

## RDF, RDF(S) e OWL: Uma Análise Comparativa Visando Atingir o Padrão Linked Data

A. N. S. Silva<sup>1</sup>, D. Y. S. Fernandes<sup>2</sup>, N. A. B. França<sup>3</sup>, R. D. Silva<sup>4</sup>, W. T. Sarinho<sup>5</sup>

<sup>1</sup>Instituto Federal da Paraíba - Campus João Pessoa

ayrtonnadal@gmail.com – damires@ifpb.edu.br – naftalifranca@hotmail.com – rafhael.ds@gmail.com –  
travassoswalter@gmail.com

### RESUMO

Todos os dias novas informações são adicionadas de forma aleatória e isoladas na rede mundial de computadores. No advento da Internet, não foi possível prever seu crescimento de forma a criar um padrão eficaz para realização de consultas e interoperabilidade das informações que eram disponibilizadas na rede. O HTML, linguagem de marcação mais difundida nesse meio, é capaz de criar apenas uma *tag* “meta” com uma definição simples de elementos de metadados que é insuficiente para inferir um significado mais concreto do que está sendo disponibilizado naquele momento. Isso obriga, quase sempre, a força humana a interpretar a informação que está sendo disposta. A proposta da web semântica é tornar os dados mais compreensíveis através da integração de linguagens, modelos e tecnologias, permitindo que agentes de software tenham consciência da informação que possuem. Essa nova forma de interpretação permitirá criar relações de seus dados com outras fontes semelhantes. Essa prática possibilita a interligação de informação, inferência de novos recursos e resgate de dados num contexto mais aproximado da necessidade real do usuário. Como base ao desenvolvimento da Web Semântica, surgiu o projeto *Linked Data*, que visa delimitar um conjunto de práticas para publicar e conectar conjuntos de dados estruturados na Web, com o intuito de criar um espaço global desses dados. Dessa maneira, RDF, RDF(S) e OWL podem ser vistos como três modelos capazes de serem utilizados conjuntamente para representação desses dados estruturados. Estes modelos possuem construtores com semântica distinta, mas, muitas vezes, complementar. Neste escopo, este trabalho apresenta um estudo de caso realizado com o objetivo de comparar semanticamente o conjunto de construtores existente em cada um desses três modelos. Para isto, define um exemplo de aplicação de dados, representando-a de acordo com cada modelo estudado. Como resultado, apresenta uma análise comparativa dos modelos e aponta caminhos para sua utilização de acordo com o padrão *Linked Data*.

**Palavras-chave:** RDF, RDF(S), OWL, *Linked Data*, Web Semântica

## 1. INTRODUÇÃO

A Internet contemporânea, nos moldes da *World Wide Web*, vive um constante processo de evolução e tem revolucionado a forma como criamos conteúdo e trocamos informações. A Web organizou as informações na Internet por meio de hipertexto e tornou a interação do usuário com a rede mundial mais amigável. Com isso, possibilitou um ambiente de compartilhamento de documentos oriundos das diversas áreas do conhecimento, publicados dinamicamente através de aplicações desenvolvidas para esse ambiente. Entretanto, esses conteúdos publicados normalmente seguem regras apenas sintáticas, com objetivos de apresentação, não permitindo que se consiga extrair semântica dos mesmos, nem ligá-los, sem que para isso seja feito um grande esforço de implementação. Considerando isso, a Web atual pode ser classificada como sintática e o processo de interpretação dos conteúdos disponibilizados fica geralmente a cargo dos usuários (COSTA e YAMATE, 2009).

Dessa forma, em sua maior parte, os dados na Web ainda são organizados para serem lidos ou compreendidos por humanos e não por agentes de software. Para que um agente de software possa entender e interpretar um dado, é necessário processar a semântica envolvida naquele dado, num determinado contexto. Neste escopo, semântica diz respeito à atribuição de significado a elementos, dados ou expressões que precisem ser interpretados numa dada situação (SOUZA *et al.*, 2011). No cenário da Web, isso representa atribuir significado aos dados interligando-os com outros conjuntos de dados ou outros domínios de conhecimento, conseguindo assim criar uma relação de significância entre os conteúdos publicados na Internet de modo que seja perceptível tanto pelo usuário quanto pelos agentes de software. Essa nova visão da Web vem sendo denominada de Web Semântica (*Semantic Web*) (BERNERS-LEE *et al.*, 2001).

A Web Semântica é considerada uma extensão da Web atual cujo objetivo principal é tornar os dados mais compreensíveis na rede através da integração de linguagens, modelos e tecnologias como o XML, RDF, RDF(S) e ontologias (LÓSCIO e WAGNER FILHO, 2009). A finalidade passa pelo desenvolvimento de um modelo que permita a partilha global de conhecimento assistido por máquinas (W3C, 2011), a partir de duas premissas básicas: (i) representar o conteúdo da Web para facilitar o seu processamento e (ii) categorizar as informações para facilitar o acesso. Como base ao desenvolvimento da Web Semântica, surgiu o projeto *Linked Data* que visa delimitar um conjunto de práticas para publicar e conectar conjuntos de dados estruturados na Web, com o intuito de criar uma “Web de Dados” (BIZER *et al.*, 2009). É baseado em tecnologias Web, como HTTP e URIs, com o objetivo de permitir a leitura dos dados conectados semanticamente, de forma automática, por agentes de software.

Para ligar dados distribuídos na Web, é necessário um mecanismo padrão que permita a especificação da existência e significado dos links entre objetos descritos nos diversos conjuntos de dados (HEATH e BIZER, 2011). Esse mecanismo é provido pelo padrão RDF (*Resource Description Framework*), um modelo para representar significado sobre recursos, que habilita agentes de software a explorar os dados de forma automática. Entretanto, usar somente RDF não é suficiente para definir a semântica de um domínio (LOSCIO e WAGNER FILHO, 2009). Para isso, pode-se usar RDF *Schema* ou RDF(S), um esquema que permite definir termos de um vocabulário e relacionamentos entre eles, ao tempo em que oferece um significado extra aos recursos e propriedades RDF (TAUBERER, 2008). Esta semântica extra especifica como um termo deve ser interpretado. Como meio de facilitar e melhorar a exploração desses dados, conhecimento semântico proveniente de vocabulários e ontologias também pode ser empregado (GRUBER, 1995). Ontologias são normalmente representadas através da linguagem OWL (*Web Ontology Language*), desenvolvida para aumentar a facilidade de expressar semântica disponível em RDF e RDF(S) e permitir inferência sobre os dados representados (TAUBERER, 2008).

Dessa maneira, RDF, RDF(S) e OWL podem ser vistos como três modelos capazes de serem utilizados conjuntamente para representação de dados estruturados na Web. Contudo, estes modelos possuem construtores com semântica distinta, mas, muitas vezes, complementar. Neste sentido, este trabalho apresenta um estudo de caso realizado com o objetivo de comparar semanticamente o conjunto de construtores existente em cada um desses três modelos. Para isto, define um exemplo de aplicação de dados e, a partir de sua modelagem, analisa como ela pode ser representada em cada modelo comparado. O trabalho contribui para a tarefa de identificação e compreensão das características de cada modelo (RDF, RDF(S) e OWL), de modo que uma pessoa que conheça um dos modelos possa entender os demais mais facilmente e, ao mesmo tempo, possa compreender a utilidade de usá-los em conjunto. Mais especificamente, o trabalho objetiva servir de base para o entendimento do padrão *Linked Data*. A ideia é que, a partir deste estudo, seja desenvolvida uma ferramenta que permita transformar dados publicados atualmente como HTML em RDF/RDF(S)/OWL, seguindo os princípios definidos no *Linked Data*.

Este trabalho está organizado da seguinte forma: a Seção 2 apresenta a aplicação de dados utilizada no estudo de caso; a Seção 3 descreve as características dos modelos RDF, RDF(S) e OWL, apresentando o modelo especificado da aplicação em cada um destes formatos; a Seção 4 analisa a comparação realizada. A Seção 5, finalmente, tece algumas considerações e indica trabalhos futuros.

## 2. ANÁLISE E INTERPRETAÇÃO DOS DADOS

### 2.1 Estudo de caso

O estudo de caso apresentado neste artigo enquadra-se no contexto de *projetos de pesquisa* e as principais entidades envolvidas no seu domínio. Neste escopo, existem as seguintes classes: projeto, relacionada com a classe nível e pesquisador, e esta (pesquisador) relacionada com publicação. A entidade pesquisador se especializa em professor ou aluno. O modelo de classes correspondente a esta descrição é apresentado na Figura 1. Este modelo será utilizado ao longo deste trabalho e será mapeado parcialmente para HTML e XML e, em seguida, para RDF, RDF(S) e OWL. O objetivo é mostrar a representação deste estudo de caso (suas entidades, relacionamentos e semânticas associadas) nos três modelos, fazendo um comparativo entre eles exaltando suas diferenças, limitações e qualidades específicas de seus respectivos construtores.

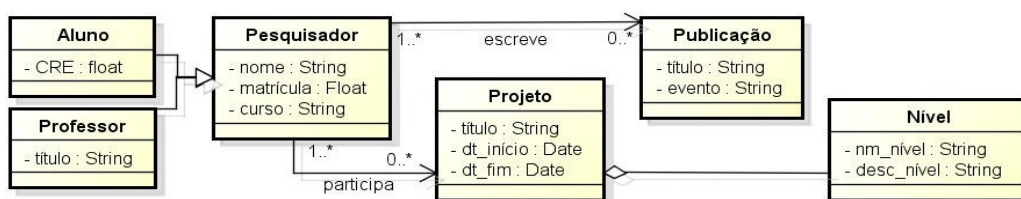


Figura 1 – Modelo de Classes para o Estudo de Caso “Projetos de Pesquisa”

### 2.2 Linguagens e Modelos para Web Semântica

Há um volume extenso de dados e documentos disponíveis na Web, mas sua utilização (consulta e interoperabilidade) tem sido dificultada em função do pouco uso de padrões. Isto acontece principalmente porque a linguagem usada para a exibição de dados (HTML) tem como objetivo a formatação dos dados, sem preocupação com definição de metadados (com exceção da tag “meta”, que possibilita a definição de elementos simples de metadados) (SANTOS, 2002). Assim, HTML - *Hypertext Markup Language*, é uma linguagem de marcação, que usa elementos chamados *tags*, que são interpretadas fornecendo informações para descrever uma página Web (W3C, 2011).

O *World Wide Web Consortium* (W3C) é um consórcio internacional responsável pela definição de padrões e diretrizes a serem usadas na Web desde 1994 (W3C, 2011). A partir das necessidades apontadas pela comunidade que utiliza a Web, o W3C vem criando padrões e aprimorando os já existentes. Uma dessas necessidades foi a quebra das limitações das *tags* predefinidas da linguagem HTML, o que propiciou a criação da linguagem XML, padronizada pelo W3C em 1998 (SANTOS, 2001). Com a linguagem XML, o requisito de exibição foi substituído pelo da descrição de conteúdo.

A XML - *Extensible Markup Language* é uma linguagem de marcação extensiva, que, assim como o HTML, utiliza-se de *tags* para representar dados, mas diferentemente dela (que possui *tags* definidas), no XML o desenvolvedor tem a liberdade de escolher o nome que lhe convier. XML foi projetada para permitir o armazenamento e compartilhamento de dados entre computadores, possibilitando a estruturação e o aninhamento desses dados em profundidades arbitrárias (W3C, 2011).

Considerando o escopo do estudo de caso introduzido na Seção 2, ilustramos a diferença entre HTML e XML, através da Tabela 1. Neste exemplo, é possível compreender a descrição de informações sobre um projeto de pesquisa, de nome “Transformando PHP em HTML”, em que a pesquisadora é “Maria” e o aluno é “Pedro”. Ou seja, informações sobre um projeto de pesquisa são codificadas em HTML com o propósito de apresentação, enquanto são representadas em XML, com o propósito de descrição. Podemos perceber que a informação é mais clara com o uso de XML.

**Tabela 1- HTML x XML**

HTML	XML						
<h2>Projetos de Pesquisa</h2> <table><tr><th>Pesquisador</th><th>Projeto</th><th>Aluno</th></tr><tr><td>Maria</td><td>Transformando PHP em HTML</td><td>Pedro</td></tr></table>	Pesquisador	Projeto	Aluno	Maria	Transformando PHP em HTML	Pedro	<pre>&lt;?xml version="1.0"?&gt; &lt;projetoPesquisa&gt;   &lt;projeto&gt;     &lt;pesquisador&gt;Maria&lt;/pesquisador&gt;     &lt;projeto&gt; Transformando PHP em HTML &lt;/projeto&gt;     &lt;aluno&gt; Pedro &lt;/aluno&gt;   &lt;/projeto&gt; &lt;/projetoPesquisa&gt;</pre>
Pesquisador	Projeto	Aluno					
Maria	Transformando PHP em HTML	Pedro					

Como mostrado, a linguagem XML é extensível e permite que o desenvolvedor crie suas *tags* livremente. Entretanto, esta liberdade pode gerar conflitos, como, por exemplo, em um mesmo documento, podem-se apresentar elementos distintos e ordens variadas, por isso existem linguagens de definição de esquema XML. Estes documentos informam quais são os elementos, atributos e ordem destes. Dentre estas linguagens, destacam-se o DTD e o XML *Schema* (LOSCIO e WAGNER FILHO, 2009), sendo esta última a recomendada pela W3C, por possuir uma melhor definição dos dados e por ser escrita em XML. Outro problema diz respeito a conflitos de nomes de *tags* com sentidos semânticos diferentes. Para evitar esses conflitos, existe o conceito de *namespaces*, em que as *tags* são referenciadas por URIs (*Uniform Resource Identifiers*). O uso de *namespaces* aumenta a flexibilidade de XML *Schema*, permitindo a reutilização de definições feitas em outros esquemas. Pode-se também redefinir tipos declarados em um determinado *namespace* (BRAGANHOLLO e HEUSER, 2001).

Toda a liberdade de estruturação dos dados foi dada ao desenvolvedor através da linguagem XML. Entretanto, isto não permitiu a padronização de metadados. Era necessário um acordo entre os desenvolvedores e usuários envolvidos quanto à nomenclatura e significado das *tags* empregadas a fim de que as mesmas fossem reconhecidas em ambientes heterogêneos (SANTOS, 2002). Esta necessidade foi suprida parcialmente com a especificação dos esquemas XML (W3C, 2011), mas, não suficientemente para representação de modelos de dados muito complexos (SANTOS, 2002). Dessa

forma, apesar de amplamente utilizada em diversas aplicações, XML não possui os requisitos necessários para descrever adequadamente a semântica de um dado. Como forma de suprir esta limitação, surgiram os modelos RDF, RDF(S) e OWL que serão descritos nas seções seguintes.

### 2.2.1 RDF

O RDF não é considerado uma linguagem, mas um modelo de dados que descreve recursos, ou seja, entidades ou algo que possua uma identidade na web (LÓSCIO e WAGNER FILHO, 2009). Pode-se entender o modelo RDF como um modelo baseado em grafos, onde a descrição dos dados é representada através de nós e arestas (nós representam os recursos, e as arestas, os relacionamentos ou predicados entre esses recursos). Para sua implementação, faz-se necessário o uso de uma linguagem que permita expressar esse modelo. A linguagem XML é uma das formas possíveis de representar modelos RDF (TAUBERER, 2008), outras, como, por exemplo, o *turtle*, também podem ser usadas (TURTLE, 2007). Neste cenário, o modelo RDF objetiva tornar a semântica dos recursos e dados Web acessíveis a agentes de softwares, sendo esta a razão pela qual é recomendado no contexto da Web Semântica. Ele utiliza o conceito de sentença ou tripla. Uma sentença é estruturada no formato sujeito, predicado e objeto, onde: (i) o sujeito tem como valor o recurso do qual se quer escrever uma sentença; (ii) o predicado especifica um relacionamento entre sujeito e objeto e é especificado através de propriedades, que permitem relacionar um recurso a dados ou a outros recursos, e (iii) o objeto, que denomina o recurso que se relaciona ao sujeito (LÓSCIO e WAGNER FILHO, 2009). A identificação do recurso a ser descrito é feita utilizando-se o atributo *about* ou *ID* (BRAGNHOLO e HEUSER, 2001), onde o atributo *ID* sinaliza a criação de um novo recurso, e o atributo *about* refere-se a um recurso já existente.

Os principais elementos do RDF são o elemento raiz, `<rdf:RDF>`, onde se define o documento XML para ser um documento RDF e que também contém uma referência às *namespaces* RDF, como também o elemento `<rdf:Description>`, que possui a identificação do recurso a ser descrito e uma lista de propriedades correlacionadas a esse recurso. Na Figura 2, é demonstrada a descrição de dados de uma publicação, dentro do contexto do estudo de caso mostrado na Figura 1.

```
<?xml version="1.0"?>
<!DOCTYPE rdf:RDF [<ENTITY xsd "http://www.w3.org/2001/XMLSchema#">]>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:pes="http://www.gddc.com/pesquisador"
        xmlns:pub="http://www.gddc.com/publicacao"
    >
    <rdf:Description rdf:about="http://www.gddc.com/publicacao/publicacao#1">
        <pub:titulo>Linked Data</pub:titulo>
        <pub:evento>SBBD</pub:evento>
    </rdf:Description>
    <rdf:Description rdf:ID="publicacao#1">
        <pub:titulo rdf:datatype="Literal">Linked Data</pub:titulo>
        <pub:evento rdf:datatype="Literal">SBBD</pub:evento>
    </rdf:Description>
    ...
    <rdf:Statement rdf:about="http://www.gddc.com/publicacao#1/">
        <rdf:subject rdf:resource="http://www.gddc.com/pesquisador/mat#123"/>
        <rdf:predicate rdf:resource="http://www.gddc.com/publica"/>
        <rdf:object rdf:resource="http://www.gddc.com/publicacao#1#titulo"/>
    </rdf:Statement>
    ...
```

&lt;/rdf:RDF&gt;

**Figura 2: Descrição de dados do Estudo de Caso em RDF.**

Aplicações RDF, em maioria, têm necessidade de descrever outras declarações RDF usando o próprio RDF, como, por exemplo, para registrar informações sobre quando as declarações foram feitas, quem as fez, ou outras informações semelhantes. O RDF fornece um vocabulário embutido, destinado para descrever declarações RDF. A descrição de uma instrução usando este vocabulário é chamada de *Reification* da declaração. O vocabulário RDF *Reification* consiste no tipo <rdf:Statement> e as propriedades <rdf:subject>, <rdf:predicate> e <rdf:object>. Uma ilustração é apresentada na Figura 2, onde se faz o uso do vocabulário mencionado para realizar o relacionamento entre os recursos publicação e pesquisador.

Além das características mencionadas, a recomendação RDF também especifica uma série de outras características que ainda não foram aderidas totalmente pela comunidade *Linked Data*. A fim de tornar mais fácil para os usuários consultarem dados, é recomendado o uso de apenas um subconjunto do modelo de dados RDF descrito acima. Em particular, segundo Heath e Bizer (2011), o RDF *Reification* deve ser evitado, pois declarações “reificadas” são bastante pesadas para consulta com a linguagem de consulta SPARQL (W3C, 2011). Em vez de usar o *Reification*, meta-informações deve ser anexadas ao documento da Web que contém as triplas relevantes. RDF *Collection* e RDF *Containers* também são elementos questionados se os dados precisarem ser consultados com SPARQL. Portanto, nos casos em que a ordem relativa de itens em um conjunto não é significativa, o uso de múltiplas triplas com o mesmo predicado é recomendado.

O modelo de dados RDF foi proposto como uma possível solução para limitações de XML, entretanto, suas especificações sobre um determinado recurso são ainda simples, não permitindo, por exemplo, definir relacionamentos de hierarquia. O modelo RDF(S) surgiu com o propósito de criar um vocabulário para a camada RDF, como apresentado a seguir.

### 2.2.2 RDF Schema ou RDF(S)

Segundo Gruber (1995), ontologia é uma especificação explícita e formal de uma conceitualização. Ou seja, ela descreve os conceitos e relacionamentos existentes em um dado domínio ou universo de discurso que é normalmente de conhecimento comum. Isto deve ser descrito de forma que seja legível a um agente de software. Um documento RDF não pode ser considerado, por si só, uma ontologia, pois não é possível especificar em RDF a definição de classes de recursos, de possíveis restrições de classes e relacionamentos e detecção de violação de restrições. O RDF *Schema* visa solucionar tais problemas provendo construtores que permitem especificar formalmente um esquema, definindo classes, subclasses, domínio e restrições de valores das propriedades no contexto da web. A ideia principal é unir RDF(S) + RDF de tal forma que todas as sentenças descritas em RDF obedeçam à semântica descrita no esquema especificado em RDF(S).

O RDF(S) é uma coleção de recursos do RDF que podem ser utilizados para descrever propriedades de outros recursos RDF para definir vocabulários para aplicações específicas (W3C, 2011). O vocabulário do núcleo é definido usando o *namespace* “rdfs”. Assim, RDF(S) é uma extensão do RDF, que descreve recursos como classes, propriedades e valores. Classes em RDF(S) são semelhantes às classes na orientação a objetos (W3C, 2011). Isto provê formas de definir recursos como instâncias de classes e subclasses. Um exemplo de RDF(S) referente ao estudo de caso é mostrado na Figura 3.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns="http://a.com/ontology#">
```



```

<rdfs:Class rdf: ID="Projeto" />
<rdfs:Class rdf: ID="Publicação" />
<rdfs:Class rdf: ID="Pesquisador">
  <rdfs:label> Pesquisador  </rdfs:label>
  <rdfs:comment> Um pesquisador pode ser um professor ou um aluno </rdfs:comment>
</rdfs:Class>
<rdfs:Class rdf: ID="Aluno"> <rdfs:subClassOf rdf:resource="#Pesquisador"/> </rdfs:Class>
<rdfs:Class rdf: ID="Professor"> <rdfs:subClassOf rdf:resource="#Pesquisador"/> </rdfs:Class>
<rdfs:Class rdf: ID="Nível" />
  <Projeto rdf: ID="Linked Data" />
  <Publicação rdf: ID="HTML, XML, RDF: uma comparação" />
  <Aluno rdf: ID="Joãozinho" />
  <Nível rdf: ID="Graduação" />
</rdfs:Class>
<rdf: Property rdf: ID="Escreve" >
<rdfs:subClassOf rdf:resource="http://www.w3.org/2000/01/rdf-schema#" />
  <rdfs:domain rdf: resource="#Aluno" />
  <rdfs: range rdf: resource="#Publicação" />
</rdf: Property>
<rdf: Property rdf: ID="Participa" >
  <rdfs:domain rdf: resource="#Aluno" />
  <rdfs: range rdf: resource="#Projeto" />
</rdf: Property>
</rdf: RDF>

```

**Figura 3: Descrição de dados do Estudo de Caso em RDF(S)**

Apesar do aumento da expressividade advindo da união de RDF com RDF(S), isto ainda não é o suficiente para prover uma rica descrição semântica acerca de um domínio de interesse. É necessário que exista um vocabulário que possa especificar regras mais complexas, como, por exemplo, restringir a cardinalidade de uma propriedade. Diante disso, surgiu o modelo OWL, projetado para descrever a estrutura de um domínio (ontologias) em termos de classes, instâncias, propriedades e axiomas.

### 2.2.3 OWL

A *Ontology Web Language* – OWL é uma linguagem recomendada pela W3C para a especificação de ontologias. Segundo Lóscio e Wagner Filho (2009), os seus idealizadores visavam prover uma linguagem que possuísse a expressividade necessária tanto na descrição de ontologias no suporte à Web Semântica, como em relacionamentos entre domínios e suas propriedades. Com a OWL, podem-se obter inferências a partir dos conceitos e relacionamentos existentes entre esses conceitos. Algumas das características fundamentais do OWL são: (i) restrições de propriedades relacionadas aos possíveis valores assumidos, por exemplo, podemos expressar que um time de futebol exige, no mínimo, onze jogadores para disputar uma partida; (ii) disjunção de classes, por exemplo, gato e cachorro são dois conceitos disjuntos, pois um animal não pode assumir a forma de um gato e um cachorro ao mesmo tempo e (iii) combinação entre classes, onde pode-se criar novos conceitos utilizando uma combinação de conceitos já especificados usando, por exemplo, a união ou interseção destes.

A OWL possui três sub-linguagens que definem níveis de amplitude diferentes para desenvolvimento (HARMALEN e MCGUINESS, 2004): OWL Lite, OWL DL e OWL Full. A OWL Lite é um subconjunto de OWL DL (*Description Logics*), que suporta apenas alguns construtores de OWL. Não suporta, por exemplo, classes enumeradas, declarações de disjunção e cardinalidades arbitrárias. A OWL DL é voltada para dar o máximo de expressividade mantendo a completude computacional

(garante que todas as conclusões serão obtidas) e a decidibilidade (o tempo de processamento é finito). Já a *OWL Full* foi projetada para dar o máximo de expressividade e liberdade sintática do RDF, porém sem garantia computacional. Todas as funcionalidades desenvolvidas para *OWL* são suportadas por ela. Um exemplo que a difere das demais, é que um identificador de *OWL Full* pode atuar ao mesmo tempo como uma classe e como um nome (ou instância).

Exemplos de construtores *OWL* são *owl:ObjectProperty* e *owl:DatatypeProperty*, ambos subclasses de *rdf:Property*. Em *OWL*, classes são definidas com o elemento *owl:Class*, que é subclasse de *<rdfs:Class>*. Considerando o estudo de caso, podemos dizer que Professor é disjuncto da classe Aluno (ambos são subclasses de Pesquisador) utilizando o elemento *owl:disjointWith* e que é igual à classe fictícia Coordenador (também subclasse de Pesquisador) utilizando *owl:equivalentClass*. Já o construtor *owl:sameAs*, que liga um indivíduo a outro, é amplamente utilizado no contexto *Linked Data* e milhões de dados publicados na web possuem ligações deste tipo (HEATH E BIZER, 2011).

Existem identificadores especiais em *OWL* usados para fornecer informações sobre propriedades e seus valores. O recurso *owl:inverseOf* é onde uma propriedade pode ser indicada para ser o inverso da outra propriedade. Por exemplo, se o predicado “éFilhoDe” é o inverso do predicado “éPaiDe”, a afirmativa “João – éPaiDe – Luiz”, permite ao raciocinador inferir que “Luiz - ehFilhoDe – João”. O *OWL* permite também a inclusão de restrições. Estes tipos são usados dentro do contexto de uma *owl:Restriction*. O elemento *owl:onProperty* indica a propriedade que será restrita.

Considerando o estudo de caso, a Figura 6 mostra a descrição do relacionamento entre as classes “pesquisador” e “publicação”, a subclasse “professor”, e as propriedades das classes mencionadas.

```
<?xml version="1.0"?>
<rdf:RDF
  xmlns:owl="http://www.w3.org/2002/07/owl#"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:pes="http://www.gddc.com/pesquisador"
  xmlns:pub="http://www.gddc.com/publicacao"
  xmlns:prof="http://www.gddc.com/professor">
  <owl:Ontology rdf:about="" />
  <owl:Class rdf:ID="#pesquisador" />
  <owl:Class rdf:ID="#publicacao" />
  <owl:ObjectProperty rdf:ID="#escreve">
    <rdfs:domain rdf:resource="#pesquisador" />
    <rdfs:range rdf:resource="#publicacao" />
  </owl:ObjectProperty>
  <owl:DatatypeProperty rdf:ID="nome">
    <rdfs:domain rdf:resource="#pesquisador" />
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="titulo">
    <rdfs:domain rdf:resource="#publicacao" />
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="evento">
    <rdfs:domain rdf:resource="#publicacao" />
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string" />
  </owl:DatatypeProperty>
  <owl:Class rdf:ID="#professor">
```



```

<rdfs:subClassOf>
<owl:Class rdf:resource="#pesquisador">
</rdfs:subClassOf>
</owl:Class>
</rdf:RDF>

```

**Figura 6: Descrição de dados do Estudo de Caso em OWL**

## 2.3 Análise Comparativa

A De acordo com o observado através do estudo de caso e da avaliação dos construtores de RDF, RDF(S) e OWL, percebe-se que o modelo RDF provê um mecanismo capaz de permitir a interoperabilidade ante a multiplicidade de formatos incompatíveis existentes na Web. O modelo RDF descreve “recursos” ou dados da Web que devem ter um identificador, possibilitando que cada recurso crie sua própria meta-informação e torne-a disponível para uso. O RDF(S) introduz construtores que permitem expressar classes, subclasses, domínio e restrições de valores das propriedades. Ou seja, RDF(S) estende as opções de construtores RDF. Os dois modelos podem ser usados conjuntamente para se obter uma maior expressividade na descrição de um conjunto de dados. A linguagem OWL, por sua vez, possui alto poder de expressão e tem como objetivo primordial possibilitar a codificação de ontologias. OWL adiciona mais vocabulário para descrever relações entre classes, cardinalidades, combinações entre classes, disjunções e restrições sobre propriedades (LÓSCIO e WAGNER FILHO, 2009). A ideia é que OWL seja usada conjuntamente com RDF/RDF(S). RDF(S) e OWL, juntas, definem classes e predicados que provêem descrições semânticas de alto nível sobre dados de um domínio de interesse. A Tabela 2 apresenta um resumo das características dos modelos analisados.

**Tabela 2 – Resumo Comparativo dos Modelos XML, RDF, RDF(S) e OWL**

Modelo/Linguagem	Características	Limitações
XML	Permite a descrição dos dados utilizando <i>tags</i> definidas pelo próprio desenvolvedor Baseada numa estrutura de árvore	Representa apenas sintaxe Não diz nada sobre o que as estruturas significam
RDF	Descreve recursos unicamente identificados por URIs Baseado em uma estrutura de grafo Permite que sujeito e objeto troquem de posição (recursos podem ser sujeito ou objeto em definições ou sentidos diferentes)	Não permite definir hierarquias Não permite definir <i>domain</i> e <i>range</i> de propriedades
RDF(S)	Provê um sistema de tipos básicos Permite a construção de estruturas hierárquicas Permite interoperabilidade de metadados de diferentes comunidades Permite codificar, trocar e reusar estruturas de metadados	Linguagem de definição de ontologia simples Não permite especificar restrições de cardinalidade, propriedades transitivas, inversas ou simétricas, nem disjunção de classes
OWL	Aumenta a facilidade de expressar semântica disponível em RDF e RDF(S) Formaliza um domínio pela definição de classes e propriedades destas classes Define instâncias e propriedades sobre as classes Raciocina a respeito de classes e instâncias até o grau permitido pela semântica formal da linguagem	

Dentro da visão da Web Semântica, percebe-se que os modelos RDF, RDF(S) e OWL são camadas de um modelo maior, onde construtores de cada um são integrados com o objetivo de descrever dados e apontar a semântica inerente a esses dados. A Figura 5 mostra um fragmento do

estudo de caso realizado através da sobreposição das três camadas RDF, RDF(S) e OWL. Ela aponta alguns construtores que podem ser usados conjuntamente para descrever os dados da aplicação em questão. Verifica-se, por exemplo, que `rdfs:Class` é a classe de todas as outras classes (a meta-classe), `rdfs:Resource` é a classe de todos os recursos (tudo é um recurso em RDF/RDF(S)), `rdf:Property` é a classe de todas as propriedades, `rdfs:domain` define o domínio de uma propriedade e `rdfs:range` define o contradomínio de uma propriedade. O construtor `owl:sameAs` é apontado como uma forma possível de ligar dados de um conjunto a outro. Neste caso, teríamos pesquisador associado à *researcher* (em outro conjunto de dados).

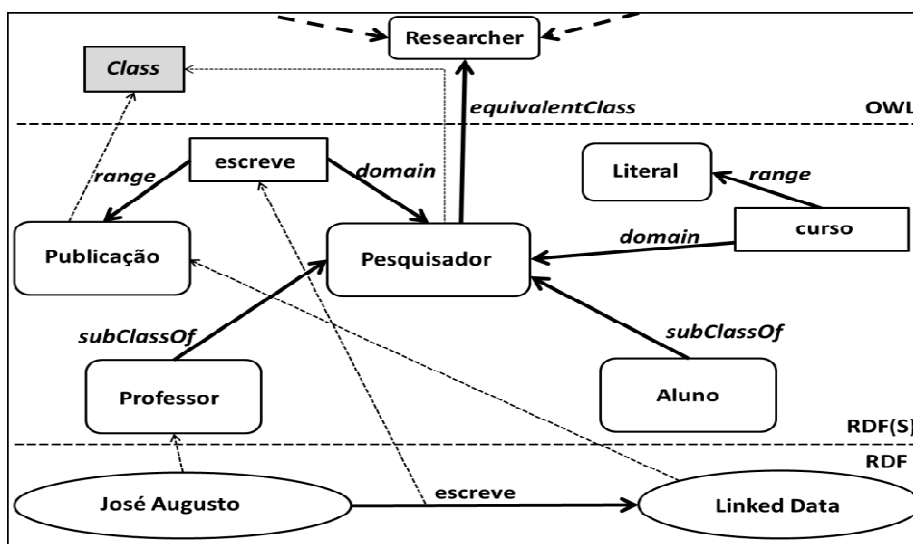


Figura 5: Fragmento do Estudo de Caso

Considerando estas três camadas, no contexto do *Linked Data*, definições de ligações nomeadas em RDF podem ser realizadas a partir do vocabulário presente em domínios específicos descritos normalmente em OWL, tais como *foaf:knows*, *dc:author* ou usufruindo de definições padrão da linguagem OWL, como *owl:sameAs* (visto na Figura 5). Através desses links explícitos, aplicações baseadas em RDF podem segui-los e, a partir daí, descobrir mais dados. Portanto, uma Web onde dados são publicados e ligados usando RDF/RDF(S)/OWL é uma Web onde dados são significativamente mais passíveis de serem descobertos e, conseqüentemente, podem ser mais usados e reutilizados. Esse é o objetivo almejado.

### 3. CONCLUSÃO

Neste trabalho, foi apresentada uma análise comparativa sobre os modelos RDF, RDF(S) e OWL. Para tal, foi utilizado um estudo de caso sobre o qual os construtores de cada modelo foram usados e comparados semanticamente. A partir desse estudo, será desenvolvida uma ferramenta que vai permitir transformar dados publicados atualmente como HTML em RDF/RDF(S)/OWL, seguindo os princípios definidos no *Linked Data*.

### REFERÊNCIAS BIBLIOGRÁFICA

BECKETT, D. Turtle – Terse RDF Triple Language. Disponível em: < <http://www.dajobe.org/2004/01/turtle/2007-09-11/>> Acesso em: 15 set 2011.

BERNERS-LEE, T.; HENDLER, J.; LASSILIA, O. **The semantic web**. Scientific American, 284(5):34–44, Mai 2001.

BIZER C.; HEATH T.; BERNERS-LEE, T. **Linked data - the story so far**. Int. J. Semantic Web Inf. Syst., 5(3):1–22, 2009.

BRAGANHOLO, V.P.; HEUSER, C.A. **XML Schema, RDF(S) e UML: uma Comparação**. In: IDEAS'2001 - Jornadas Iberoamericanas de Ingenieria de Requisitos y Ambientes de Software, 2001, Santo Domingo, Costa Rica. Memorias. Costa Rica : CIT, 2001. v. 1. p. 78-78.

COSTA A.; YAMATE F. **Semantic Lattes: uma ferramenta de consulta baseada em ontologias**. Trabalho de Graduação em Engenharia de Computação - Escola Politécnica, IME/USP, 2009.

GRUBER T. R. **Toward principles for the design of ontologies used for knowledge sharing**. 1995. International Journal Human-Computer Studies Vol. 43, Issues 5-6, November 1995, p.907-928.

HARMELEN, F. V.; MCGUINNESS, D. L. **OWL Web Ontology Language Overview**. Disponível em: <<http://www.w3.org/TR/2004/REC-owl-features-20040210/>> Acesso em: 19 set 2011.

HEATH, T.; BIZER, C. **Linked Data: Evolving the Web into a Global Data Space**. 1. ed. US. Morgan & Claypool Publishers. 2011. 136p.

LÓSCIO, B.; FILHO WAGNER, F. **Web Semântica: Conceitos e Tecnologias**. In: MINI CURSO – UNIVERSIDADE FEDERAL DO PIAUÍ, Capítulo 9, 2009.

SANTOS, Domingos S. A.: **Applying RDF in the data interoperability of different domains**. Campina Grande, 2002, 120p. Dissertação de Mestrado. Universidade Federal da Paraíba.

SILVA, G. C.; LIMA, T. S. **RDF e RDFS na Infra-estrutura de Suporte à Web Semântica**. Disponível em: <<http://www.ic.uff.br/~gsilva/slreic.pdf>> Acesso em: 19 ago 2011.

SOUZA, D.; PIRES, C. E.; KEDAD, Z.; TEDESCO, P.; SALGADO, A. C.: **A Semantic-based Approach for Data Management in a P2P System**. To be published in LNCS Transactions on Large-Scale Data- and Knowledge-Centered Systems, 2011.

TAUBERER, J. **What is RDF and what is good for?** Disponível em: <[www.rdfabout.com/intro](http://www.rdfabout.com/intro)> Acesso em: 31 ago 2011.

W3C. **World Wid Web Consortium**. Disponível em: <<http://www.w3.org>> Acesso em: 31 ago 2011.