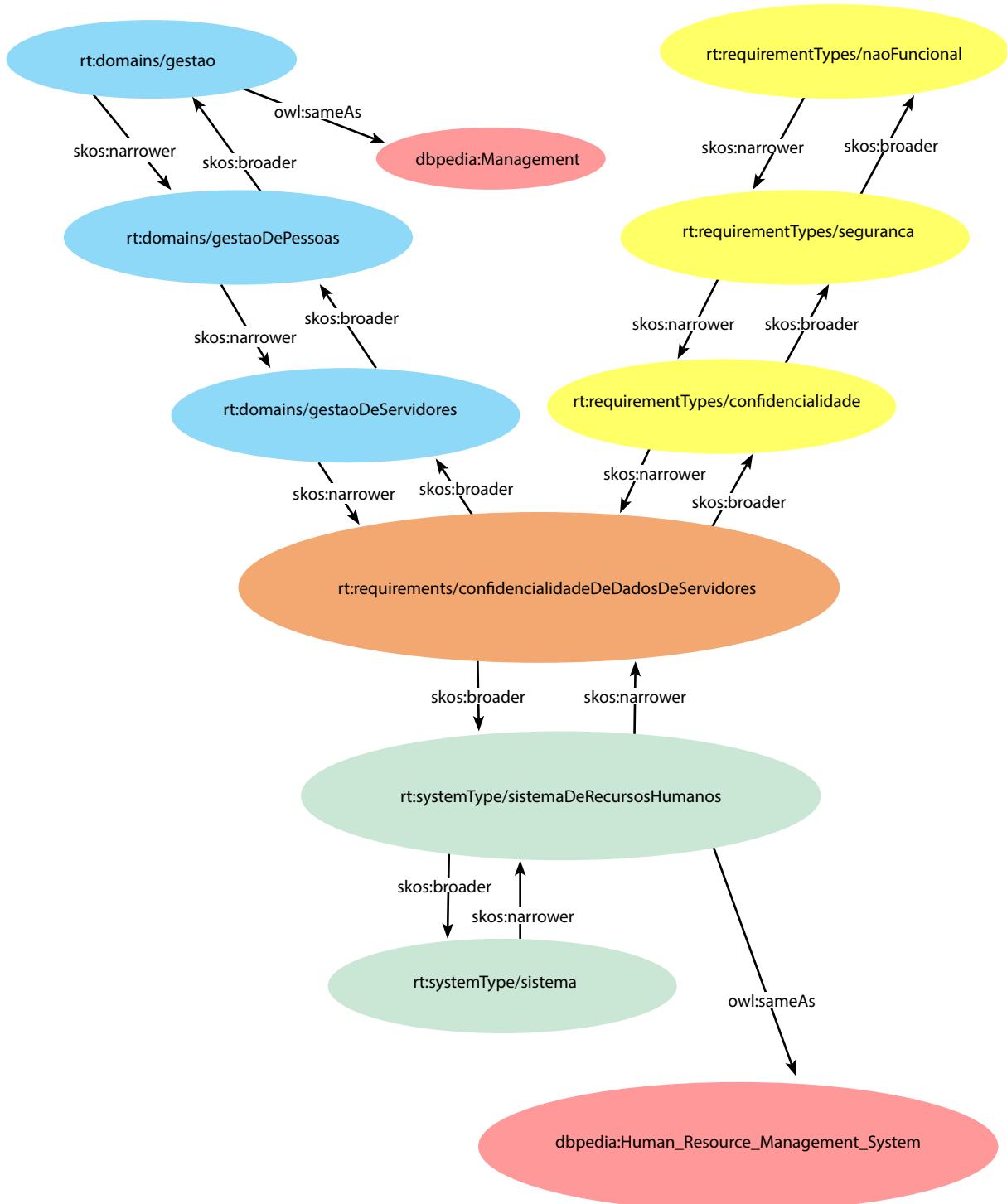


Figura 3. Grafo RDF com relacionamentos hierárquicos do requisito “Confidencialidade de Dados de Servidores” com os tesauros de Domínios (Azul), de Tipos de Requisito (Amarelo) e de Tipos de Sistema (Verde).

Fonte: Elaborado pelos autores (2020).

A Figura 4 exemplifica o uso do *Web service* desenvolvido. Figura 4A apresenta a inserção (método HTTP POST) do recurso correspondente ao requisito “Confidencialidade de Dados de Servidores”, representando os dados de entrada em JSON. Já Figura 4B mostra a posterior recuperação (método HTTP GET) desse mesmo recurso, obtendo sua representação RDF na sintaxe Turtle.

(A) POST Request:

Method: POST
URL: localhost:8080/requirementsThesauri/requirements
Headers: (9)
Body (raw JSON):

```

1  {
2    "requirementID": "confidencialidadeDeDadosDeServidores",
3    "label": "Confidencialidade de Dados de Servidores",
4    "language": "pt-BR",
5    "prefLabel": "Confidencialidade de Dados de Servidores",
6    "altLabel": "Sigilo de Dados de Servidores",
7    "problem": "Capacidade de permitir acesso aos dados somente por pessoas devidamente autorizadas.",
8    "context": "Informações pessoais ou classificadas conforme a lei de acesso à informação - LEI Nº 12.527, DE 18 DE NOVEMBRO DE 2011.",
9    "template": "O sistema deve ser capaz de garantir <proteção aos seguintes dados do servidor: CPF, RG> [usando tecnologias de autenticação /autoriação e criptografia reconhecidamente seguras].",
10   "example": "O sistema deve ser capaz de garantir <proteção aos seguintes dados do servidor \"João da Silva\": CPF (111.111.111-11) e RG (000.000.0-0) [usando \"security roles\" de acesso e criptografia RSA].",
11   "broaderRequirementID": "confidencialidade",
12   "broaderRequirementID": "confidencialidadeDeDados",
13   "broaderDomainID": "gestaoDeServidores",
14   "broaderSystemTypeID": ["SistemaDeRecursosHumanos"],
15   "narrowerRequirementID": [null],
16   "relatedRequirementID": ["consultarDadosDeServidores"],
17   "note": "Os dados pessoais dos servidores devem ter acesso restrito a pessoas autorizadas."
18 }

```

(B) GET Request:

Method: GET
URL: localhost:8080/requirementsThesauri/requirements/confidencialidadeDeDadosDeServidores
Headers: (8)
Query Params:

KEY	VALUE	DESCRIPTION	...	Bulk Edit
Key	Value	Description		

Response:

```

@prefix dcterms: <http://purl.org/dc/terms/> .
@prefix dom: <http://localhost:8080/requirementsThesauri/domains/> .
@prefix rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#> .
@prefix req: <http://localhost:8080/requirementsThesauri/requirements/> .
@prefix rqt: <http://localhost:8080/requirementsThesauri/requirementTypes/> .
@prefix schema: <http://schema.org/> .
@prefix skos: <http://www.w3.org/2004/02/skos/core#> .
@prefix sys: <http://localhost:8080/requirementsThesauri/systemTypes/> .
@prefix xml: <http://www.w3.org/XML/1998/namespace> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

req:confidencialidadeDeDadosDeServidores a skos:Concept ;
  rdfs:label "Confidencialidade de Dados de Servidores" ;
  dcterms:inLanguage "pt-br" ;
  schema:language "pt-br" ;
  skos:altLabel "Sigilo de Dados de Servidores" ;
  skos:broader dom:gestaoDeServidores,
    rqt:confidencialidade,
    sys:sistemaDeRecursosHumanos ;
  skos:definition "O sistema deve ser capaz de garantir <proteção aos seguintes dados do servidor: CPF, RG> [usando tecnologias de autenticação /autoriação e criptografia reconhecidamente seguras].";
  skos:example "O sistema deve ser capaz de garantir <proteção aos seguintes dados do servidor \"João da Silva\": CPF (111.111.111-11) e RG (000.000.0-0) [usando \"security roles\" de acesso e criptografia RSA].";
  skos:note "Os dados pessoais dos servidores devem ter acesso restrito a pessoas autorizadas." ;
  skos:prefLabel "Confidencialidade de Dados de Servidores" ;
  skos:related req:consultarDadosDeServidores ;
  skos:scopeNote "Capacidade de permitir acesso aos dados somente por pessoas devidamente autorizadas." ;
  "Informações pessoais ou classificadas conforme a lei de acesso à informação - LEI Nº 12.527/2011." .

```

Figura 4. (A) Cadastro do novo requisito “Confidencialidade de Dados de Servidores” via requisição HTTP (POST) com dados em JSON. (B) Posterior acesso ao mesmo requisito via HTTP (GET) obtendo o respectivo grafo RDF na sintaxe Turtle.

Fonte: Elaborado pelos autores (2020).

Após o planejamento da estrutura do tesauro e da codificação do *Web service* para que dê suporte a todo processo descrito, foi criada uma aplicação cliente para consumir o *Web service* por meio de uma interface gráfica, contendo um formulário que faz alusão ao *template*. A aplicação também disponibiliza mecanismos para consulta, inserção, remoção, atualização e navegação entre os recursos cadastrados nos tesauros.

A partir disso, foram convidados profissionais de TI para participar do estudo, com objetivo de avaliar o método e seu possível impacto na atividade de reúso de requisitos. O grupo é formado por oito profissionais atuantes nos setores público e privado, com experiência profissional em desenvolvimento de sistemas e de titulação que varia de nível técnico a mestrado.

O estudo foi organizado em duas etapas: apresentação e interação com o protótipo e preenchimento de questionário sobre a experiência. Este apresenta um conjunto com dezessete afirmações baseadas em hipóteses relacionadas à eficácia do método proposto para organização do conhecimento e para a recuperação e o reúso de requisitos, que devem ser avaliadas pelos participantes utilizando a Escala de Likert. Trata-se de um tipo de escala de resposta psicométrica muito usada questionários e, principalmente, em pesquisas de opinião. Ao responderem esta escala, os perguntados especificam seu nível de concordância com relação a uma afirmação (*Concordo Totalmente*, *Concordo Parcialmente*, *Indiferente*, *Discordo Parcialmente* e *Discordo Totalmente*).

Ao analisar as respostas obtidas no questionário, pode-se destacar que a opção “Concordo Totalmente”, a qual corrobora as hipóteses, foi escolhida pela maioria dos participantes em doze das dezessete proposições apresentadas. A escolha da opção “Concordo Parcialmente” também foi frequente na maior parte das questões. Já em relação ao extremo da escala, a opção “Discordo Totalmente” só apareceu quando foi afirmado que os participantes não tinham dificuldades para organizar requisitos para reúso. No fim do questionário, foram feitas perguntas abertas e foi disponibilizado um espaço para considerações.

Como parecer geral documentado durante as duas etapas do estudo, o grupo avalia que o método pode trazer novas perspectivas para área de reúso de requisitos, destacando, principalmente, o poder de inferência da máquina sobre os dados trazidos pelos axiomas das ontologias. O fato de sugerir informações ou recursos extras durante uma consulta pode agilizar o processo de documentação dos requisitos ou também a descoberta de outros requisitos não levantados previamente. Todavia, o grupo também faz ressalvas sobre possíveis dificuldades que possam surgir devido ao fato de alguns desenvolvedores não possuírem familiaridade com os conceitos ou com tecnologias da *Web Semântica* apresentados.

Por razões de espaço, não foi possível descrever com detalhes a arquitetura e as funções do protótipo, assim como todo o relato sobre a condução e resultados do estudo. Essas informações podem ser consultadas na íntegra na dissertação de mestrado de Zacarias (2020).

Resultados e Discussão

Como pode ser observado na descrição da aplicação do método no estudo de viabilidade, o protótipo desenvolvido permite a inserção, a recuperação, a atualização e a remoção de recursos, que representam os requisitos, nos tesauros. Uma vez que estes requisitos são publicados na *Web* e são identificados por um URI globalmente único, acessá-los e reusá-los é bastante direto, bastando acessar o URI. Essa abordagem fomenta, sobremaneira, o reúso de requisitos em escala mundial, graças a toda infraestrutura pré-existente da *Web*, promovida pelo protocolo HTTP.

Além do acesso via URI, outra forma precisa para recuperar um requisito é por meio de consultas SPARQL submetidas ao SPARQL *endpoint* do banco de dados RDF utilizado para armazenar o tesauro de requisitos. Exemplificando, para recuperar todos os requisitos (URI) não funcionais que possuam o termo “servidores” (sem diferenciar maiúsculas e minúsculas) no nome preferencial, e, portanto, dentre eles, o requisito “Confidencialidade de Dados de Servidores”, do estudo de viabilidade apresentado na seção anterior, basta realizar a seguinte consulta SPARQL, com inferência ontológica habilitada no banco de dados RDF:

```

prefix rt: <http://localhost:8080/requirementsThesauri/requirementTypes>
prefix skos: <http://www.w3.org/2004/02/skos/core#>
SELECT ?requirement
WHERE
{
  rt:naoFuncional skos:narrowerTransitive ?requirement .
  ?requirement skos:prefLabel ?name .
  FILTER regex(str(?name), "servidores", "i") .
}

```

É importante destacar que para recuperar um requisito não é necessário ter conhecimento de todas as informações mapeadas previamente. Como pode ser observado na consulta SPARQL apresentada, somente alguns parâmetros foram descritos, como o tipo de requisito – não funcional –, e solicitados dentre eles aqueles que possuíam a palavra “servidores” no nome preferencial.

Com uso da propriedade *skos:narrowerTransitive*, graças à inferência ontológica, realizada, de forma transparente, pelo módulo raciocinador (*reasoner*, em inglês) do banco de dados, foram retornados todos requisitos descendentes (diretos e indiretos) do tipo de requisito “não funcional” e que atendam ao filtro de nome, podendo, inclusive, trazer recursos/requisitos extras que podem influenciar em uma tomada de decisão, durante a construção do documento de requisitos. Tudo isso de forma automática e sem muito esforço por parte do usuário.

Por meio do protótipo de *Web service*, foi possível verificar esses resultados na prática durante o estudo de viabilidade. Os profissionais participantes ressaltaram que a agilidade na recuperação dos requisitos e as possibilidades de obtenção de dados extras por meio da inferência ontológica são pontos de destaque para o método. Os índices de concordância verificados no questionário auxiliam na investigação dos benefícios da estruturação de requisitos em tesouros para sua documentação, bem como para recuperação e reuso.

Em relação aos aspectos da notação semântica de requisitos, Machado *et al.* (2011) também utilizam um método baseado em *templates* para dar apoio ao gerenciamento de mudanças nos requisitos durante a evolução do projeto. Neste trabalho, também é desenvolvida uma arquitetura cujo foco é o repositório semântico e o módulo de anotação semântica (mapeamento). Os principais benefícios são o controle da rastreabilidade entre os requisitos e o histórico de mudanças. Uma das limitações é a necessidade de uma interface amigável para o usuário.

Kaiya e Saeki (2005) propõem um método para análise de requisitos de *software* baseado em uma ontologia de domínio. O diferencial é que esse sistema ontológico consiste em um tesouro e em regras de inferência. Uma parte do tesouro é responsável por conter os conceitos do domínio e relacionamentos específicos para o processamento adequado da ontologia. Por meio de um estudo de caso, os autores puderam comprovar que o tesouro contribuiu positivamente para a construção da ontologia de domínio, tornando mais precisos os processamentos semânticos.

Motta *et al.* (2017) sugerem um modelo para redução do impacto de problemas existentes na comunicação entre *stakeholders* durante a elicitação de requisitos, gerando um impacto nos atributos de qualidade relacionados. Esse modelo cria uma hierarquia de conceitos que auxiliam na análise da consistência e da completude da informação da arquitetura de informação dos requisitos, permitindo rastreabilidade. A mesma estrutura pode ser aplicada para requisitos funcionais ou não funcionais. A proposta de compartilhamento de informações estruturadas em triplas ontológicas se mostrou viável e útil.

Blumauer (2014) apresenta os benefícios em utilizar SKOS para organização e para integração de dados corporativos por meio de grafos de conhecimento. As principais vantagens estão relacionadas à simplicidade do vocabulário e à fácil compreensão por parte dos *stakeholders*, resultando em menor resistência em sua utilização.

Em uma pesquisa recente, Fagundes *et al.* (2020) realizam uma análise sobre as possibilidades de utilização dos SOC: taxonomias, tesouros e ontologias durante as atividades da ER. Em relação aos tesouros, foi constatada que sua utilização pode trazer benefícios para as atividades de concepção, de levantamento e de especificação de requisitos. Eles podem auxiliar na compreensão de termos específicos de um domínio e na documentação, permitindo que cada equipe possa construir seu próprio tesouro para utilização em projetos futuros. É esperado que a pesquisa inspire outras que realizem a interseção entre Ciência da Informação e ER.

Em comparação com esses trabalhos, o método proposto nesta pesquisa corrobora a proposta de Kaiya e Saeki (2005). O mapeamento das informações dos requisitos com o auxílio de ontologias para estruturá-los em tesouros trouxe benefícios ao procedimento de consulta e de recuperação dos requisitos. A partir da inferência realizada sobre tesouros semânticos, o protótipo conseguiu sugerir novos recursos como respostas para a consulta do usuário, tornando a busca por requisitos mais inteligente e ágil.

A criação dos grafos dos tesouros semânticos teve como um dos embasamentos a estrutura taxonômica de Motta *et al.* (2017), o que facilitou bastante a implementação com a biblioteca Jena. Também auxiliou no mapeamento das informações de RF e RNF para triplas ontológicas, permitindo a aplicação dos conceitos de termos genérico-específicos na construção de tesouros.

Como o protótipo desenvolvido para avaliação desta proposta é um *Web service*, o desenvolvimento de uma interface amigável é facilitado, considerando-se o ponto levantado por Machado *et al.* (2011), em relação ao uso *templates* em sua arquitetura de anotação semântica. Nesta pesquisa, os desenvolvedores podem criar seus próprios formulários e os modos de exibição em sua aplicação cliente.

A utilização de SKOS contribui para a difusão do conhecimento relacionado a requisitos de forma organizada e sistemática. Isso também ajuda a precisão e a automatização de consultas e a recuperação da informação na *Web*, rastreadas por meio do URI único de cada conceito, devido ao refinamento agregado às relações semânticas entre os termos dos tesouros. Assim como mencionado por Blumauer (2014), é possível a reutilização dessas informações em diferentes projetos, por meio de uma descrição simples dessas relações, facilmente entendida pelas partes envolvidas.

Por fim, o método proposto pode ser considerado uma confirmação dos resultados da pesquisa Fagundes *et al.* (2020). Os tesouros são uma forma eficaz para documentar o conhecimento relativo a requisitos, promovendo a recuperação para reúso, tendo como base os resultados iniciais obtidos no estudo de viabilidade. Como o trabalho de Fagundes *et al.* (2020) também é recente, isso pode indicar que conexões ou aplicações de SOC na ER são um tema emergente para pesquisa.

Conclusão

Neste trabalho, foi apresentado um método para a criação sistemática de tesouros de requisitos, usando as tecnologias e os padrões da *Web Semântica*, a fim de facilitar o acesso e o reúso desses requisitos. Entre os benefícios trazidos pelo método, é possível destacar a organização do conhecimento relativo aos requisitos em formato de tesouros semânticos, resultando em um padrão estruturado pré-definido de documentação compreensível tanto para humanos quanto para máquinas, que, por sua vez, podem realizar inferências de conhecimento, a partir dos relacionamentos estabelecidos, auxiliando na busca por requisitos e na tomada de decisão. Além disso, a estrutura facilita a naveabilidade entre os requisitos, aprimorando, consequentemente, a descoberta e a precisa recuperação de informação.

Em relação às fragilidades e às limitações, o estudo de viabilidade ressaltou dois pontos: a qualidade do requisito disponibilizado para reúso e uma possível resistência por parte da comunidade para desenvolver no ambiente da *Web Semântica*. Esses aspectos trazem uma reflexão sobre a proposta e auxiliam na elaboração de metas futuras e de próximos passos para o trabalho.

Como trabalhos futuros, será dada continuidade à pesquisa realizada sobre mecanismos de aferição da qualidade do requisito para que sejam implementados o mais breve possível. Nesse processo, serão consideradas as sugestões dos profissionais participantes do estudo de viabilidade e será consultado como essa ação é realizada em outros repositórios de requisitos. Também serão estudados procedimentos de gerenciamento das sugestões da comunidade desenvolvedora para melhorias no *Web service*.

Outro trabalho futuro mais ambicioso é estender esta noção de descrição semântica de requisitos para outros tipos de artefatos produzidos durante o desenvolvimento de um sistema. Em outras palavras, criar/reusar ontologias para cada tipo de artefato, de modo a viabilizar a criação de um extenso grafo de conhecimento RDF, contemplando todo o ciclo de vida de um projeto, desde os requisitos até a implementação.

Colaboradores

R. O. ZACARIAS contribuiu na concepção, desenho e edição de imagens, análise e interpretação dos dados, escrita e formatação do artigo. M. D. A. JACYNTHO e A. P. V. VASCONCELOS contribuíram na concepção, análise e interpretação dos dados, escrita, revisão e aprovação da versão final do artigo.

Referências

- Altounian, M. M. A.; Gomes, B. P. M. A recuperação semântica da informação no contexto do controle externo. *Revista do Tribunal de Contas da União*, n. 137, p. 31-41, 2016. Disponível em: <https://revista.tcu.gov.br/ojs/index.php/RTCU/article/view/1376>. Acesso em: 7 mar. 2019.
- Apache Software Foundation. *Apache Jena*. [S.l.:s.n.], 2011. Available from: <https://jena.apache.org/index.html>. Access on: May 3 2020.
- Barcelos, L. V. *Especificação de requisitos no domínio de sistemas de informação como uso de padrões*. 2016. 108p. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de São Carlos, São Carlos, 2016. Disponível em: <https://repositorio.ufscar.br/handle/ufscar/7116>. Acesso em: 10 jul. 2019.
- Beckett, D. et al. *RDF 1.1 Turtle*. Cambridge: W3C, 2014. Available from: <https://www.w3.org/TR/turtle/>. Access on: May 9 2020.
- Berners-Lee, T. *Linked Data*. Cambridge: W3C, 2009. Available from: <https://www.w3.org/DesignIssues/LinkedData.html>. Access on: May 3 2020.
- Berners-Lee, T. et al. Tabulator: exploring and analyzing linked data on the semantic web. In: International Semantic Web User Interaction Workshop, 3., 2006, Athens. *Proceedings* [...]. [Athens: s.n.], 2006. Available from: https://www.researchgate.net/publication/247696282_Tabulator_Exploring_and_Analyzing_linked_data_on_the_Semantic_Web. Access on: Jan 9 2019.
- Berners-Lee, T.; Hendler, J.; Lassila, O. The semantic web. *Scientific American*, n. 1, p. 34-43, 2001. Available from: <http://www.scientificamerican.com/article.cfm?id=the-semantic-web>. Access on: Jan 17 2019.
- Blumauer, A. SKOS as a key element in enterprise linked data strategies. In: International Semantic Web Conference, 13., 2014. Riva del Garda. *Proceedings* [...]. Aachen: CEUR Workshop Proceedings, 2014. Available from: <http://ceur-ws.org/Vol-1383/paper23.pdf>. Access on: Aug. 28 2019.
- Brickley, D. et al. *RDF Schema 1.1: W3C recommendation*. Cambridge: W3C, 2014. Available from: <https://www.w3.org/TR/rdf-schema/>. Access on: May 3 2020.
- Brickley, D.; Miller, L. *FOAF vocabulary specification 0.99*. [S.l.: s.n.], 2014. Available from: <http://xmlns.com/foaf/spec/>. Access on: May 3 2020.
- Dermeval, D. et al. Applications of ontologies in requirements engineering: a systematic review of the literature. *Requirements Engineering*, v. 21, p. 405-437, 2016.
- Diamantopoulos, T.; Symeonidis, A. Enhancing requirements reusability through semantic modeling and data mining techniques. *Enterprise Information Systems*, v. 12, p. 960-981, 2018.
- Dublin Core Metadata Initiative. *DCMI Usage Board*. Ohio: DCMI, 2012. Available from: <http://dublincore.org/documents/dcmi-terms/>. Access on: May 3 2020.
- Fagundes, P. B. et al. Taxonomias, ontologias e tesouros: possibilidades de contribuição para o processo de Engenharia de Requisitos. *Em Questão*, v. 26, p. 237–254, 2020.
- Fielding, R. T. et al. *Hypertext transfer protocol: HTTP/1.1*. Cambridge: W3C, 1999. Available from: <https://www.w3.org/Protocols/rfc2616/rfc2616.html>. Access on: May 3 2020.
- Gabriel Junior, R. F.; Laipelt, R. D. C. F. THESA: ferramenta para construção de tesouro semântico aplicado interoperável. *P2P & Inovação*, v. 3, p. 124-145, 2017.
- Gandon, F.; Schreiber, G. *RDF 1.1 XML Syntax*. Cambridge: W3C, Available from: <https://www.w3.org/TR/rdf-syntax-grammar/>. Access on: May 9 2020.
- Harris, S.; Seaborne, A. *SPARQL 1.1 Query Language*. Cambridge: W3C, 2013. Available from: <https://www.w3.org/TR/sparql11-query/>. Access on: May 12 2020.

- Heath, T.; Bizer, C. *Linked Data: evolving the Web into a global data space*. San Raphael: Morgan & Claypool, 2011. v. 1.
- Horrocks, I. et al. *Web ontology language*. Cambridge: W3C, 2012. Available from: <https://www.w3.org/OWL/>. Access on: May 3 2020.
- Isaac, A.; Summers, E. *SKOS Simple Knowledge Organization System Primer*. Cambridge: W3C, 2009. Available from: <https://www.w3.org/TR/2009/NOTE-skos-primer-20090818/>. Access on: Aug 24 2019.
- Isotani, S. et al. Ontology driven software engineering: a review of challenges and opportunities. *IEEE Latin America Transactions*, v. 13, p. 863-869, 2015.
- Jacyntho, M. D.; Schwabe, D. A multigranularity locking model for RDF. *Journal of Web Semantics*, v. 39, p. 25-46, 2016.
- Juristo, N. et al. Guidelines for eliciting usability functionalities. *IEEE Transactions on Software Engineering*, v. 33, p. 744-758, 2007.
- Kaiya, H.; Saeki, M. Ontology based requirements analysis: lightweight semantic processing approach. In: International Conference on Quality Software, 5, 2005, Melbourne. *Proceedings* [...]. Melbourne: IEEE Xplore, 2005. Available from: <http://ieeexplore.ieee.org/document/1579139/>. Access on: Jul. 21 2019.
- Kaiya, H. et al. Enhancing domain knowledge for requirements elicitation with web mining. In: Asia Pacific Software Engineering Conference, 2010, Sydney, Australia. *Proceedings* [...]. Sydney: IEEE Xplore, 2010.
- Machado, B. et al. Using semantic annotations for supporting requirements evolution. In: International Conference on Software Engineering & Knowledge Engineering, 23, 2011, Miami. *Proceedings* [...]. Miami: Knowledge Systems Institute Graduate School, 2011.
- Maculan, B. C. M. S. *Estudo e aplicação de metodologia para reengenharia de tesouro: remodelagem do THESAGRO*. 2015. 339f. Tese (Doutorado em Ciência da Informação) – Universidade Federal de Minas Gerais, Belo Horizonte, 2015. Disponível em: <https://repositorio.ufmg.br/handle/1843/BUBD-9ZKMUV>. Acesso em: 5 maio 2020.
- Manola, F.; Miller, E. *RDF Primer*. Cambridge: W3C, 2004. Available from: <https://www.w3.org/TR/rdf-primer/>. Access on: May 8 2019.
- Marcondes, C.H. "Linked data": dados interligados e interoperabilidade entre arquivos, bibliotecas e museus na web. *Encontros Bibli: Revista Eletrônica de Biblioteconomia e Ciência da Informação*, v. 17, p. 22, 2012. Disponível em: <https://periodicos.ufsc.br/index.php/eb/article/view/1518-2924.2012v17n34p171>. Acesso em: 3 maio 2020.
- Mendes, P. R. et al. Tesauros no acesso à informação: uma retrospecção. *Revista ACB: Biblioteconomia em Santa Catarina*, v. 20, p. 49-66, 2015.
- Montequin, V. R. et al. Analysis of the success factors and failure causes in Information and Communication Technology (ICT) projects in Spain. *Procedia Technology*, v. 16, p. 992-999, 2014.
- Motta, R. W. et al. Elicitação de requisitos orientada por ontologias de domínio. *Revista Gestão do Conhecimento e Tecnologia da Informação*, v. 1, n. 2, p. 20, 2017.
- Palomares, C. et al. Requirements reuse and requirement patterns: a state of the practice survey. *Empirical Software Engineering*, v. 22, p. 2719-2762, 2017.
- Pressman, R. S. *Engenharia de software: uma abordagem profissional*. 7th. ed. Porto Alegre: AMGH Editora, 2011.
- Ramalho, R. A. S. Análise do modelo de dados SKOS: Sistema de Organização do Conhecimento Simples para a Web. *Informação & Tecnologia*, v. 2, p. 66-79, 2015.
- Schema.org Community Group. *Schema.org*. [S.I.]: Schema.org, 2011. Available from: <http://schema.org/>. Access on: May 3 2020.
- Silva, A. R. *Uma Abordagem para Definição de Guias de Elicitação de Requisitos Não-Funcionais*. 2016. 241f. Tese (Doutorado em Informática Aplicada) – Universidade de Fortaleza, Fortaleza, 2016. Disponível em: http://bdtd.ibict.br/vufind/Record/UFOR_8074ec2b34096207e1ac59bba7436379. Acesso em: 5 maio 2020.
- Sommerville, I. *Software engineering*. 9th. ed. Boston: Pearson, 2011.
- Sporny, M. et al. *JSON-LD 1.0: A JSON-based serialization for linked data*. Cambridge: W3C, 2014. Available from: <https://www.w3.org/TR/json-ld/>. Access on: May 9 2020.
- Xavier, J. R. *Criação e instanciação de arquiteturas de software específicas de domínio no contexto de uma infra-estrutura de reutilização*. 2001. 123f. Dissertação (Engenharia de Sistemas e Computação) – Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2001. Disponível em: <https://www.cos.ufrj.br/index.php/pt-BR/publicacoes-pesquisa/details/15/85>. Acesso em: 5 maio 2020.
- Zacarias, R. O. *Uma proposta de documentação e reúso de requisitos baseada em tesauros semânticos*. 2020. 137f. Dissertação (Mestrado Profissional em Sistemas Aplicados à Engenharia e Gestão) – Instituto Federal Fluminense, Campos dos Goytacazes, 2020. Disponível em: <https://drive.google.com/open?id=143E5suirXSyk31K48xQVWAA40f6gNVDY>. Acesso em: 5 maio 2020.