

COMPARANDO AS LINGUAGENS PARA REPRESENTAÇÃO DE ONTOLOGIAS¹

COMPARING LANGUAGES FOR THE REPRESENTATION OF ONTOLOGIES

Mirkos O. Martins² e Giovanni Rubert Librelotto³

RESUMO

O objetivo, neste artigo, é contextualizar um *Ontology Builder*, um extrator de ontologias, a partir de uma família de documentos XML (*eXtensible Markup Language*), uma linguagem de marcação que armazena tanto os dados quanto seus identificadores, obtido de uma XSO (*XML Specification for Ontologies*), especificação de ontologias em XML que, em sua execução, gera outro XSO, o qual é a representação de ontologias em RDF (*Resource Description Framework*), OWL (*Web Ontology Language*) ou XTM (*XML Topic Maps*). A apresentação dessas linguagens de representação e sua consequente comparação fornecerão um indicador que será melhor entre elas para a adoção ao *Ontology Builder*, possibilitando trabalhos futuros que sejam uma métrica já testada para construir suas próprias ontologias.

Palavras-chave: *Ontology Builder*, documentos XML, *Resource Description Framework*, *Web Ontology Language*.

ABSTRACT

The goal of this article is to put in context an Ontology Builder based on a family of XML documents (eXTENSIBLE Markup Language), a marking language that stores data along with their identifiers, obtained from an XSO (XML Specification for Ontologies), and in its running it generates another XSO than the ontology specification in RDF (Resource Description Framework), OWL (Web Ontology Language) or XTM (XML Topic Maps). The presentation of these representation languages

¹ Trabalho de Iniciação Científica - PROBIC.

² Mestrando em Nanociências - UNIFRA.

³ Orientador - UNIFRA.

and consequently comparison will allow an indicator of which is the best among them for adoption to the Ontology Builder, allowing future works to follow a metric already tested to construct its own ontologies.

Keywords: *Ontology Builder, XML documents, Resource Description Framework, Web Ontology Language.*

INTRODUÇÃO

Devido ao gigantismo de informações presentes na *World Wide Web* (doravante designada abreviadamente por *web*) e à velocidade com que essa informação é modificada, a tarefa de efetuar buscas torna-se complexa. No intuito de minimizar o problema da complexidade de procuras na *web*, surgiu uma nova área de pesquisa: a *web* semântica.

Em muitos domínios da *web*, a relevância da informação procurada deve ser preferencialmente estimada, num nível de conhecimento profundo do domínio de problema em questão. Nesse contexto, algumas propostas envolvem a criação de metadados que seguem modelos de ontologias. Uma ontologia (GRUBER, 1993) é um conjunto de termos hierarquicamente estruturado para a descrição de um domínio, o qual pode ser utilizado como um esqueleto fundamental para uma base de conhecimento.

As ontologias colaboram para se obter uma *web* em que os recursos disponíveis são acessíveis não somente por seres humanos, mas também por processos automatizados. Essa automação provoca a elevação do *status* da *web* de *machine-readable* (lida mecanicamente) para algo que é chamado de *machine-understandable* (entendida mecanicamente). Isso reflete a visão de Berners Lee sobre a *web* semântica (BERNERS-LEE et al., 2001).

A fim de poder extrair a ontologia a partir de documentos XML (*eXtensible Markup Language*), decidiu-se criar um extrator que permite, a partir de uma especificação, abstrair dos documentos XML uma ontologia. Esse extrator será aumentado de modo a retirar partes de ontologia de outras fontes, como páginas (documentos HTML) na *web*, base de dados, etc. e será chamado de *ontology builder* (*OntBuild*).

Um *OntBuild* é fortemente dependente da estrutura dos recursos de informação. Trabalhando com vários tipos de documentos XML, é necessário implementar diversos *ontbuilds* (um para cada tipo). Para minimizar o esfor-

ço em estudo neste problema, foi criado um modelo XML para descrever a extração do conhecimento de documentos XML a fim de produzir uma ontologia. É, assim, apresentada uma proposta de linguagem para especificação de ontologias a partir de um tipo de documentos XML, chamada de XSO (*XML Specification for Ontologies*).

Ao processar uma família de documentos no *ontbuild*, obtém-se uma ontologia que deve ser representada em uma linguagem apropriada: RDF (*Resource Description Framework*), OWL (*Web Ontology Language* ou XTM (*XML Topic Maps*)). A decisão sobre qual dessas linguagens será adotada no *ontbuild* será vista na seção 5, após uma comparação entre elas.

Como forma de introduzir o tema do artigo, a seção 2 é dedicada aos conceitos básicos: *web* semântica e ontologias. Na seção 3, será apresentado o extrator de ontologias, o *OntBuild*. As linguagens para representação de ontologias, assim como o estudo comparativo entre elas serão descritos na seção 4. Por fim, na conclusão, serão mostradas as considerações que levaram a decidir pela adoção de XTM e, será, então, apresentada uma visão concreta e completa do gerador e extrator *Ontbuild*.

WEB SEMÂNTICA E ONTOLOGIA

A *Semantic Web* (*Web* semântica) expressa o conhecimento do significado de recursos da *web* e é armazenada por meio da utilização de (meta) dados processáveis por máquinas. Acredita-se que a *web* semântica não seja separada da *web*, mas uma extensão da tecnologia corrente. Basicamente, os mecanismos a serem desenvolvidos para o estabelecimento da *web* semântica compreendem duas vertentes: a disponibilização de um conjunto de coleções estruturadas de informações e regras de inferência associadas a esses conjuntos; e a criação de agentes de *software* capazes de percorrer a *Web* realizando tarefas complexas com base nessas estruturas de conhecimento.

Na verdade, o componente responsável por representar e manter tais coleções estruturadas são as ontologias. Uma ontologia é uma teoria lógica para relacionar o significado pretendido de um vocabulário formal, isto é, seu comprometimento com uma conceituação particular do mundo.

O desenvolvimento de ontologias irá promover o mecanismo de construção da parte semântica da *web*. O modelo em camadas, proposto por Berners-Lee et al. (2001), tem sido aceito, principalmente, como representação para a arquitetura da

web semântica. O desenvolvimento de tais mecanismos depende obrigatoriamente de linguagens que expressem a informação de maneira a ser entendida por máquinas. O desafio é proporcionar uma linguagem que manipule igualmente, de maneira eficiente, dados e regras para deduções sobre esses dados e que permita que as regras existentes em qualquer sistema de representação de conhecimento possam ser exportadas para *web*.

EXTRATOR DE ONTOLOGIAS

A partir da criação do manual de algumas ontologias (descritas em XML), verificou-se que a tarefa de criação, além de consumir tempo, é bastante repetitiva. Nesse contexto, decidiu-se desenvolver um extrator de ontologias a partir de um conjunto de documentos XML, isto é, um *ontology builder* (*OntBuild*).

O *ontbuild* é um conversor de uma linguagem XML em outra linguagem XML, uma *stylesheet* XSL que recebe um documento XML como entrada e gera outro documento XML que contém a especificação de uma ontologia.

Mesmo recorrendo aos serviços de um extrator como o *ontbuild*, a tarefa de criar ontologias é árdua, pois o extrator depende do tipo de documento XML a processar. Isso significa a recodificação do *ontbuild* cada vez que a família de documentos a manipular obedeça a um esquema diferente. Essa constatação levou a pensar em unir ao *ontbuild* um gerador automático de *ontbuilders*, formando, assim, um sistema completo cuja arquitetura genérica é ilustrada na figura 1.

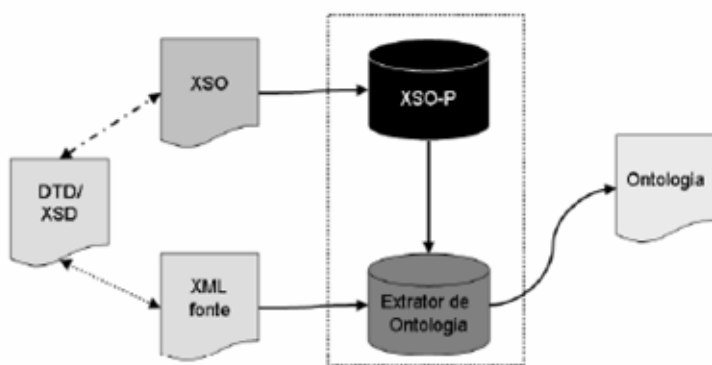


Figura 1 - A arquitetura do sistema *OntoBuild*.

Para poder concretizar essa ideia de gerar o *ontbuild*, é necessário especificar formalmente o processo de extração concreto para cada tipo de

documentos a processar. Isso levou a definição de uma nova linguagem XML: a XSO (*XML Specification for Ontologies*).

A formalização da linguagem XSO foi publicada em outro artigo (LIBRELOTTO et al., 2003). A XSO, porém, não é suficiente para montar o sistema da figura 1; falta a definição da linguagem de representação de ontologias que o *ontbuild* irá utilizar. Para essa decisão, serão analisadas as características de cada uma das três alternativas mais importantes para descrição de ontologias (RDF(S), OWL ou XTM) na seção seguinte.

Nessas circunstâncias, entende-se que é possível gerar automaticamente o extrator de ontologias, desenvolvendo outro processador para transformar as especificações em XSO para o código da linguagem de representação a ser escolhida nas seções posteriores.

O processador XSO é uma das principais peças nessa arquitetura, como pode ser visto na figura 1. A partir de uma especificação em XSO (uma instância XML), o processador gera uma *stylesheet* XSL, a qual processará o conjunto de documentos de entrada XML para extrair a ontologia desejada.

LINGUAGENS DE REPRESENTAÇÃO DE ONTOLOGIAS

Nesta seção, são apresentadas e avaliadas três linguagens XML, usualmente utilizadas para a construção de ontologias: RDF, RDF(S), OWL e *Topic Maps*.

RDF E RDF SCHEMA (RDF(S))

A linguagem RDF (LASSILA; SWICK, 1999) é a base para o processamento de metadados na *web* semântica, fornecendo um simples modelo de dados e uma sintaxe padronizada para metadados. Basicamente, ele comprova a linguagem para a escrita das declarações. A próxima é a camada de esquema, fornecida pela especificação RDF(S). Uma linguagem de representação de conhecimento formal pode ser usada como a terceira camada, a camada lógica.

A modelagem das primitivas oferecidas por RDF são muito básicas e o modelo básico de dados consiste em três tipos de objetos. Essas três partes de uma declaração são chamadas, respectivamente, de sujeito, predicado e objeto. Entretanto, RDF(S) (BRICKLEY; GUHA, 2000) define além das primitivas de modelagem de RDF. Exemplos são classes, subclasses, restrições de domínio (*domain*) e alcance (*range*) para propriedades e subpropriedades.

OWL

A OWL foi baseada em OIL e DAML + OIL (linguagem fundada em conjunto pelo DARPA e o instituto europeu IST) e atualmente é uma recomendação da W3C para linguagem, a fim de definir e instanciar ontologias na *web* (SMITH et al., 2004). A OWL é dividida em três sublinguagens: OWL *Lite*, OWL DL e OWL *Full*. A OWL *Lite* suporta classificação hierárquica e restrições simples. A OWL DL tem como características a computabilidade (todas as conclusões podem ser computáveis) e decidibilidade (todas as computações terminarão num tempo finito). Por fim, a OWL *Full* possui uma liberdade sintática e não uma garantia computacional (como a OWL DL por exemplo).

XML *TOPIC MAPS* (XTM)

Topic maps (PARK; HUNTING, 2003) é um formalismo para representar conhecimento acerca da estrutura de um recurso de informação e para organizar isso em tópicos. Esses tópicos têm ocorrências e associações que representam e definem relacionamentos entre os tópicos. A informação sobre os tópicos pode ser inferida ao se examinarem as associações e ocorrências ligadas ao tópico. Uma coleção desses tópicos e associações é chamada de *topic maps*. Também pode ser visto como um paradigma que permite organizar, manter e navegar pela informação, para transformá-la em conhecimento.

Topic maps pode ser visto como uma descrição de um ponto de vista, por tópicos formalmente declarados, e pela ligação de partes relevantes do conjunto de informação para os tópicos apropriados.

COMPARAÇÃO ENTRE AS LINGUAGENS

Nesta seção, realiza-se uma comparação entre as três linguagens apresentadas na seção 4, com base nos trabalhos de Ratnakar e Gil (2000) e Gómez-Pérez e Corcho (2000). A definição dos critérios de comparação foge ao escopo deste trabalho, porém acredita-se que a síntese a seguir é suficiente para proporcionar o entendimento necessário sobre cada uma das dimensões consideradas no estudo que se sintetiza na tabela 1 (evita-se explicar os parâmetros cuja designação os torna óbvios).

- Contexto: modularidade é uma importante consideração, especialmente em ambientes distribuídos. O mesmo termo pode ser interpretado de

acordo com o contexto no qual ele é definido. É importante que a linguagem expresse os diferentes contextos alternativos.

- Classes e propriedades: relações entre classes e objetos. Propriedades relatam diferentes classes de objetos. Classes e propriedades, por vezes, são chamadas de conceitos e funções, de objetos e atributos ou, de coleções e predicados.
- Herança: indica que as restrições e os valores das propriedades das classes-pai são os mesmos das subclasses.
- Tipos de dados primitivos: um conjunto comum de tipos de dados primitivos, como *strings* e *números*, pode ser diretamente usado para compor novos tipos de dados complexos.
- Instâncias: esses objetos denotam indivíduos, os quais podem ser descritos em termos de suas propriedades ou especificados para serem membros de uma classe.
- Valores: além de escopo e cardinalidade, os valores podem ser associados a uma propriedade ainda mais restrita, como, por exemplo, um valor por omissão pode ser provido. Uma enumeração de possíveis valores pode ser fornecida como um conjunto de escolhas. Conjuntos ordenados especificam a ordem dos elementos.
- Negação, conjunção e disjunção: são declarações de negação de qualquer descrição permitidas em uma linguagem. São, frequentemente, úteis, mas o custo computacional é íngreme e com resultado somente em formas limitadas de negação, são tipicamente suportadas em uma dada linguagem. Expressões de disjunção são, geralmente, usadas para descrever relações entre subclasses.
- Definições: se as condições necessárias e suficientes para classes membros podem ser especificadas. O sistema pode usar essas definições para ponderar relações de classes, subclasses e para determinar se instâncias são membros de classes, como na descrição de sistemas lógicos.

Há outras dimensões em relação à expressividade que poderiam ser consideradas, mas as principais são as que se priorizaram neste trabalho.

Tabela 1 – comparação entre RDF(S), OWL e XTM.

Dimensões	Detalhes	RDF Schema	OWL	XTM
Contextos	Contextos	Sim	Sim	Sim <scope>
Classes	Classes de Objetos e Propriedades	Sim (rdfs:Class e rdfs:Property)	Sim (daml:Class, ObjectProperty, daml:DatatypeProperty)	Sim, <topicMap>; <topic>; <instanceOf>
	Herança	Sim, (Propriedades e Classes) (rdfs:subClassOf, rdfs:subPropertyOf)	Sim (Propriedades e Classes) Usa sintaxe RDF	Sim <instanceOf>; <topic>; <topicMap>
Propriedades de Restrições de Elementos	Propriedade / Faixa de Elemento	Sim (Somente local – rdfs:range)	Sim (Global – rdfs:range Local – daml:Restriction_onProperty, toClass)	Sim <topicRef> Permite apontar para Faixas definidas em um URI
	Propriedade/ Domínio de Elemento	Sim (Somente global – rdfs:domain)	Sim (Global) – rdfs:domain	Sim <topicRef> Permite apontar para domínios definidos em um URI
	Propriedade / Cardinalidade de Elemento	Não	Sim (Local – minCardinality, xCardinality, Cardinalidade Global – UniqueProperty, ou Restriction subClasse de “#Resource”)	Sim <topicRef>; <association> Permite apontar cardinalidades definidas em um URI
Tipos de Dados e Instâncias	Tipos Básicos	Não (Somente literais, nesta versão)	Sim, permite o uso de tipos de dados de XMLSchema	Sim <topic>; <association>; <roleEspec>; <subjectIndicatorRef> Permite apontar para Tipos de dados definidos em um URI
	Enumerados	Não	Sim <daml:oneOf> Pode apontar para tipos enumerados de XMLSchema	Sim <topicRef> Permite apontar para Tipos de dados definidos em um URI
	Instâncias	Sim <rdf:ID>	Usa sintaxe RDF	Sim <topic>; <baseName>; <baseNameString>; <occurrence>
Conjuntos de Dados	Listas	Não	Sim <daml:collection>	Sim <variant> Permite apontar para dados definidos em um URI
	Listas ordenadas	Sim <rdf:Seq>	Sim <daml:list>	Sim <variantName> permite apontar para dados definidos em um URI

Negação / Disjunção / Conjunção	Negação	Não	Sim <daml:ComplementOf>	Não diretamente
	Classes Dis- juntivas	Não	Sim <daml:disjointUnionOf> <daml:unionOf>	Não diretamente
	Classes Con- juntivas	Sim, Múltiplos <rdf:subClassOf>	Sim <daml:intersectionOf>	Sim <mergeMap>; <as- sociation>; <member>
Definições	Condições Necessárias e Suficientes para afiliação	Não	Sim, <daml:sameClassAs> <daml:UnambiguousProperty>	Sim <baseName>; <baseNameString>
Tipos de Propriedades	Inversa	Não	Sim <daml:inverseOf>	Sim <roleSpec>
	Transitiva	Não	Sim <daml:TransitiveProperty>	Sim <roleSpec>
Reificação	Reificação	Não	Não diretamente	Sim <subjectIdentity>
Contexto	Contexto	Não	Sim diretamente	Sim (SCOPES) <subjec- tIdentity>

Antes de concluir qual linguagem de representação de ontologias será escolhida, mais relações serão realizadas entre os diferentes modelos. Para isso, a comparação deter-se-á entre RDF e XTM, pois a OWL utiliza o mesmo modelo de dados de RDF.

RDF e XTM têm o mesmo conceito central, o qual pode ser chamado *entidade*. Eles têm noções completamente diferentes de como as características são associadas a essas *entidades*; suas ideias de como estabelecer as identidades são diferentes também. Em declarações RDF, a forma de triplas é a única maneira de associar características, enquanto em XTM, os tópicos podem ter nomes, ocorrências e participarem (desempenhar papéis) em associações.

Em RDF, a URI de um recurso identifica-o. Em XTM, os tópicos podem ter uma URI que aponta para o recurso, que é o tópico, ou qualquer número de URIs que apontam para os recursos que demonstram o que o tópico é.

Na tabela 2, cada coluna mostra uma possível correlação de termos entre a XTM e RDF. Ela provém de um sumário das comparações dos modelos.

Tabela 2 – correlação de termos entre XTM e RDF.

Termo XTM	Relacionamento	Termo RDS
Topic map	comparável à	RDF graph
Topic	comparável à	Resource
Subject	comparável à	Resource
Resource	comparável à	Network-tetrieveable resource
Non-addressable subject	comparável à	Non-network-retrievable resource
Association	é visto como	Statement
Occurence	é visto como	Statement
Name assignment	é visto como	Statement
Class of topics	comparável à	Class

RDF tem um nível inferior a XTM e isso significa que é impossível compreender modelos RDF sem o conhecimento de seu próprio esquema (RDF(S)). Entretanto, com XTM é possível. Baseando-se na comparação dos modelos, sabe-se que um mapeamento de esquemas de RDF para XTM é praticamente impossível, porque XTM tem um nível mais alto de abstração.

CONCLUSÃO

Neste artigo, compararam-se os modelos de dados e as características com linguagens de anotação mais coerentes para descrição de ontologias – RDF, OWL e XTM. Concluiu-se que RDF e OWL partilham o mesmo modelo de dados e são uma família de especificação simples, enquanto XTM é completamente diferente.

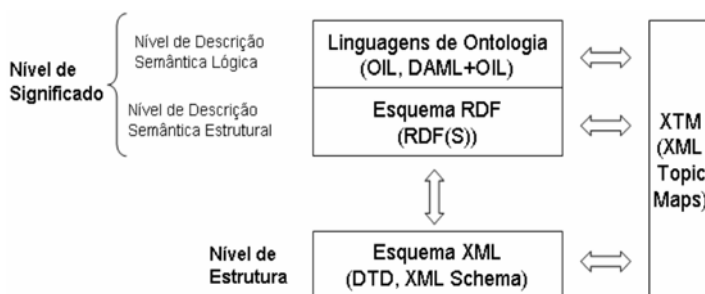


Figura 2 – Arquitetura para representação do conhecimento.

Uma diferença chave é que a XTM utiliza um conceito centrado em tópicos, enquanto a RDF vale-se de um conceito centrado em recursos. XTM inicia com tópicos e modela uma camada de rede semântica sobre os recursos de informação; eles têm, a capacidade de modelar conhecimentos sem qualquer referência para os recursos fundamentais. RDF inicia com recursos e os anota diretamente. Somente pelo entendimento do modelo, além de seu real objetivo, pode-se representar a camada abstrata.

Após a análise das linguagens estudadas neste trabalho, na figura 2, apresenta-se uma possível arquitetura, contendo as ligações entre essas linguagens de forma a estabelecer níveis de descrição para representação do conhecimento. Foram considerados dois grandes níveis: o de estrutura e o de significado. No nível sintético, existe o *XML Schema*, descrevendo a estrutura dos documentos XML. O nível de significado foi subdividido em dois outros que são: descrição semântica estrutural e descrição semântica lógica. No nível de descrição semântica estrutural existe o esquema RDF; e no nível de descrição semântica lógica estão as linguagens de ontologia, como, por exemplo, a OIL e OWL.

Com base nos argumentos apresentados, decidiu-se que a linguagem para a representação de ontologias a ser adotada pelo *ontbuild* será *XML topic maps* (XTM). Sendo assim, a linguagem XSO passa a ser denominada linguagem XSTM (*XML Specification for Topic Maps*), pois o documento final a ser gerado (contendo a ontologia extraída do conjunto de recursos XML) é um *topic map*. Assim, o nome genérico *ontology bilder* passa a ser *topic map builder*, ou, simplesmente, *TM-Builder*.

Ainda, este trabalho fornece subsídio na hora de escolher uma linguagem de representação para a construção de uma ontologia, poupando esforços do desenvolvedor para comparações (e todo seu tempo dedicado a estudar as três linguagens contempladas neste artigo), liberando-o para a construção da ontologia propriamente dita. Com a redução de tempo no desenvolvimento, a equipe poderá estudar com mais detalhamento a linguagem escolhida e, com isso, utilizá-la da forma mais precisa possível. A relevância deste trabalho reside no fato de que futuras construções de ontologias poderão ser otimizadas em relação a apenas uma linguagem de representação, aumentando a produtividade e a coesão de informações entre as diferentes pessoas envolvidas nos projetos.

REFERÊNCIAS

BERNERS-LEE, T.; HENDLER, J. , LASSILA, O. The Semantic Web. **Scientific American**, 2001.

BRICKLEY, D.; GUHA, R. V. **Resource Description Framework (RDF) Schema Specification 1.0**, 2000. World Wide Web Consortium.

GOMEZ-PREZ, A.; CORCHO, O. **Evaluating Knowledge Representation and Reasoning Capabilities of Ontology Specification Languages**, 2000. In: PROCEEDINGS OF THE ECAI WORKSHOP ON APPLICATIONS OF ONTOLOGIES AND PROBLEM-SOLVING METHODS.

GRUBER, T. R. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. In: GUARINO, N.; POLI, R., editors, **Formal Ontology in Conceptual Analysis and Knowledge Representation**, Deventer, The Netherlands. Kluwer Academic Publishers, 1993.

LASSILA, O.; SWICK, R. R. **Resource Description Framework (RDF) Model and Syntax Specification**. World Wide Web Consortium, 1999.

LIBRELOTTO, G. R.; RAMALHO, J. C.; Henriques, P. R. **TM-Builder: um Construtor de Ontologias baseado em Topic Maps**. In: XXIX CONFERENCIA LATINOAMERICANA DE INFORMATICA, La Paz, Bolívia, 2003.

PARK, J.; HUNTING, S. **XML Topic Maps: Creating and Using Topic Maps for the Web**, 2003, ISBN 0-201-74960-2. Addison Wesley.

RATNAKAR, V.; GIL, Y. **A Comparison of (Semantic) Markup Languages**. 2000. Disponível em: <http://trellis.semanticweb.org/expect/web/semanticweb/paper.pdf>.

SMITH, M. K.; WELTY, C.; MCGUINNESS. **Web Ontology Language OWL) – Guide Version 1.0**. Disponível em: <http://www.w3.org/TR/2004/REC-owl-guide-20040210/>. 2004.