

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

ANDRÉ ROLIM BEHR

**OBAA-LEME: Um Editor de Conteúdo de
Metadados para Objetos de Aprendizagem a
partir de Ontologias e Perfis de Aplicação**

Monografia apresentada como requisito parcial para
a obtenção do grau de Bacharel em Ciência da
Computação.

Orientadora: Prof.^a Dra. Rosa Vicari

Porto Alegre
2014

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitor: Prof. Carlos Alexandre Netto

Vice-Reitor: Prof. Rui Vicente Oppermann

Pró-Reitor de Graduação: Prof. Sérgio Roberto Kieling

Diretor do Instituto de Informática: Prof. Luís da Cunha Lamb

Coordenador do Curso de Ciência da Computação: Prof. Raul Fernando Weber

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

AGRADECIMENTOS

Meu caminho tortuoso até o desfecho da formatura de graduação certamente me fará esquecer alguém. Assim, já me antecipo em pedir desculpas!

Não há como iniciar sem agradecer aos meus familiares. A minha mãe Magda, minha parceira, incentivadora e amiga, que me aguenta em todos os momentos, inclusive nos de mal humor. A minha vó Gessy, que constantemente abre caminhos para minha educação e na minha vida. Ao meu avô José, sempre presente em meus pensamentos, minha gratidão eterna pelo exemplo que foi em vida. Ao meu padrasto Celso que considero como um pai, por todo apoio durante todos esses anos de convivência. Aos tios, pelo incentivo nos momentos de vestibulando.

A UFRGS que me disponibilizou um ensino gratuito e de qualidade, tanto na universidade quanto na antiga ETCOM. Além de todas as amizades geradas nesses períodos. Aos colegas de sala 209 no Campus do Vale, obrigado pela convivência diária.

Em especial ao Tiago Primo por todos incentivos durante meu longo período de bolsista, além de preciosas dicas para escrita de artigos! Também a Kelen Bernardi, que junto dele me acolheu muito bem no início da minha jornada de pesquisa científica.

A Prof.^a Dra. Rosa Vicari por todas as portas abertas e oportunidades. Meu agradecimento pela orientação neste trabalho e a esperança de continuar sendo orientado por esta excelente pessoa em um futuro mestrado.

Ao Prof. Dr. Bernhard Mitschang pela experiência proporcionada de intercâmbio na Universidade de Stuttgart. A todas as amizades feitas durante este período.

Aos parceiros de futebol I:=Lógicos, Obrigado Pablo, Lego FC, Informática UFRGS e de outras junções.

Meu muito obrigado a todos que fizeram parte da minha história em algum dos caminhos trilhados.

RESUMO

Atualmente, grande parte das atividades ligadas à educação está diretamente atrelada a meios digitais. A Web vem se firmando dentre as alternativas para que estudantes venham a obter ambientes mais ricos de aprendizagem. Progressivamente, a Web vem sendo estendida pela Web Semântica, com o objetivo das máquinas poderem ter um comportamento mais ativo, assim como os humanos.

Dentre as tecnologias que disponibilizam uma interoperabilidade de máquinas, podemos citar as recomendações da W3C: Extensible Markup Language (XML), Resource Description Framework (RDF) e Web Ontology Language (OWL). A OWL é a linguagem mais forte e com a maior possibilidade de ser interpretada por máquinas dentre elas.

Deste modo, o presente trabalho vem a utilizar a OWL para propor um editor de metadados para objetos de aprendizagem utilizados para educação e descrevê-los em forma de ontologias. O objetivo central é tanto a adaptação de objetos de aprendizagem já existentes quanto novos a este novo contexto da Web Semântica.

Esta descrição dos metadados de objetos de aprendizagem se dá a partir dos Padrões OBAA e Videoaula@RNP de Metadados. A partir deles, a semântica dos metadados é extraída para a elaboração de ontologias para a descrição de metadados dos objetos.

Com isso, estes podem ser descritos como indivíduos de ontologias. A partir desta descrição pode-se realizar inferências e verificar a consistência de dados.

O editor proposto objetiva abstrair a gerência de ontologias para o usuário. Além disso, ele vem a se utilizar de perfis de aplicação como meio de facilitar o carregamento e o preenchimento dos metadados.

Palavras-chave: Web semântica. Objeto de aprendizagem. Metadados educacionais. Ontologia. Perfil de aplicação.

OBAA-LEME: A Learning Object Metadata Content Editor based on Ontologies and Application Profiles

ABSTRACT

Nowadays, the major part of the educational activities is directly linked to digital ways. The Web is playing an important role as an alternative to students get a richer learning environment. Gradually, the Web has been extended by the Semantic Web, aiming that machines can be a more active behavior, like humans.

Among technologies that provide interoperability to machines, the W3C recommendations can be cited: Extensible Markup Language (XML), Resource Description Framework (RDF) and Ontology Web Language (OWL). OWL is the strongest and with greater possibility to be interpreted among them.

In this way, the present work is going to use OWL to propose a metadata content editor to learning objects used in education and describe them as ontologies. The main objective is both to adapt existing learning objects as new ones to this new Semantic Web context.

This metadata description of learning objects is made through the OBAA and Videoaula@RNP Metadata Standards. The metadata semantic is extracted from them to create ontologies to describe the object metadata.

Therewith, it can be described as ontology individuals. Through this description, inference and data consistency verification can be made.

The propose editor aims to abstract the ontologies management to the user. Furthermore, it uses application profiles in the way to facilitate the metadata loading and filling.

Keywords: Semantic web. Learning object. Educational metadata. Ontology. Application profile

LISTA DE FIGURAS

Figura 2.1: Exemplo de estrutura XML.	16
Figura 2.2: Exemplo de estrutura RDF.....	17
Figura 2.3: Extensão do grupo de metadados Educational.....	20
Figura 2.4: Extensão do grupo de metadados Technical.....	20
Figura 2.5: Quadro de elementos de metadados do padrão Videoaula@RNP	22
Figura 2.6: Quadro de cardinalidades dos metadados do padrão Videoaula@RNP	23
Figura 2.7: Quadro dos elementos de metadados que compõe o perfil OBAA-LITE ...	24
Figura 2.8: Quadro comparativo entre editores de conteúdos de metadados	26
Figura 2.9: Número de utilizantes por editores de ontologias.....	27
Figura 3.1: Hierarquia geral LOM e detalhamento do grupo de metadados Rights.....	28
Figura 3.2: Lógica para espaço de valores do metadado Name	31
Figura 3.3: Domínio e escopo da propriedade de objeto hasOrComposite	31
Figura 3.4: Cardinalidade da propriedade de dados lifeCycle.Contribute.Role	32
Figura 4.1: Hierarquia de classes VideoAula	34
Figura 4.2: Novas propriedades de dados da ontologia VideoAula	35
Figura 4.3: Novas propriedades de objetos da ontologia VideoAula	36
Figura 5.1: Exemplo de cadeia de propriedade de objetos.....	38
Figura 5.2: Classe de equivalência OBAA-LITE.....	38
Figura 5.3: Classe de equivalência OBAA-VideoAula.....	39
Figura 5.4: Classe de equivalência OBAA-DTV.....	40
Figura 6.1: Representação geral do OA Célula de Combustível Alcalina	41
Figura 6.2: Representação geral do OA Módulo 1	42
Figura 6.3: Representação geral do OA Ramis	43
Figura 6.4: Propriedades de dados dos indivíduos do OA Ramis	44
Figura 6.5: Representação geral do OA TV Educativa	45
Figura 6.6: Propriedades de dados dos indivíduos do OA TV Educativa	45
Figura 6.7: Exemplo de inconsistência por tipo de valor	46
Figura 6.8: Exemplo de inconsistência por espaço de valor	47
Figura 6.9: Exemplo de inconsistência por valor condicional	48
Figura 6.10: Exemplo de inconsistência por cardinalidade	48
Figura 6.11: Exemplo de inferência sobre perfil OBAA-LITE.....	49
Figura 6.12: Exemplo de inferência sobre perfil OBAA-VideoAula.....	50

Figura 6.13: Exemplo de inferência sobre perfil OBAA-DTV	50
Figura 7.1: Janela de escolha de perfil de aplicação.....	51
Figura 7.2: Janela principal a partir do perfil OBAA-LITE	53
Figura 7.3: Visualização do OA Célula de Combustível Alcalina	54
Figura 7.4: Exemplo de mensagem de inconsistência	55
Figura 8.1: Captura de tela do preenchimento dos metadados do OA Ábaco.....	56
Figura 8.2: Captura de tela do preenchimento dos metaados do OA A Cartomante.....	57
Figura 8.3: Visualização do OA Módulo 2	58
Figura 9.1: Exemplo de conjunto de atributos de arquivo.....	60
Figura 9.2: Exemplo de reuso de indivíduos de OAs	61
Figura 9.3: Exemplo de versionamento de OAs.....	61

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
DC	Dublin Core
DTV	Digital Television
GIA	Grupo de Pesquisa em Inteligência Artificial
IDE	Integrated Development Environment
IEEE	Institute of Electrical and Electronics Engineers
IEEE-LOM	IEEE Standard for Learning Object Metadata
IMS	Instructional Management System
IMS-LOM	IMS Standard for Learning Object Metadata
IRI	Internationalized Resource Identifier
LOM	Learning Object Metadata
OA	Objeto de Aprendizagem
OBAA	Padrão de Objetos de Aprendizagem Baseados em Agentes
OWL	Web Ontology Language
RDF	Resource Description Framework
RNP	Rede Nacional de Pesquisa
SCORM	Sharable Content Object Reference Model
URI	Uniform Resource Identifier
W3C	World Wide Web Consortium
WS	Web Semântica
WWW	World Wide Web
XML	Extensible Markup Language

SUMÁRIO

RESUMO	4
ABSTRACT	5
LISTA DE FIGURAS	6
LISTA DE ABREVIATURAS E SIGLAS	8
SUMÁRIO	9
1 INTRODUÇÃO	11
1.1 Motivação	12
1.2 Objetivos	12
1.3 Organização do Texto	13
2 DEFINIÇÕES E CONCEITOS PRINCIPAIS	14
2.1 Web	14
2.2 Web Semântica	15
2.2.1 XML	16
2.2.2 RDF	16
2.2.3 OWL	17
2.3 Ontologias	18
2.4 Padrão OBAA de Metadados	19
2.5 Padrão Videoaula@RNP de Metadados	21
2.6 Perfis de Aplicação	21
2.6.1 OBAA-FULL	23
2.6.2 OBAA-LITE	23
2.6.3 OBAA-VideoAula	24
2.6.4 OBAA-DTV	24
2.7 Trabalhos Relacionados	24
2.8 Ferramental	26
2.8.1 Protégé	26
2.8.2 OWLAPI	27
3 UMA ONTOLOGIA PARA O PADRÃO OBAA DE METADADOS	28
3.1 Hierarquia de Classes	28
3.2 Propriedades	30
3.2.1 Propriedades de Dados	30
3.2.2 Propriedades de Objetos	31
3.3 Cardinalidades	32
3.4 Indivíduos	32
3.5 Anotações e Documetação	33
4 UMA ONTOLOGIA PARA O PADRÃO VIDEOAULA@RNP DE META-DADOS	34
4.1 Hierarquia de Classes	34
4.2 Propriedades	35
4.2.1 Propriedades de Dados	35
4.2.2 Propriedades de Objetos	35
4.3 Cardinalidades	36
4.4 Indivíduos	36
4.5 Anotações e Documetação	36
5 ONTOLOGIAS PARA PERFIS DE APLICAÇÃO	37
5.1 Perfil OBAA-LITE	38
5.2 Perfil OBAA-VideoAula	38

5.3	Perfil OBAA-DTV	40
6	RESULTADOS PARCIAIS	41
6.1	Representação de Objetos de Aprendizagem como Ontologias	41
6.1.1	Célula Combustível Alcalina.....	41
6.1.2	Apresentação do Serviço Videoaula@RNP: Módulo 1	42
6.1.3	Ramis	43
6.1.4	TV Educativa.....	44
6.2	Verificação de Consistência	44
6.2.1	Tipo de Valor.....	46
6.2.2	Espaço de Valores	46
6.2.3	Valor Condicional	46
6.2.4	Cardinalidade.....	47
6.3	Processo de Inferência sobre Perfis de Aplicação	47
6.3.1	Perfil OBAA-LITE.....	49
6.3.2	Perfil OBAA-VideoAula	49
6.3.3	Perfil OBAA-DTV	49
7	OBAA-LEME	51
7.1	Escolha do Perfil de Aplicação	51
7.2	Preenchimento de Conteúdo	52
7.3	Exportação	53
7.4	Visualização	53
7.5	Verificação de Consistência	54
7.6	Publicação	55
8	RESULTADOS FINAIS	56
8.1	Ábaco	56
8.2	A Cartomante	56
8.3	Infraestrutura do Serviço Videoaula@RNP: Módulo 2	57
9	CONCLUSÃO	59
9.1	Trabalhos Futuros	59
9.1.1	Autopreenchimento do Conteúdo de Metadados.....	59
9.1.2	Reuso e Versionamento de Objetos de Aprendizagem	60
	REFERÊNCIAS	62
	APÊNDICE A MAPA MENTAL DO PADRÃO DE METADADOS VIDEOAULA@RNP	64
	ANEXO A MAPA MENTAL DO PADRÃO DE METADADOS OBAA	65

1 INTRODUÇÃO

A Web foi proposta em 1989 por Tim Berners-Lee (BERNERS-LEE, 1996). Em 13 de março de 2014, a página web WorldWideWebSize¹ contabilizou pelo menos 2.09 bilhões de páginas indexadas. Este número é estimado pelo número de páginas indexadas pelo Google², Bing³ e Yahoo Search⁴.

Com intuito principal em gestão de documentos, a Web permite aos utilizadores clicarem em links para abrir arquivos em computadores localizados em qualquer lugar do mundo conectados pela internet. Além disso, a publicação de conteúdos se realiza de forma gratuita.

Progressivamente, a Web tem sido estendida pela Web Semântica (WS). Esta nova Web, compatível com a Web atual, visa uma maior interação das máquinas com o conteúdo disponível. Esta interação se dá através de uma semântica bem definida e de identificadores universais para os conteúdos. A interoperabilidade entre conteúdos é recomendada pela utilização de ontologias, através da OWL, uma linguagem de ontologias para Web.

A Web Ontology Language (OWL) está em crescente adoção, mas ainda necessita de ferramentas e ambientes de desenvolvimento de software que suportem sua produção e aplicação. Estas estão começando a surgir, mas como ainda existem em pequena quantidade significa gerar rotineira e passivamente anotações de WS usando esta ou outra linguagem no momento de criação ou uso de conteúdo (SHADBOLT et al., 2006).

A Web está diretamente relacionada com um possível futuro da educação. De acordo com o Engineering Education Report⁵, as três principais alternativas, em ordem decrescente de relevância, a serem aplicadas em um futuro próximo são: arquiteturas e plataformas de ensino online; livros eletrônicos e bibliotecas digitais; e cursos online abertos massivos.

Professores têm experimentado novos métodos, se afastando das aulas em classe tradicionais e do conjunto de problemas extraclasse. Muitos deles estão agora incorporando componentes de ensino online, os quais tem sido apresentados para melhorar tanto as atitudes de alunos quanto sua performance acadêmica (PRUNUSKE et al., 2012).

¹<http://www.worldwidewebsize.com/>

²<http://www.google.com>

³<http://www.bing.com>

⁴<http://search.yahoo.com>

⁵http://ohm.ieec.uned.es/eer/consulta_years.php

1.1 Motivação

As principais alternativas para o futuro próximo da educação anteriormente citadas se utilizam de um ferramental básico: objeto de aprendizagem (OA). Um OA é definido como uma entidade, digital ou não, que possa ser usada para o ensino, educação ou treinamento (COMMITTEE 2002). OAs são comumente descritos na Web por um conjunto de metadados, com o intuito de facilitar sua busca e reuso.

Metadados são dados utilizados para descrever outros dados ou simplesmente definidos como dados sobre dados (BARGMEYER et al., 2000). Metadados são informações sobre um objeto, seja ele físico ou digital. Como o número de objetos cresce exponencialmente e as necessidades de aprender expandem-se drasticamente igual, a falta de informações ou metadados sobre objetos tem uma restrição crítica e fundamental na habilidade de descoberta, gerenciamento e uso de objetos (COMMITTEE 2002).

Portanto, padrões de metadados constituem de material importante para a descrição de objetos de aprendizagem na Web e podem ser reutilizados na Web Semântica. Através deles, é possível definir a estrutura hierárquica e os axiomas para a definição de ontologias que descrevem objetos de aprendizagem nesta nova Web.

Com a crescente demanda de objetos de aprendizagem para serem utilizados em ambientes de ensino online, em conjunto com esta nova metodologia de descrição de conteúdos na Web Semântica, produziu-se um novo cenário de criação e descrição de material educacional online. Assim, o desenvolvimento de novas ferramentas que tornem compatíveis os conteúdos já existentes e que descrevam novos conteúdos de forma semântica é necessário.

1.2 Objetivos

A partir das motivações anteriormente citadas, o presente trabalho pretende:

1. Definir ontologias para descrição dos metadados de objetos de aprendizagem através dos relatórios técnicos de padrões de metadados existentes no âmbito do projeto OBAA - OBjetos de Aprendizagem baseados em Agentes (ver Seção 2.4);
2. Definir ontologias para perfis de aplicação, de forma a facilitar o preenchimento do conteúdo dos metadados;
3. Desenvolver um protótipo de ferramenta que descreva um objeto de aprendizagem através de uma ontologia a partir de um perfil de aplicação.

1.2 Organização do Texto

O presente trabalho aborda primeiramente no Capítulo 2 os principais conceitos que embasam este trabalho. Os Capítulos 3, 4 e 5 descrevem a metodologia para o desenvolvimento de ontologias utilizadas para a obtenção dos resultados parciais. Exemplos de resultados parciais são apresentados no Capítulo 6. A seguir, a concepção do protótipo da ferramenta almejada é apresentada no Capítulo 7. Alguns resultados de publicação de OAs obtidos através da implementação da ferramenta são apresentados no Capítulo 8. Por fim, as conclusões e trabalhos futuros são apresentados no Capítulo 9.

2 DEFINIÇÕES E CONCEITOS PRINCIPAIS

Nesta seção serão apresentados definições e conceitos principais que o autor julga pertinente para o entendimento do trabalho. Cabe salientar que estes não estão em sua totalidade, cabendo uma consulta mais ampla em outras bibliografias para um total entendimento.

2.1 Web

Em 19 de março de 1989, a Web tem sua proposta de arquitetura apresentada por Tim Berners-Lee. Também é denominada como World Wide Web ou WWW.

A World Wide Web é simplesmente definida como o universo da informação global acessível por redes. É um espaço abstrato no qual pessoas podem interagir e é principalmente povoada por páginas interligadas de texto, imagens e animações, com sons ocasionais, vídeos e mundos de três dimensões (BERNERS-LEE, 1996).

O projeto da Web foi baseado em alguns critérios (BERNERS-LEE, 1996):

- Um sistema de informação deve ser capaz de gravar associações randômicas entre objetos arbitrários, não como a maioria dos sistemas de banco de dados.
- Se dois conjuntos de usuários começam a usar o sistema independentemente, criar um link de um sistema para outro deveria ser um esforço incremental, não necessitando operações não escaláveis, como uma fusão de links de banco de dados.
- Tentativas de limitar usuários a linguagens particulares ou sistemas operacionais são sentenciadas ao insucesso.
- Informação deve estar disponível em todas as plataformas.
- Tentativas de forçar usuários a lidar com a informação da mesma forma que os computadores lidam estão condenadas ao insucesso.
- Para garantir correção, a informação deve ser de fácil inserção e estar correta.

Através da evolução da Web, nem sempre com consenso da comunidade, foram sugeridas subdivisões (ou eras):

- Web 1.0: Web estática; somente leitura; aplicações proprietárias; até final da década de 90.

- Web 2.0: Web de blogs, serviços Web, mapeamento, mesclagem de informações¹, etc.
- Web 3.0: Web de dados; Web Semântica.

Atualmente a Web é regida pelo World Wide Web Consortium (W3C), criado em 1994. A partir deste consórcio, companhias e pessoas podem trabalhar em conjunto para a criação de especificações de uso comum.

A maioria da informação contida na Web encontra-se dispersa. A Web Semântica se propõe a conter esta fragmentação, integrando os dados por meio de ontologias.

2.2 Web Semântica

A maioria dos conteúdos da Web atualmente é projetada para leitura de humanos, não para que programas de computadores possam manipulá-los significativamente. Computadores podem habilmente analisar páginas Web para diagramação e processamento de rotinas - um cabeçalho, um link para outra página - mas em geral, computadores não tem um caminho confiável para processar semânticas (BERNERS-LEE et al., 2001).

Em seu artigo em 1996, Tim Berners-Lee já almejava uma maior interação das máquinas na Web:

Com o crescimento das aplicações comerciais na Web, a análise de máquina permitiria aos computadores facilitar os negócios, servindo como agentes, com o poder de agir financeiramente. Para fazer isso, dados potencialmente úteis na Web deveriam estar disponíveis em uma forma entendível por máquinas, com semântica definida [...] (BERNERS-LEE, 1996, p. 76).

Uma proposta oficial da Web Semântica foi proposta em 2001 por Tim Berners-Lee. Esta nova Web não seria uma proposta em separado da Web corrente. A WS seria uma extensão da atual Web, tornando-as compatíveis.

A WS é definida como uma Web de informação acionável. Esta informação é derivada de dados através de uma teoria semântica para a interpretação de símbolos. A teoria semântica provê uma descrição do “significado” na qual a conexão lógica de termos estabelece interoperabilidade entre sistemas (SHADBOLT, 2006).

Dentre as tecnologias empregadas na Web Semântica, destacam-se: a Extensible Markup Language (XML), o Resource Description Framework e a Web Ontology Language

¹Em inglês: mashups.

(OWL), todas sendo recomendadas pela W3C.

2.2.1 XML

O XML permite a criação de etiquetas². Estas etiquetas servem para anotar páginas Web ou seções de texto encontrados nas mesmas. As etiquetas ficam ocultas ao usuário e permitem criar uma estrutura arbitrária de organização. Atualmente, existe uma grande gama de dados XML obtidos através de banco de dados.

A Figura 2.1 exemplifica um trecho de arquivo XML. Esta organização não fornece nenhuma informação a respeito da semântica dos dados, como por exemplo, quem de fato é Tove e Jani, ou que intervalo de tempo “this weekend” representa. Assim, um dos temas de pesquisas é a conversão de dados XML em conteúdo RDF.

Figura 2.1 – Exemplo de estrutura XML

```
<note>
  <to>Tove</to>
  <from>Jani</from>
  <heading>Reminder</heading>
  <body>Don't forget me this weekend!</body>
</note>
```

Fonte: <http://www.w3schools.com/xml/note.xml>.

2.2.2 RDF

O conteúdo RDF foi primeiramente definido pela W3C em 1997, tornando-se uma recomendação em 1999. Este conteúdo permite uma linguagem representação baseada em triplas e objetiva uma representação minimalista de conhecimento para a Web. Uma tripla é composta por um sujeito, um predicado e um objeto. Dependendo do componente, este pode ser uma IRI (string UNICODE conforme RFC 3987), um literal (string, números, datas, etc.) ou um nodo em branco (disjunção de IRI e literal):

- Sujeito: IRI ou nodo em branco;
- Predicado: IRI;
- Objeto: IRI, literal ou nodo em branco.

²Em inglês: tags.

Figura 2.2 – Exemplo de estrutura RDF

```

<?xml version="1.0"?>

<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:si="http://www.w3schools.com/rdf/">

<rdf:Description rdf:about="http://www.w3schools.com">
  <si:title>W3Schools</si:title>
  <si:author>Jan Egil Refsnes</si:author>
</rdf:Description>

</rdf:RDF>

```

Fonte: http://www.w3schools.com/webservices/ws_rdf_intro.asp.

A informação em RDF é representada a partir do formato XML. A Figura 2.2 ilustra um exemplo, onde o sujeito “http://www.w3schools.com” possui o predicado “http://www.w3schools.com/rdf/title” com o literal “W3Schools” e o predicado “http://www.w3schools.com/rdf/author” com o literal “Jan Egil Refsnes”.

2.2.3 OWL

A recomendação da W3C apresenta três versões de OWL, dependendo do grau de poder de expressão requerido. A ideia principal da OWL é possibilitar uma representação eficiente de ontologias que também são responsáveis por processos de decisão. A OWL verifica se uma ontologia é logicamente consistente ou determina se um conceito em particular se encaixa dentro de uma ontologia. A OWL se utiliza da conexão provida pelo RDF para permitir que ontologias sejam distribuídas através de sistemas. Ontologias podem se tornar distribuídas, pois a OWL permite ontologias referenciar termos em outras ontologias. Neste sentido, OWL é especialmente construída para a Web e WS (SHADBOLT, 2006).

Além da verificação de consistência, a linguagem OWL é capaz de realizar inferências. Dentre as mais comuns, destaca-se a subsunção e a classificação.

As três versões correntes da OWL são: DL, Lite e Full. O presente trabalho não visa à explanação de cada uma delas. Este se utiliza da OWL 2, através da lógica descritiva³ (DL).

No decorrer do presente trabalho, links para arquivos OWL serão apresentados. Portanto, estes também servem de exemplo de aplicação para esta linguagem.

³ Em inglês: description logic.

2.3 Ontologias

Studer et al. definiu o termo ontologia como sendo “uma especificação formal explícita de uma conceptualização compartilhada” (1998, p. 25), unificando as definições de Gruber e Burst.

Ontologias são caminhos para um compromisso explícito para o compartilhamento de significado entre comunidades interessadas, mas qualquer um pode usar estas ontologias para descrever seus próprios dados. Similarmente, qualquer um pode estender ou reusar elementos de uma ontologia se desejável (SHADBOLT et al., 2006).

A consolidação de ontologias se dá através da união de “pequenas” ontologias desenvolvidas por pequenos grupos ou pelo seu lançamento através de grandes consórcios e organizações que atuam sobre determinados padrões, capazes de confeccionar ontologias com uma rigorosa formalidade.

O desenvolvimento de ontologias está inserido em um campo de pesquisa denominado engenharia de ontologias, através de um processo cuidadoso e explícito que objetiva remover ambiguidade. Ela é uma metodologia que nos dá uma base lógica projetável, permitindo a acumulação de conhecimento e suporte ao reuso e compartilhamento de conhecimento. Além disso, ela provê uma representação de conceito explícita em diferentes níveis de abstração. O propósito final da engenharia de ontologias é: “prover uma base para a construção de modelos e todas as coisas nas quais a ciência da informação está interessada no mundo” (MIZOGUCHI; IKEDA, 1996).

O ciclo de desenvolvimento de uma ontologia é dividido em etapas: projeto, avaliação, manutenção, implantação, mapeamento, integração, compartilhamento e reuso. A construção de ontologias é árdua, longa e custosa, particularmente se o objetivo é uma ontologia que seja suficientemente formal para permitir uma inferência automatizada. No campo de engenharia de ontologias, não existe classificação quanto ao domínio de problemas e nem caracterização de ontologias, para avaliar e comparar a sua adequação e performance (GRÜNINGER; LEE, 2002).

O presente trabalho não objetiva se aprofundar nas etapas do desenvolvimento de ontologias. Cabe salientar que o processo utilizado para a elaboração das ontologias a seguir foi baseado no processo “top-down”, contido no trabalho de Noy e McGuinness (2001).

Uma grande quantidade de dados interconectados é relevante para que a WS seja concretizada. Estes dados devem seguir um padrão, serem alcançáveis e gerenciáveis por

ferramentas. Uma maior integração de dados é obtida através da adoção de ontologias como conceptualizações compartilhadas.

As ontologias irão fornecer a semântica para que a WS seja desenvolvida, gerenciada e endossada por comunidades. A WS ideal faz reuso substancial de dados e ontologias existentes. É um espaço de informação interconectada na qual os dados são enriquecidos e adicionados (SHADBOLT et al., 2006).

2.4 Padrão OBAA de Metadados

O Padrão OBAA especifica uma proposta de padrão de metadados para OAs compatível com o padrão IEEE-LOM (COMMITTEE, 2002). Além disso, permite a interoperabilidade desses objetos nas plataformas Web, TV Digital e dispositivos móveis. Requisitos de acessibilidade para pessoas com necessidades especiais e o registro de informações educacionais específicas do contexto brasileiro também são suportados. A amplitude do conjunto deve-se a sua completude com relação aos objetivos de interoperabilidade e adequação ao cenário educacional brasileiro (VICARI et al., 2009).

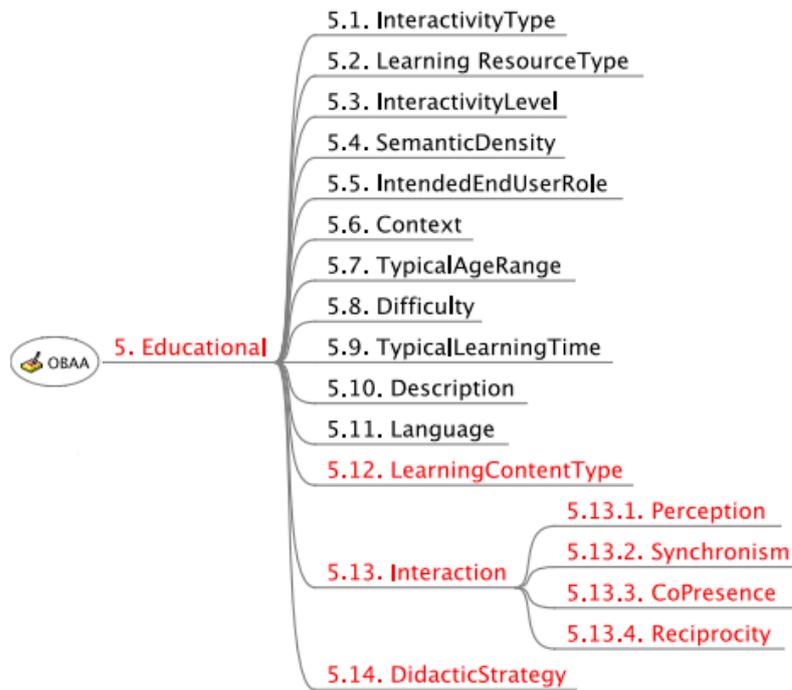
O conjunto de metadados Technical (item número 4 do IEEE-LOM) é estendido com metadados para promover a interoperabilidade com TV Digital. O conjunto Educational (item número 5 do IEEE-LOM) é ampliado de forma a contemplar aspectos da educação no Brasil. Acrescenta-se também os conjuntos Accessibility (item número 10) e SegmentInformationTable (item número 11). O conjunto Accessibility propõe a utilização do padrão IMS AccessForAll (CONSORTIUM, 2004), o qual contempla grande parte dos requisitos para a utilização de diferentes dispositivos para cidadãos com necessidades especiais, bem como alguns requisitos de usabilidade. Finalmente, o item 11 propõe o reuso e extensões ao padrão TV-ANYTIME, possibilitando indexar segmentos de um objeto de aprendizagem por assuntos, destaques, atividades, etc., esteja esse objeto no formato de vídeo, hiperdocumento ou outro qualquer (VICARI et al., 2009).

Um trecho adaptado do mapa mental do Padrão OBAA⁴ é representado na Figura 2.3. Ela ilustra a extensão do grupo de metadados Educational do padrão IEEE-LOM, onde os metadados adicionados estão representados em vermelho.

A extensão do grupo de metadados Technical é representada no trecho de mapa mental da Figura 2.4. Os metadados anexados também estão em vermelho.

⁴http://www.portalobaa.org/padrao-obaa/OBAA-v2.mm/at_download/file

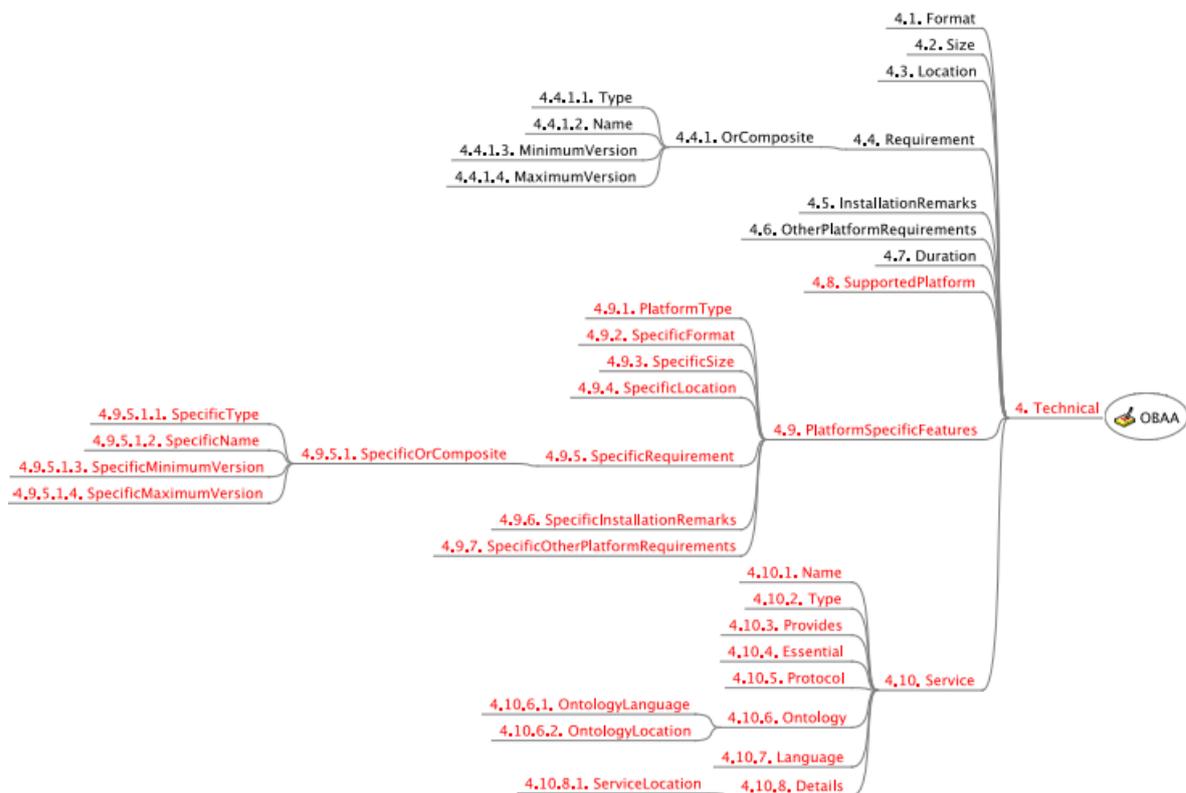
Figura 2.3 – Extensão do grupo de metadados Educational



Fonte: http://www.portalobaa.org/padrao-obaa/OBAA-v2.mm/at_download/file.

O mapa mental completo do Padrão OBAA pode ser encontrado no Anexo A. Os conjuntos de metadados Accessibility e SegmentInformationTable são ilustrados neste mapa.

Figura 2.4 – Extensão do grupo de metadados Technical



Fonte: http://www.portalobaa.org/padrao-obaa/OBAA-v2.mm/at_download/file.

2.5 Padrão Videoaula@RNP de Metadados

O projeto Videoaula@RNP é desenvolvido em parceria com a Rede Nacional de Pesquisa (RNP) e tem por objetivo desenvolver um produto com o qual usuários possam desenvolver vídeos com fim pedagógico. Seu objetivo é disponibilizar material didático, em suma OAs, compostos por múltiplas mídias sincronizadas. A característica educacional de tal serviço aproximou o referido projeto ao padrão de metadados OBAA. Além das características educacionais, o serviço visa compatibilidade com os repositórios de conteúdos educacionais já existentes (PRIMO, 2013).

A compatibilização de tal serviço para repositórios de conteúdos educacionais se dá a partir da utilização do Padrão OBAA. Esta aproximação conduziu o processo de engenharia de conhecimento que culminou na delimitação de um subconjunto de metadados OBAA para descrever uma Videoaula, bem como, metadados específicos para este domínio (PRIMO, 2013).

Os novos metadados foram adicionados aos grupos Educational e Technical. A Figura 2.5 ilustra um quadro com todos os elementos que compõe o Padrão Videoaula@RNP, a numeração dos novos elementos é sugerida. Uma outra visualização destes metadados é encontrada no Apêndice A, a partir de um mapa mental.

Por sua vez, a Figura 2.6 apresenta um quadro com a cardinalidade de cada novo metadado. O valor “N” representa qualquer valor positivo que se queira.

2.6 Perfis de Aplicação

Um perfil de aplicação é um conjunto de elementos de metadados selecionados a partir de um ou mais esquemas de metadados e combinados em um esquema composto. Perfis de aplicação fornecem os meios para expressar princípios de modularidade e extensibilidade. O propósito de um perfil de aplicação é adaptar ou combinar esquemas existentes em um pacote que é adaptado às exigências funcionais de uma determinada aplicação, mantendo a interoperabilidade com os esquemas de base original. Parte de tal adaptação pode incluir a elaboração de elementos de metadados locais que têm importância em uma determinada comunidade ou organização, mas que não se espera que sejam importantes em um contexto mais amplo (DUVAL et al., 2002).

Figura 2.5 – Quadro de elementos de metadados do padrão Videoaula@RNP

<i>Elementos</i>							
1 General	2 Lifecycle	3 MetaMeta data	4 Technical		5 Educational	6 Rights	7 Relation
1.1 Identifier	2.2 Status	3.1 Identifier	4.1 Format	4.10.2 Type	5.2 LearningResourceType	6.2 CopyrightAndOtherRestrictions	7.1 Kind
1.1.1 Catalog	2.3 Contributor	3.1.1 Catalog	4.2 Size	4.10.3 Provides	5.5 IntendedEndUserRole	6.3 Description	7.2 Resource
1.1.2 Entry	2.3.1 Role	3.1.2 Entry	4.4 Requirement	4.10.4 Essential	5.6 Context		7.2.1 Identifier
1.2 Title	2.3.2 Entity	3.2 Contributor	4.4.1 OrComposi	4.10.5 Protocol	5.9 TypicalLearningType		7.2.1.1 Catalog
1.3 Language	2.3.3 Date	3.2.1 Role	4.4.1.1 Type	4.10.8 Details	5.10 Description		7.2.1.2 Entry
1.4 Description		3.2.2 Entity	4.4.1.2 Name	4.10.8.1 ServiceLocation	5.15 Course		7.2.2 Description
1.5 Keyword		3.2.3 Date	4.7 Duration	4.11 Bitrate	5.15.1 Title		
		3.3 MetadataSchema	4.8 SupportedPlatform	4.12 Thumbnail	5.15.2 Code		
			4.9 PlatformSpecificFeatures	4.12.1 Catalog	5.16 Program		
			4.9.8 SpecificAspectRatio	4.12.2 Entry			
			4.9.9 SpecificStandardResolution	4.13 RelatedMedia			
			4.10 Service	4.13.1 Catalog			
			4.10.1 Name	4.13.2 Entry			

Fonte: Elaborado pelo autor.

Como o padrão OBAA de metadados possui uma quantidade considerável de metadados em sua composição, perfis de aplicação foram elaborados para facilitar sua utilização. Estes perfis são definidos através de estudos de utilização em áreas específicas.

Figura 2.6 – Quadro de cardinalidades dos metadados do padrão Videoaula@RNP

<i>Item de metadado</i>	<i>Cardinalidade</i>
4 Technical	1
4.11 Bitrate	1
4.12 Thumbnail	N
4.12.1 Catalog	1
4.12.2 Entry	1
4.13 RelatedMedia	3
4.13.1 Catalog	1
4.13.2 Entry	1
5 Educational	1
5.15 Course	N
5.15.1 Title	N
5.15.2 Code	N
5.16 Program	N

Fonte: Elaborado pelo autor.

2.6.1 OBAA-FULL

O perfil de aplicação OBAA-FULL contém o conjunto completo de metadados OBAA. A versão completa se trata da versão atualmente proposta, com elementos que possibilitam descrever objetos de aprendizagem multiplataforma e com requisitos de acessibilidade (SILVA, 2011).

O Padrão OBAA é considerado amplo devido ao seu objetivo de compatibilidade com o cenário educacional brasileiro. Devido a sua extensão, uma alta granularidade de indivíduos seria obtida em uma ontologia a partir da metodologia proposta neste trabalho. Este fator acaba por prejudicar a execução do motor de inferência, exemplificado no decorrer da pesquisa. Portanto, uma ontologia para perfil OBAA-FULL não será especificada neste trabalho.

2.6.2 OBAA-LITE

O perfil de aplicação OBAA-LITE é uma proposta de um conjunto resumido de metadados OBAA, baseados nos elementos utilizados de fato nos grupos focal estruturado e semiestruturado. A versão reduzida visa fornecer um modelo simplificado, fácil de ser descrito pelos desenvolvedores de objetos de aprendizagem e implantados nos repositórios (SILVA, 2011).

Este perfil é composto pelos elementos dos grupos General, Technical, Educational e Rights. Estes elementos são representados na Figura 2.7.

Figura 2.7 – Quadro dos elementos que compõe o perfil OBAA-LITE

<i>Elementos</i>			
1 General	4 Technical	5 Educational	6 Rights
1.2 Title	4.3 Location	5.2 LearningResourceType	6.2 CopyrightAndOtherRestrictions
1.3 Language		5.5 IntendedEndUserRole	6.3 Description
1.4 Description		5.6 Context	
1.5 Keyword		5.9 TypicalLearningTime	
		5.10 Description	

Fonte: Elaborado pelo autor.

2.6.3 OBAA-Videoaula

O OBAA-VideoAula possui os metadados mínimos para a representação de um OA pelo Padrão Videoaula@RNP. A Figura 2.5 descreve todos os metadados que compõe o Padrão. Os metadados em negrito são os que compõem o perfil OBAA-VideoAula.

2.6.4 OBAA-DTV

O perfil OBAA-DTV é proposto a partir do Padrão OBAA para conteúdos educacionais de forma compatível com a TV Digital (VICARI et al., 2009). No caso da TV Digital, como se reproduziu diretamente no Ginga-NCL Virtual Set-Top Box, o perfil é composto com o valor “ginga” para o conteúdo do metadado 4.9.5.1.2 SpecificName e o valor “Português do Brasil” para 1.3 Language.

2.7 Trabalhos Relacionados

Dentre os editores de conteúdo relacionados com este trabalho, pode-se considerar relevante a ponto de comparação ferramentas que:

- Utilizem ao menos um dos padrões de metadados anteriormente citados;
- Utilizem ontologias ou RDF para descrição de um OA;
- Representem algum padrão de metadados como ontologia.

Além de ser uma ferramenta de autoria, o eXe Learning⁵ também realiza a descrição dos metadados de um OA pelo padrão de metadados Dublin Core. Além disso, a exportação para IMS Content Packaging e SCORM pode ser feita. Esta ferramenta não é compatível com

a Web Semântica. Em cada campo de metadado a ser preenchido existe uma opção de ajuda, porém o conteúdo preenchido não é verificado posteriormente. A abordagem de perfis de aplicação não é utilizada.

Muitos dos editores de conteúdos de metadados são baseados a partir do padrão de metadados IEEE-LOM. Como por exemplo, uma ferramenta de autoria integrada em um Sistema Tutor Inteligente (GAROFALAKIS et al., 2007). Esta ferramenta não possui compatibilidade com a Web Semântica, vindo a se utilizar de uma base de dados para gerenciar o relacionamento dos metadados. Um arquivo XML pode ser visualizado e exportado, mas o seu conteúdo não é verificado.

Existem ferramentas de metadados muito funcionais, possuindo habilidades superiores para o gerenciamento de metadados IEEE-LOM. Por exemplo, o LOM Editor⁶ e LOMPAD⁷. Contudo, além da exportação dos metadados, nenhum outro fator comparável ao do presente trabalho é apresentado.

O framework SCAM⁸ tem como objetivo principal tanto carregar quanto armazenar os metadados de maneira que facilite a utilização por softwares que venham a utiliza-lo como base. O framework é independente do uso de perfis de aplicação. Mapeamentos RDF para Dublin Core e IEEE-LOM são utilizados. A verificação de conteúdos aparentemente não é realizada.

Outra ferramenta compatível com a Web Semântica é LOIT (GHEBGHOUBA et al., 2009), uma ferramenta de indexação baseada em uma ontologia LOM. É possível a exportação dos metadados e o acesso a informação de ajuda. LOIT não se utiliza de perfis de aplicação.

A Figura 2.8 apresenta um resumo comparativo das ferramentas mencionadas. A maioria dos itens de comparação é sinalizada com: (S) sim, utilizado; (N) não, não utilizado; (?) não encontrado, não relatado. Os itens de comparação empregados foram:

- WS: compatibilidade com a Web Semântica, utilização de ontologias ou RDF;
- Exp.: realiza exportação do OA em arquivo;
- Padrão: padrões de metadados utilizados pelo editor;
- Verif.: realiza verificação se os conteúdos preenchidos estão de acordo com o

⁵<http://exelearning.org/>

⁶<http://dbis.rwth-aachen.de/cms/projects/LOMEditor>

⁷<http://sourceforge.net/projects/lompad/>

⁸<http://scam.sourceforge.net/html/index.html>

padrão, tal como espaço de valores, cardinalidades, etc.;

- Perfil: utiliza perfis de aplicação para carregamento de metadados;
- Ajuda: possui documentação/ajuda incorporado ao editor.

Figura 2.8 - Quadro comparativo entre editores de conteúdos de metadados

<i>Nome</i>	<i>WS</i>	<i>Exp.</i>	<i>Padrão</i>	<i>Verif.</i>	<i>Perfil</i>	<i>Ajuda</i>
eXe Learning	N	S	DC,IMS, CP, SCORM	N	N	S
LOM Editor	N	S	LOM	N	N	N
LOMPAD	N	S	CanCore, LOM, Normetic, SCORM	N	N	N
?	N	S	LOM	N	?	S
SCAM	S	N	DC, IMS CP, LOM	N	N	N
LOIT	S	S	LOM	?	N	S

Fonte: Elaborado pelo autor.

2.8 Ferramental

Dentre as tecnologias utilizadas, as que tiveram maior relevância para a concretização do presente trabalho foram: o framework Protégé e a biblioteca OWLAPI.

2.8.1 Protégé

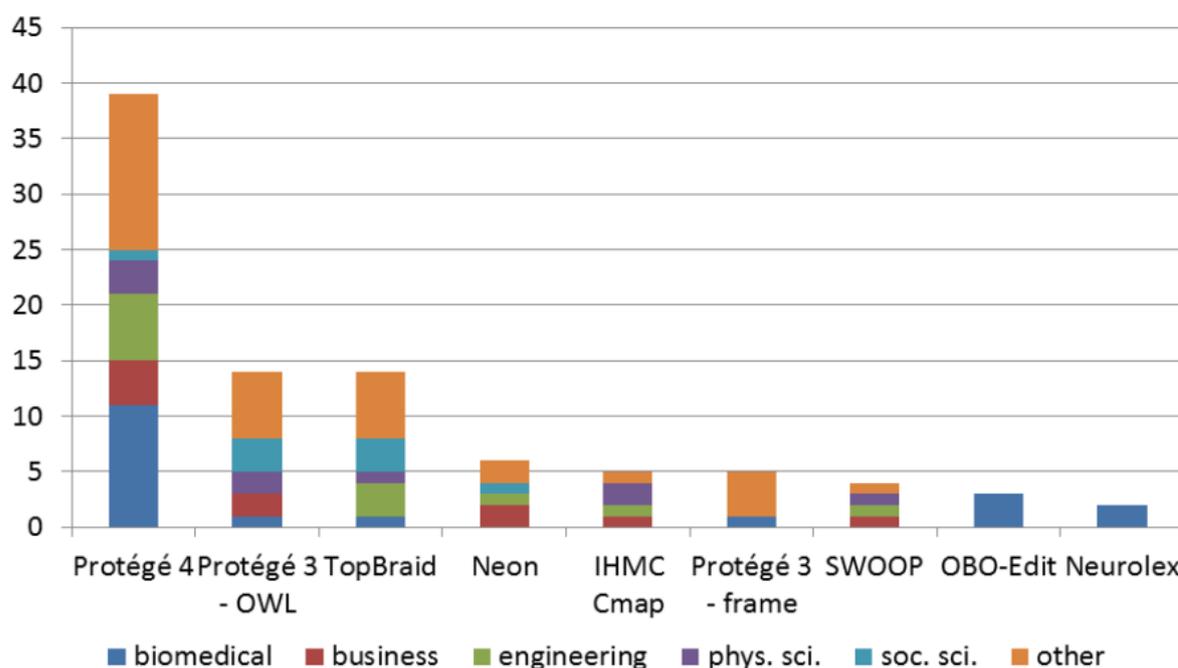
O editor de ontologias Protégé⁹ foi utilizado para a elaboração da primeira etapa deste trabalho, compreendida nos Capítulos 3, 4, 5 e 6. O Protégé é um framework baseado em conhecimento, editor de ontologias, livre e de código aberto. Dentre os motores de inferência disponíveis no framework, foi utilizado o Hermit, em versão 1.3.6.

Uma pesquisa com usuários de ontologias foi realizada pelo Knowledge Media Institute (KMI) em 2013. Ao todo, foram 118 respondentes. Um número de entrevistados respondeu apenas um subconjunto das perguntas, a maioria das perguntas resultou em várias dezenas de respostas. Nenhuma tentativa foi feita para conseguir uma amostra representativa de todos os usuários de ontologias, e isso se deve ter em mente na interpretação dos resultados (WARREN, 2013).

Os resultados obtidos relacionados ao uso de editores de ontologias são ilustrados na Figura 2.9. É importante salientar que os campos de Ciência da Computação e Tecnologia da Informação estão inseridos na área de aplicação “other”.

⁹<http://protege.stanford.edu/>

Figura 2.9 – Número de usuários por editores de ontologias



Fonte: Warren (2013, p. 10).

2.8.2 OWLAPI

A OWLAPI¹⁰ é uma API de alto nível para trabalhar com ontologias OWL. A API é fortemente alinhada com a especificação estrutural da OWL 2. Ela oferece suporte à análise e renderização em sintaxes definidas na especificação W3C (Sintaxe Funcional, RDF/XML, OWL/XML e a Sintaxe Manchester OWL); manipulação de estruturas ontológicas; e o uso de motores de raciocínio. A referência de implementação da OWLAPI é escrita em Java, inclui validadores para os vários perfis da OWL 2 - OWL 2 QL, OWL 2 EL e OWL 2 RL. A OWLAPI tem seu uso bem difundido em uma variedade de ferramentas e aplicações (HORRIDGE; BECHHOFER, 2011).

Como a implementação do editor de conteúdos de metadados fora desenvolvida na linguagem Java, a OWLAPI se mostra extremamente compatível com a ideia sugerida de trabalho. O IDE utilizado para codificação foi o NetBeans¹¹, gratuito e de código aberto.

¹⁰<http://owlapi.sourceforge.net/>

¹¹<https://netbeans.org/>

3 UMA ONTOLOGIA PARA O PADRÃO OBAA DE METADADOS

A ontologia OBAA, a seguir descrita, deriva-se de duas outras ontologias, que foram anteriormente criadas: LOM¹ e IMS AccessForAll². Estas duas ontologias são reusadas através de importação. Como o desenvolvimento entre elas se assemelha, somente a ontologia OBAA será explanada.

A seguir serão abordadas as etapas realizadas para obtenção da ontologia. Cabe salientar que mesmo que as etapas estejam descritas por uma ordem cronológica, muitas vezes as etapas tiveram que ser retomadas para ajustes na ontologia.

3.1 Hierarquia de Classes

A partir de cada item de metadado do Padrão OBAA foi criada uma classe correspondente na ontologia, seguindo-se assim a hierarquia pré-determinada, resultando em classes e subclasses. Tal metodologia é semelhante a utilizada por Qin e Finneran, onde cada componente de um objeto de aprendizagem é normalizado em um grupo de classes em uma classe mais genérica (QIN; FINNERAN, 2002).

A nomenclatura de cada classe respeitou os nomes dos metadados propostos nos respectivos relatórios técnicos: LOM (COMMITTEE, 2002), IMS AccessForAll (CONSORTIUM, 2004) e OBAA (VICARI et al., 2009). Contudo, a ferramenta Protégé sugere que não sejam utilizados mesmos nomes para entidades diferentes. Isto ocorre devido a cada entidade possuir uma URI única referenciável em uma ontologia. Esta característica é denominada Unique Name Assumption (UNA). Por exemplo, nos metadados do LOM encontramos Description como sendo o nome dos metadados de identificadores 1.4, 5.10, 6.3, 8.3 e 9.3.

Portanto, para obter classes com diferentes nomes, foi concatenado o nome do metadado pai e o caractere ponto ao nome do metadado até que se resulte em um nome único. Por exemplo, General e Rights contem o metadado Description. Então, as classes resultantes serão General.Description e Rights.Description, respectivamente.

¹<http://gia.inf.ufrgs.br/ontologies/LOM.owl>

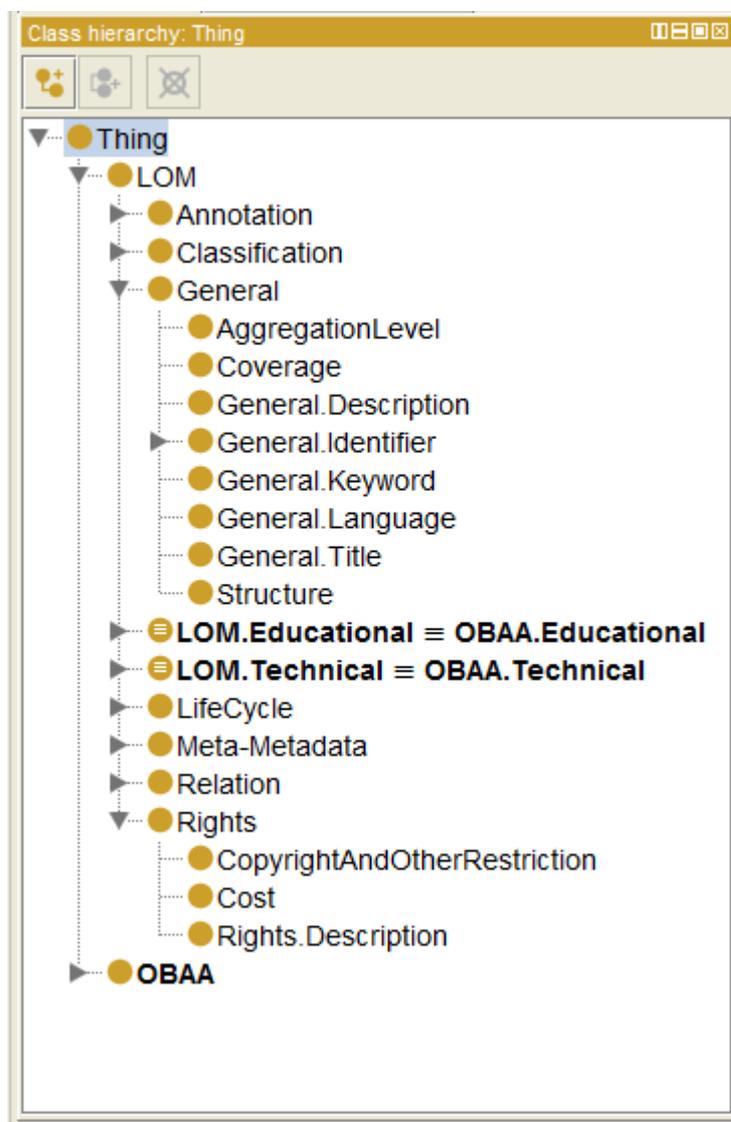
²http://gia.inf.ufrgs.br/ontologies/IMS_AccessForAll.owl

O grupo de metadados Rights contém os metadados Cost, Copyright and Other Restrictions e Description. A hierarquia de classe resultante deste grupo é ilustrada na Figura 3.1.

Por fim, as classes LOM.Educational e LOM.Technical foram determinadas como equivalentes a OBAA.Educational e OBAA.Technical, respectivamente. A Figura 3.1 também ilustra estas equivalências. Além disso, a classe Accessibility do padrão IMS teve sua superclasse definida como sendo a classe OBAA.

A hierarquia de classes é utilizada para determinar domínios de propriedades. Estes domínios são importantes para a identificação dos tipos do objeto de aprendizagem para sua posterior classificação. Além disso, é possível definir as cardinalidades de propriedades em nível de classe, como será abordado a seguir.

Figura 3.1 – Hierarquia geral LOM e detalhamento do grupo de metadados Rights



Fonte: Elaborado pelo autor.

3.2 Propriedades

Cada metadado também resultou em uma propriedade. Estas propriedades foram definidas de acordo com a característica do metadado. Se o metadado é um nodo folha (não possui filhos), uma propriedade de dado será criada para ele. Caso o metadado seja do tipo container (possui filhos) e com uma cardinalidade maior que um, será criada uma respectiva propriedade de objeto na ontologia. Assim, se a cardinalidade de um metadado container for um, não será criada uma propriedade de objeto para ele.

A criação de uma propriedade por classe e a não utilização de propriedades compartilhadas deveu-se ao fato de o processo de inferência ser prejudicado. Com o compartilhamento de propriedades, o domínio da mesma acaba por conter uma união de classes. Assim, ao verificar os tipos do indivíduo, ao término do processo de inferência, constata-se que é definida sempre uma classe que é uma superclasse comum à definida pela união do domínio. Outro agravante, é o fato de que uma propriedade compartilhada acaba tendo diferentes cardinalidades, o que acaba por imprecisar qual a cardinalidade em questão.

3.2.1 Propriedades de dados

Como nomenclatura, as propriedades de dados foram nomeadas com o mesmo nome das classes, diferenciando-se apenas pela primeira letra estar em forma minúscula. Cada propriedade teve seu domínio definido com sua classe correspondente bem como seu escopo (tipo de dado aceitável) delineado.

As propriedades de dados têm como principal objetivo armazenar as informações contidas nos metadados de nodos folhas. A partir do escopo, é possível determinar se a propriedade foi ou não corretamente preenchida. Esta verificação é dada através de um motor de inferência que também classifica os indivíduos que possuem determinadas propriedades a partir dos domínios das mesmas.

Caso um metadado tenha um espaço de valores definido de forma condicional, como o metadado Name (número 4.4.1.2), esta restrição é feita em nível de superclasse e o escopo da propriedade é definido contendo todos os valores possíveis. Tal processo é ilustrado na Figura 3.2, onde se tem o espaço de valores: “amaya, any, ms-internet explorer, netscape communicator e opera” para Name quando o valor de Type (número 4.4.1.1) for “browser” e “macos, ms-windows, multi-os, none, pc-dos e unix” como espaço de valores para Name se Type contiver o conteúdo “operating-system”.

Figura 3.2 – Lógica para espaço de valores do metadado Name

```

(((orComposite.Name value "amaya")
 or (orComposite.Name value "any")
 or (orComposite.Name value "ms-internet explorer")
 or (orComposite.Name value "netscape communicator")
 or (orComposite.Name value "opera"))
 and (orComposite.Type value "browser"))
 or (((orComposite.Name value "macos")
 or (orComposite.Name value "ms-windows")
 or (orComposite.Name value "multi-os")
 or (orComposite.Name value "none")
 or (orComposite.Name value "pc-dos")
 or (orComposite.Name value "unix"))
 and (orComposite.Type value "operating system"))

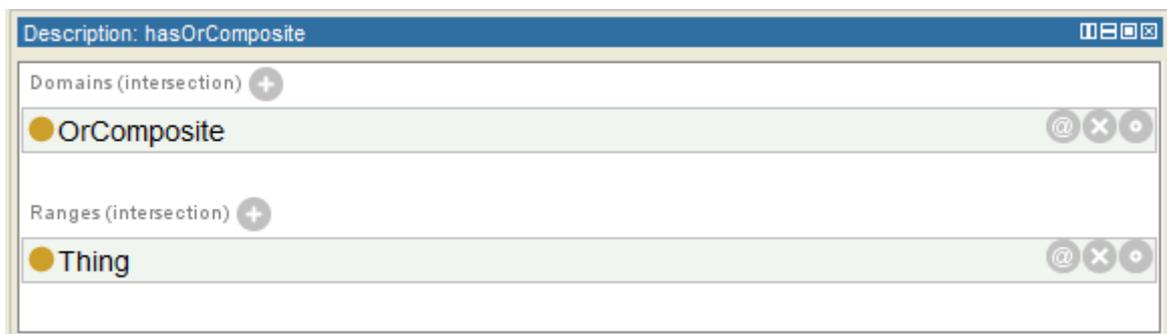
```

Fonte: Elaborado pelo autor.

3.2.2 Propriedades de Objetos

Por sua vez, as propriedades de objetos foram nomeadas com o mesmo nome da classe, porém precedidas pelo prefixo “has”. Os domínios foram definidos de acordo com a classe a qual a propriedade de objeto pertence. Já os escopos foram todos determinados com a classe Thing, a superclasse mais alta na hierarquia de uma ontologia. Este procedimento é exemplificado na Figura 3.3.

Figura 3.3 – Domínio e escopo da propriedade de objeto hasOrComposite



Fonte: Elaborado pelo autor.

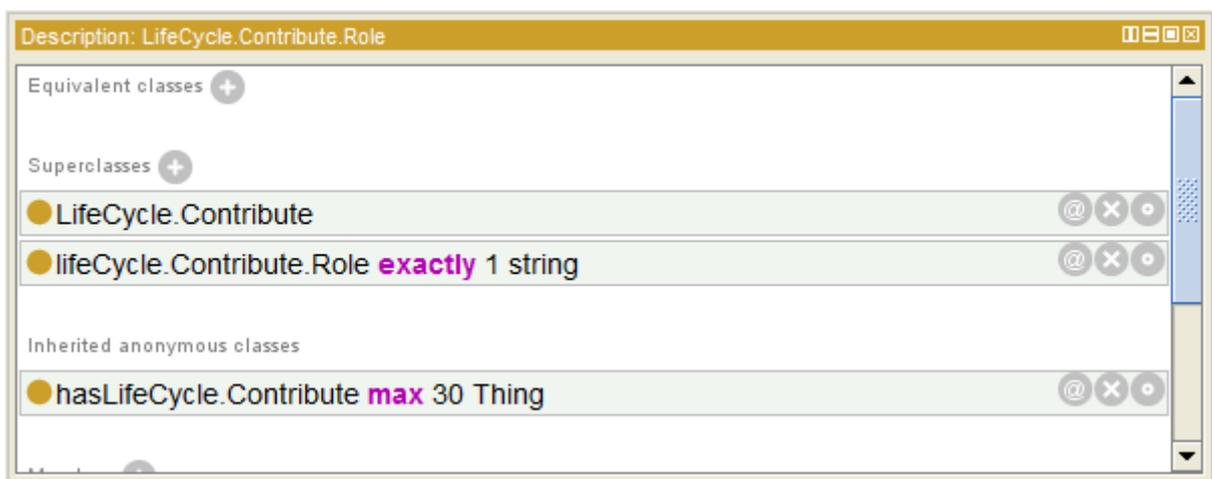
Este escopo deve-se ao fato de que como as propriedades de objeto relacionam indivíduos na ontologia, estes por sua vez podem ser classificados em qualquer classe. Assim, as propriedades de objetos relacionam indivíduos, que por sua vez representam os metadados container de um objeto de aprendizagem ou diferentes objetos de aprendizagem, como a seguir será abordado.

3.3 Cardinalidades

As cardinalidades foram definidas a partir dos relatórios técnicos. Cada cardinalidade é associada com seu metadado correspondente e todas as cardinalidades das propriedades de dado e de objeto foram definidas em nível de superclasse, objetivando um padrão.

Também é possível restringir cardinalidade de uma propriedade como funcional, quando esta apresenta cardinalidade um. Neste caso, optou-se por realizar as duas restrições, visto que a forma funcional é verificada primeira pelo motor de inferência. A Figura 3.4 exemplifica a restrição em nível de superclasse para a propriedade de dado `lifeCycle.Contribute.Role`, referente ao metadado 2.3.1 Role.

Figura 3.4 – Cardinalidade da propriedade de dados `lifeCycle.Contribute.Role`



Fonte: Elaborado pelo autor.

O principal objetivo das cardinalidades é limitar o número de propriedades que um indivíduo possa ter. Assim, por exemplo, se um objeto de aprendizagem deve ter somente um título, isto será restringido por meio da cardinalidade.

3.4 Indivíduos

Com o objetivo de obter uma ontologia de granularidade mínima possível, uma quantidade mínima de indivíduos deverá ser criada. Por este motivo, somente os metadados container com cardinalidade maior que um deverão ser representados por um indivíduo. Por fim, todos os indivíduos são sinalizados como diferentes.

Com um indivíduo, é possível manter a relação entre as propriedades de dados do metadado container e sua cardinalidade máxima. Entretanto, é necessário criar um indivíduo para a utilização desta propriedade sempre que um metadado container for utilizado.

Um objeto de aprendizagem será representado por um ou mais indivíduos. Existirá um indivíduo principal, o qual representará o objeto de aprendizagem em si, e que estará ligado a representações dos metadados container antes descritos.

Assim, as propriedades de dados de um item de metadado container de cardinalidade um serão vinculadas a um indivíduo que represente o metadado pai deste container. Caso não haja um metadado pai com cardinalidade maior que um, estas propriedades serão relacionadas ao indivíduo principal.

Com isso, será possível consistir esta representação do OA, verificando se as propriedades estão corretamente preenchidas. Também é possível obter inferências através do processo de raciocínio.

3.5 Anotações e Documentação

A OWL 2 permite que se possa utilizar anotações em conjunto com ontologias. Assim, a partir dos relatórios técnicos do OBAA, LOM e IMS AccessForAll é possível criar anotações com os mesmos rótulos das tabelas dos relatórios e armazenar suas descrições. Além disso, é possível criar diferentes comentários, tais como versão, problemas, itens a fazer, etc.

Por motivo de padronização, os nomes das propriedades de anotações seguem a mesma nomenclatura dos rótulos do relatório técnico do padrão LOM. Assim, os padrões OBAA e IMS AccessForAll foram relacionados com a sua tradução ou contexto correspondente ao padrão LOM.

Além disso, um plugin para Protégé chamado OWL Doc³ gera uma página HTML contendo toda a documentação acima mencionada. Isto permite uma visão mais abrangente da ontologia mesmo não possuindo a ferramenta Protégé.

Toda a documentação foi importada para uma página Web: <http://gia.inf.ufrgs.br/documentacao>. Assim, é possível acessar toda a semântica da ontologia através de uma URI.

³<http://www.co-ode.org/downloads/owldoc/>

4 UMA ONTOLOGIA PARA O PADRÃO VIDEOAULA@RNP DE METADADOS

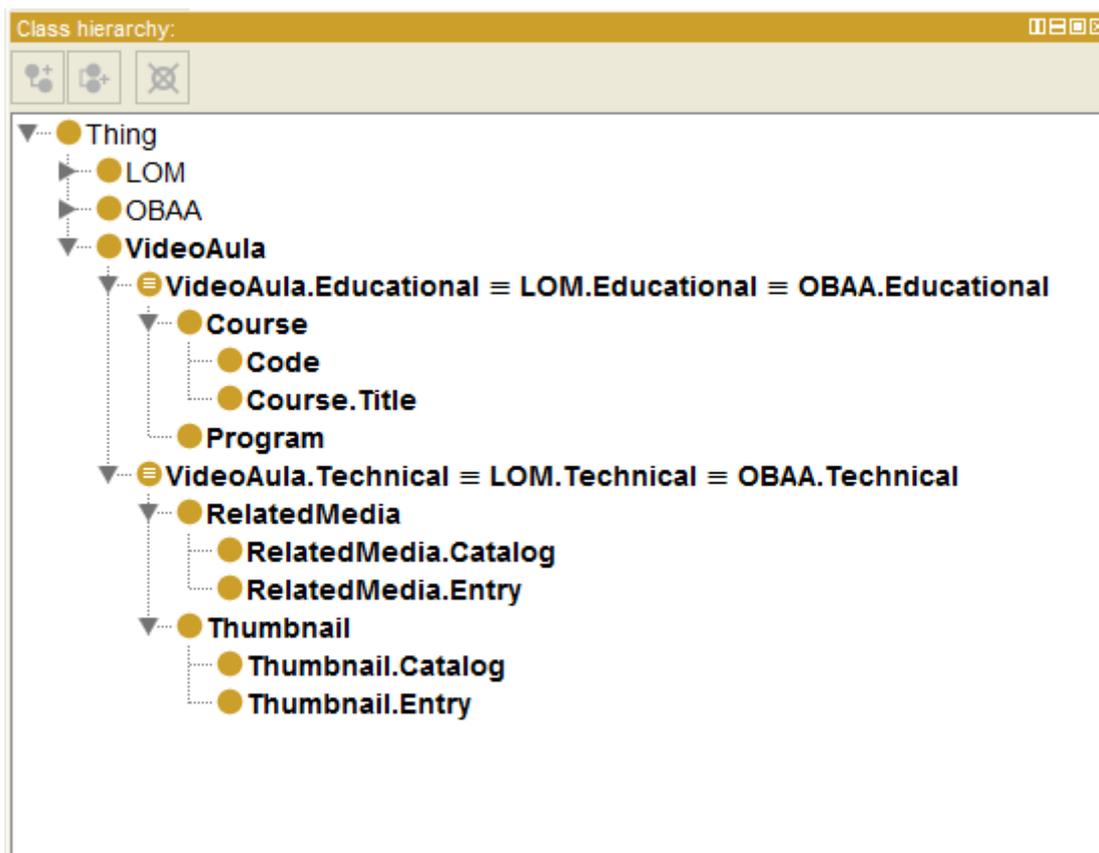
A ontologia a seguir é delineada a partir da definição do padrão de metadados descrito na Seção 2.5. A metodologia se assemelha a aplicada no capítulo anterior.

4.1 Hierarquia de Classes

A hierarquia de classes foi delineada a partir da hierarquia de metadados do Padrão Videoaula@RNP, presente na Figura 2.5. Cada classe foi criada com o mesmo nome do metadado, salvo quando ocorre a repetição de nome do metadado. Em caso de repetição, é realizado a mesma metodologia empregada na Seção 3.1.

Como o Padrão Videoaula@RNP realiza uma extensão sobre os grupos metadados Educational e Technical do Padrão OBAA, a ontologia OBAA foi importada. Além disso, as classes VideoAula.Educational e VideoAula.Technical foram sinalizadas como equivalentes a OBAA.Educational e OBAA.Technical, respectivamente. Como consequência, acabam por equivaler-se também a LOM.Educational e LOM.Technical, como ilustrado na Figura 4.1.

Figura 4.1 – Hierarquia de classes VideoAula



Fonte: Elaborado pelo autor.

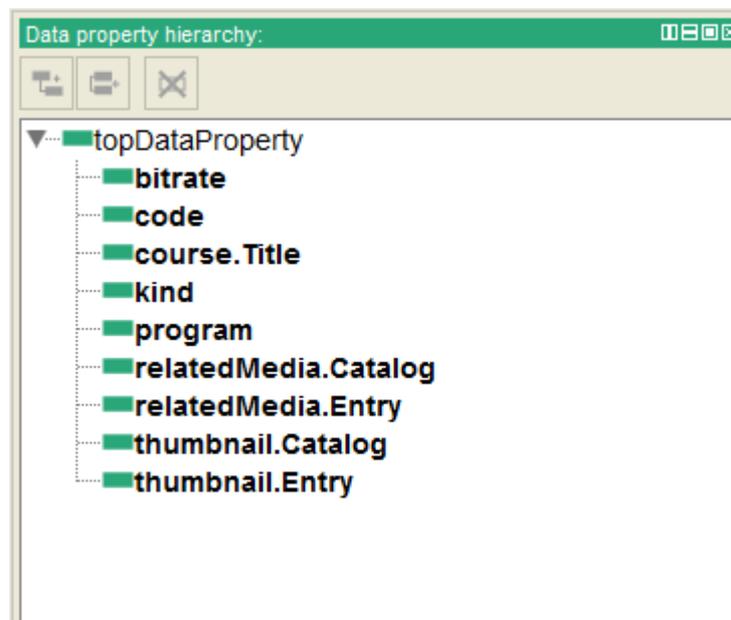
4.2 Propriedades

A definição das propriedades ocorreu da mesma forma como realizada para com as propriedades da ontologia OBAA. A Seção 3.2 explana tal procedimento detalhadamente.

4.2.1 Propriedades de Dados

Os itens de metadados que não possuem nenhum metadado filho são representados na ontologia também como propriedades de dados. A Figura 4.2 ilustra as novas propriedades criadas, as propriedades da ontologia OBAA foram suprimidas para fins de ilustração.

Figura 4.2 – Novas propriedades de dados da ontologia VideoAula



Fonte: Elaborado pelo autor.

4.2.2 Propriedades de Objetos

De outra maneira, os itens de metadados que possuem metadados filhos e que tenham cardinalidade maior que um também são representados na forma de uma propriedade de objeto. As novas propriedades de objeto são representadas na Figura 4.3, as demais propriedades de objeto foram suprimidas para fins de ilustração.

Figura 4.3 – Novas propriedades de objetos da ontologia VideoAula



Fonte: Elaborado pelo autor.

4.3 Cardinalidades

As cardinalidades de cada metadado são ilustradas na Figura 2.6. Da mesma forma que apresentado na Seção 3.3, as cardinalidades foram delineadas em nível de superclasse e quando o valor for um, a propriedade também é sinalizada como funcional.

4.4 Indivíduos

Da mesma forma que na ontologia OBAA, um OA será representado por um indivíduo principal relacionado por propriedades de objetos com outros indivíduos que representarão metadados container de cardinalidade maior que um. Por sua vez, estes conterão propriedades de dados e, caso necessário, propriedades de objetos para relacionamento com outros indivíduos. Ao final do processo, todos os indivíduos são sinalizados como diferentes.

A relação acima citada é importante por manter a relação hierárquica de informações tal como em um arquivo de metadados. No decorrer do trabalho, esta representação é melhor ilustrada no Capítulo 6.

4.5 Anotações e Documentação

Semelhante a ontologia OBAA, a documentação da ontologia VideoAula também foi criada a partir de anotações realizadas na ontologia. A documentação está disponível em <http://gia.inf.ufrgs.br/ontologies/videoaula/documentacao/>.

5 ONTOLOGIAS PARA PERFIS DE APLICAÇÃO

Cada perfil de aplicação resultou na elaboração de sua respectiva ontologia. Esta ontologia constitui, principalmente, de uma classe de equivalência. Isto se deve porque a OWL realiza a descrição de “mundo aberto”, pois é baseada na lógica descritiva.

A hipótese de mundo aberto¹ diz que nós não podemos concluir alguma afirmação χ seja falsa simplesmente porque não podemos mostrar que χ seja verdadeiro. Afinal, nossos axiomas podem simplesmente descomprometer o estado de χ . Em outras palavras, nós não podemos deduzir falsidade de uma ausência da verdade (ANTONIOU; VAN HARMELEN, 2004).

A partir disso, uma classe de equivalência é necessária para especificar as condições suficientes e necessárias do perfil de aplicação. Através destas condições, é possível consistir se a descrição do objeto de aprendizagem foi realizada de forma coerente.

Elas também possibilitam a classificação dos objetos de aprendizagem, verificando quais objetos pertencem a determinados perfis.

Para a realização da classificação de objetos de aprendizagem foi necessária a criação de cadeias de propriedades de objetos de acordo com o perfil. Estas cadeias são responsáveis por propagar todos os tipos dos indivíduos que compõem o OA para o indivíduo principal (o qual representa o OA como um todo).

Com isso, ao final de um processo de inferência obteremos um resultado mais refinado, pois será retornado um só indivíduo e não todos os indivíduos relacionados com o indivíduo principal. Estas cadeias também são importantes como um acesso às propriedades de dados dos indivíduos que compõem o OA.

Como exemplo, tomando o nodo PlatformSpecificFeatures (ilustrado na Figura 2.4, presente na Seção 2.4), a cadeia seria composta pelos nodos: PlatformSpecificFeatures, SpecificRequirement e SpecificOrComposite.

Assim, uma cadeia será formada por cada caminho e nível do nodo. Como resultado do exemplo, uma propriedade de objeto com o nome hasPlatformSpecificFeaturesChain e suas cadeias é ilustrada pela Figura 5.1.

¹ Em inglês: open-world assumption.

Figura 5.1 – Exemplo de cadeia de propriedade de objetos

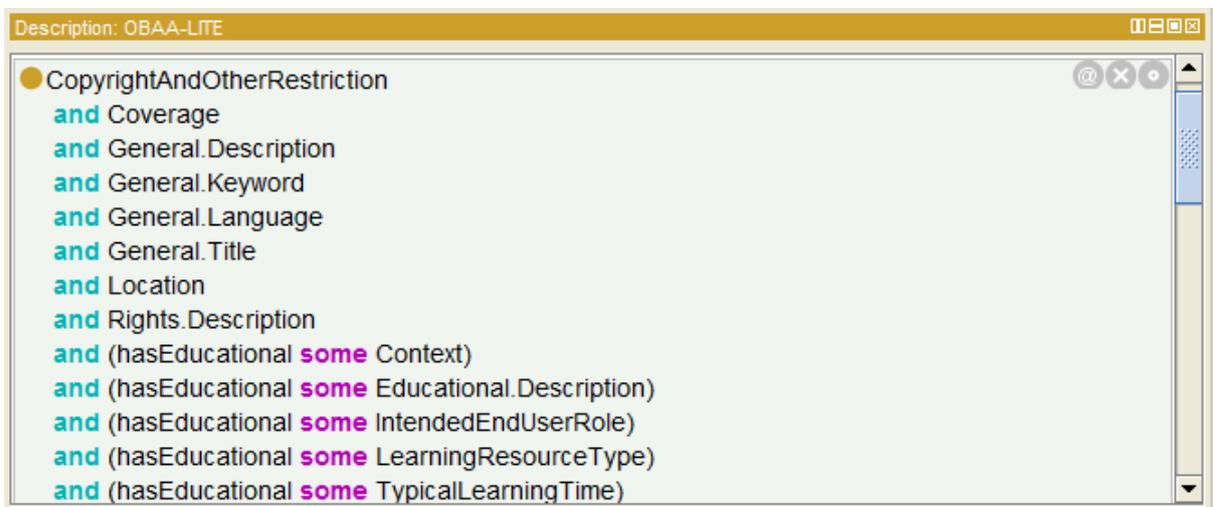


Fonte: Elaborado pelo autor.

5.1 Perfil OBAA-LITE

A partir dos metadados que compõem o perfil, presentes na Seção 2.6.2, foi definida uma classe de equivalência, ilustrada pela Figura 5.2. A ontologia completa do perfil pode ser acessada em: <http://gia.inf.ufrgs.br/ontologies/OBAA-LITE.owl>.

Figura 5.2 – Classe de equivalência OBAA-LITE



Fonte: Elaborado pelo autor.

5.2 Perfil OBAA-VideoAula

A partir dos metadados mínimos para a descrição de uma vídeo aula, sinalizados em negrito na Figura 2.5, foi delineada uma classe de equivalência. A Figura 5.3 representa esta classe. Em <http://gia.inf.ufrgs.br/ontologies/OBAA-VideoAula.owl> pode-se acessar a ontologia completa.

Figura 5.3 – Classe de equivalência OBAA-VideoAula



Fonte: Elaborado pelo autor.

5.3 Perfil OBAA-DTV

O perfil OBAA-DTV difere-se por ser determinado a partir do conteúdo de metadados, presentes na Seção 2.6.4. A classe de equivalência baseada nestes conteúdos é apresentada na Figura 5.4. A ontologia que representa este perfil é encontrada em: <http://gia.inf.ufrgs.br/ontologies/OBAA-DTV.owl>.

Figura 5.4 – Classe de equivalência OBAA-DTV



Fonte: Elaborado pelo autor.

6 RESULTADOS PARCIAIS

Este capítulo visa apresentar os resultados obtidos através do desenvolvimento do trabalho realizado nos Capítulos 3, 4 e 5 através do framework Protégé. Note que o indivíduo principal, anteriormente citado, é o único que possui somente setas partindo para outros indivíduos.

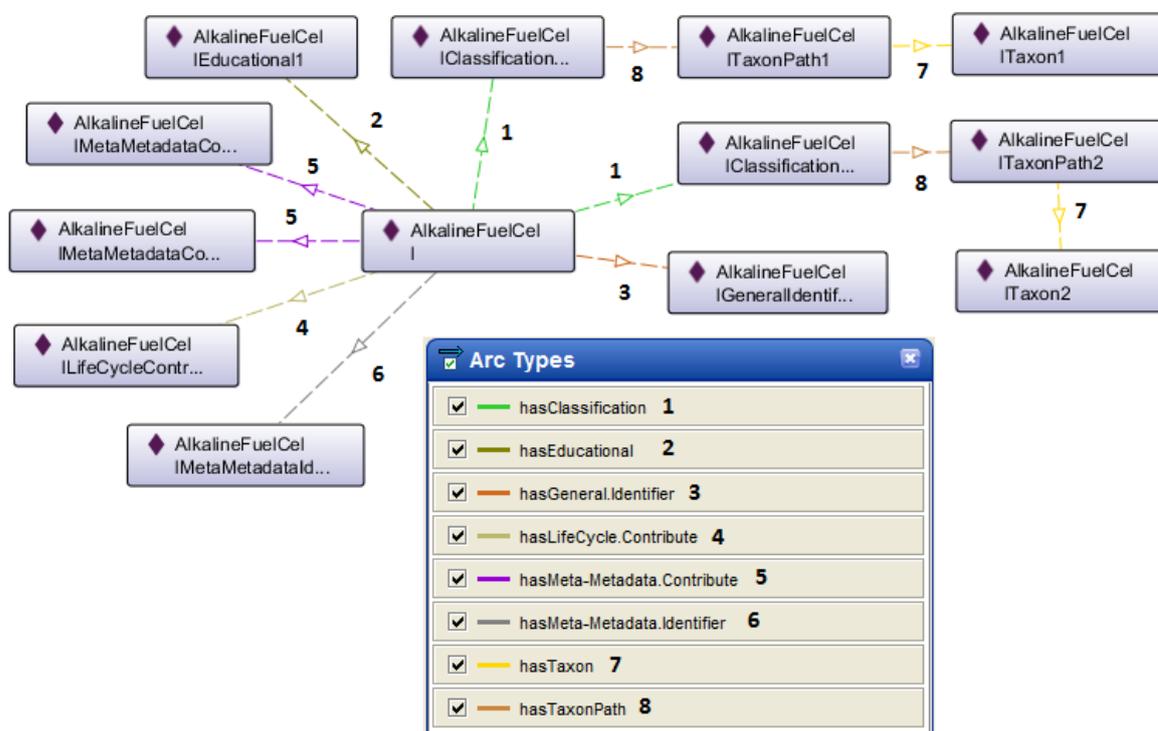
6.1 Representação de Objetos de Aprendizagem como Ontologias

A seguir serão apresentados alguns metadados de objetos de aprendizagem representados como ontologias. Uma visão geral destas representações será ilustrada, para o detalhamento completo acesse o link correspondente.

6.1.1 Célula Combustível Alcalina

O objeto de aprendizagem Célula Combustível Alcalina (SILVA, 2011) possui seus indivíduos e propriedades de objetos dos metadados representados pela Figura 6.1. As relações apresentadas pelas propriedades de objetos têm como objetivo manter a hierarquia de informações contidas em seus arquivos XML originais.

Figura 6.1 – Representação geral do OA Célula Combustível Alcalina



Fonte: Elaborado pelo autor.

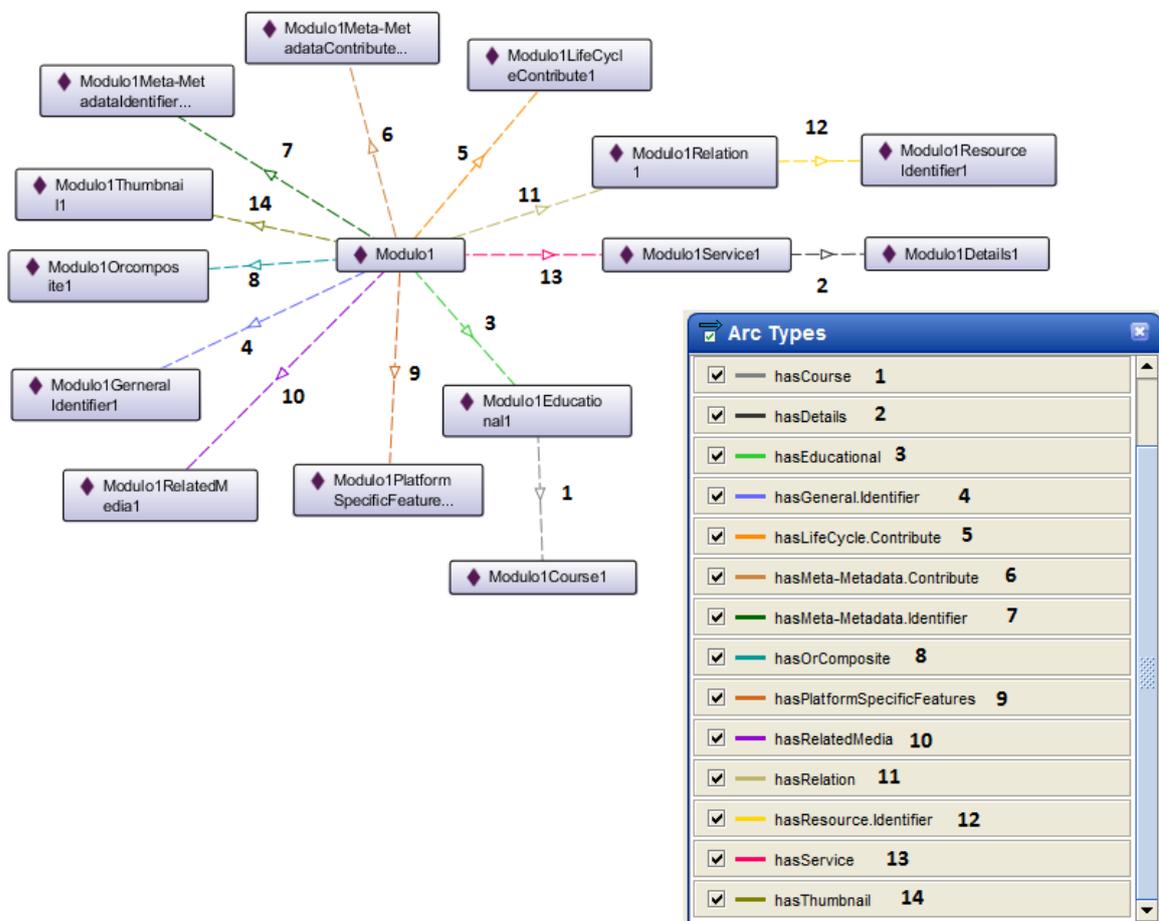
Cada relação é apresentada com uma numeração correspondente a sua propriedade de objeto. Estas relações vinculam os indivíduos que representam os metadados container e possuem propriedades de dados que armazenam o conteúdo dos metadados do OA.

A ontologia completa pode ser acessada em: <http://gia.inf.ufrgs.br/ontologies/OAS/AlkalineFuelCell/AlkalineFuelCell.owl>.

6.1.2 Apresentação do Serviço Videoaula@RNP: Módulo 1

Os relacionamentos entre os indivíduos que representam os metadados do objeto de aprendizagem Apresentação do Serviço Videoaula@RNP: Módulo 1² são ilustradas pela Figura 6.2. Esta figura descreve uma relação hierárquica relativamente mais completa do que a apresentada na seção anterior.

Figura 6.2 – Representação geral do OA Módulo 1



Fonte: Elaborado pelo autor.

²http://videoaula.rnp.br/rioflashclient.php?xmlfile=/rnp/treinamentos/videoaulas/modulo_1/modulo_1.xml

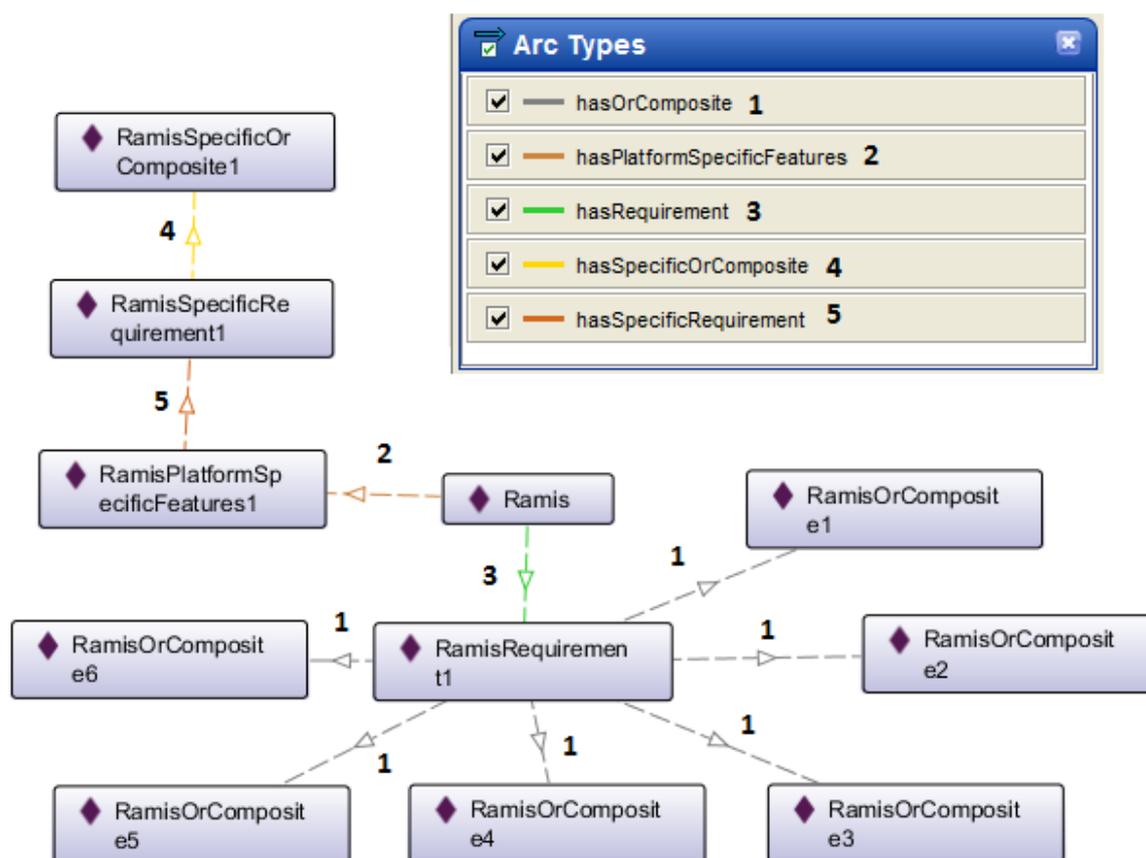
A ontologia correspondente a descrição dos metadados do OA³ pode ser visualizada em: <http://gia.inf.ufrgs.br/ontologies/OAS/Modulo1/Modulo1.owl>.

6.1.3 Ramis

A Figura 6.3 apresenta os indivíduos e seus relacionamentos na representação dos metadados do OA Ramis⁴. A Figura 6.4 exibe as propriedades de dados destes indivíduos, cada propriedade é responsável por armazenar o conteúdo dos metadados do OA.

Em <http://gia.inf.ufrgs.br/ontologies/OAS/Ramis/Ramis.owl> pode ser obtida a representação completa deste OA.

Figura 6.3 – Representação do OA Ramis



Fonte: Elaborado pelo autor.

⁴<http://www.portalobaa.org/padrao-obaa/exemplos-de-oa-convertidos-para-o-obaa/objetos-interoperaveis/objeto-de-aprendizagem-ramis>

³http://va05-cps.rnp.br/riotransfer/rnp/treinamentos/videoaulas/modulo_1/modulo_1.xml

Figura 6.4 – Propriedades de dados dos indivíduos do OA Ramis

Ramis Data property assertions + duration "PT00H17M00S"^^string format "CSS"^^string general.Keyword "IMC"^^string general.Keyword "Atividade Fisica"^^string format "Imagens JPEG"^^string location "http://gia.inf.ufrgs.br/OBAI/storage/o: format "Texto XHTML"^^string format "Videos H.264"^^string supportedPlatforms "DTV"^^string general.Language "Português do Brasil"^^string supportedPlatforms "Mobile"^^string general.Description "Apresenta idéias para melhorar estilo de vida"^^string size "14MB"^^string supportedPlatforms "Web"^^string general.Keyword "hábitos alimentares"^^string general.Keyword "Reeducação Alimentar"^^string	RamisOrComposite1 Data property assertions + orComposite.Type "browser"^^string minimumVersion "7.0"^^string orComposite.Name "Internet Explorer"^^string RamisOrComposite2 Data property assertions + orComposite.Type "browser"^^string orComposite.Name "Mozilla Firefox"^^string minimumVersion "3.0"^^string RamisOrComposite3 Data property assertions + orComposite.Name "Windows Mobile"^^string minimumVersion "6.1"^^string orComposite.Type "operating system"^^string RamisOrComposite4 Data property assertions + orComposite.Name "Internet Explorer Mobile"^^string orComposite.Type "browser"^^string RamisOrComposite5 Data property assertions + orComposite.Name "Symbian"^^string orComposite.Type "operating system"^^string	RamisOrComposite6 Data property assertions + orComposite.Type "browser"^^string orComposite.Name "S60 Browser"^^string RamisPlatformSpecificFeatures1 Data property assertions + specificSize "1,8 MB"^^string specificLocation "http://gia.inf.ufrgs.br/OBAI/storage/oar: specificInstallationRemarks "Descompactar o ZIP, transferir arquivos do objeto de aprendizagem para um pen drive e abrir o arquivo \curso.ncl" pelo Ginga NCL Live CD 1.0, conforme descrito na comunidade Ginga no Portal de Software Público Brasileiro (www.softwarepublico.gov.br)"^^string platformType "DTV"^^string specificFormat "ZIP (Conteúdo NCL, XHTML, CSS, Imagens Jpeg)"^^string RamisSpecificOrComposite1 Data property assertions + specificType "middleware"^^string specificMinimumVersion "1.0"^^string specificName "ginga"^^string
--	--	--

Fonte: Elaborado pelo autor.

6.1.4 TV Educativa

Os metadados do OA TV Educativa⁵ têm seus indivíduos e as propriedades de objetos que os relacionam caracterizados na Figura 6.5. As propriedades de dados de cada indivíduo e seus conteúdos são relatadas na Figura 6.6.

A ontologia completa pode ser encontrada em: <http://gia.inf.ufrgs.br/ontologies/OAS/TVEducativa/TVEducativa.owl>.

6.2 Verificação de Consistência

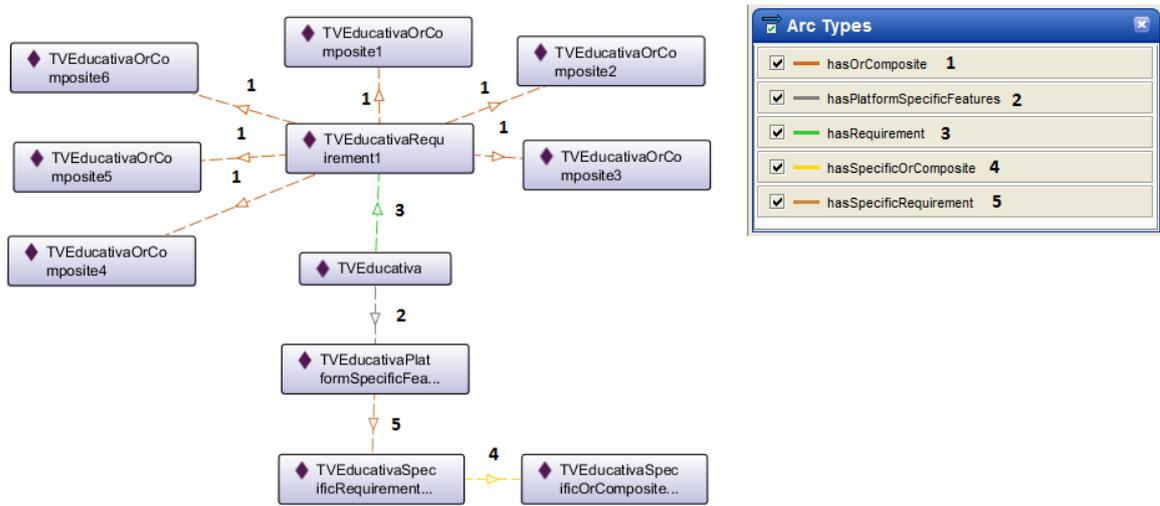
Através de motores de inferência é possível realizar a consistência da ontologia. A seguir serão dados exemplos de casos de inconsistência, relativos a: tipo de valor incorreto, valor fora do espaço de valores proposto, valor condicional não compatível e cardinalidade acima do permitido.

O framework Protégé ainda permite obter qual o indivíduo e o valor inconsistente, por

⁵<http://www.portalobaa.org/padrao-obaa/exemplos-de-oa-convertidos-para-o-obaa/objetos-interoperaveis/objeto-de-aprendizagem-tv-educativa>

meio da opção de explicação⁶.

Figura 6.5 – Representação geral do OA TV Educativa



Fonte: Elaborado pelo autor.

Figura 6.6 – Propriedades de dados dos indivíduos do OA TV Educativa

TVEducativa Data property assertions + size "76 MB"^^string format "CSS"^^string format "Imagens JPEG"^^string general.Keyword "Implementação TVD"^^string general.Language "Português do Brasil"^^string duration "PT00H15M00S"^^string general.Keyword "TVD na Escola"^^string format "Videos H.264"^^string location "http://gia.inf.ufrgs.br/OBAI/storage/tv supportedPlatforms "Mobile"^^string supportedPlatforms "Web"^^string general.Keyword "Interação TVD"^^string supportedPlatforms "DTV"^^string format "Texto XHTML"^^string general.Description "Destina-se a discussão sobre a TVD nas escolas"^^string	TVEducativaOrComposite1 Data property assertions + orComposite.Name "Internet Explorer"^^string minimumVersion "7.0"^^string orComposite.Type "browser"^^string TVEducativaOrComposite2 Data property assertions + orComposite.Type "browser"^^string orComposite.Name "Mozilla Firefox"^^string minimumVersion "3.0"^^string TVEducativaOrComposite3 Data property assertions + minimumVersion "6.1"^^string orComposite.Name "Windows Mobile"^^string orComposite.Type "operating system"^^string TVEducativaOrComposite4 Data property assertions + orComposite.Type "browser"^^string orComposite.Name "Internet Explorer Mobile"^^string TVEducativaOrComposite5 Data property assertions + orComposite.Type "operating system"^^string orComposite.Name "Symbian"^^string	TVEducativaOrComposite6 Data property assertions + orComposite.Name "S60"^^string orComposite.Type "browser"^^string TVEducativaPlatformSpecificFeatures1 Data property assertions + specificSize "107 MB"^^string specificFormat "ZIP (Conteúdo NCL, XHTML, CSS, Imagens Jpeg, Video Mpeg-4/H.264)"^^string platformType "DTV"^^string specificLocation "http://gia.inf.ufrgs.br/OBAI/storage/tve specificInstallationRemarks "Descompactar o ZIP, transferir arquivos do objeto de aprendizagem para um pen drive e abrir o arquivo 'objeto_tv.ncl' pelo Ginga NCL Live CD 1.0, conforme descrito na comunidade Ginga no Portal de Software Público Brasileiro (www.softwarepublico.gov.br)"^^string TVEducativaSpecificOrComposite1 Data property assertions + specificName "ginga"^^string specificType "middleware"^^string specificMinimumVersion "1.0"^^string
---	--	---

Fonte: Elaborado pelo autor.

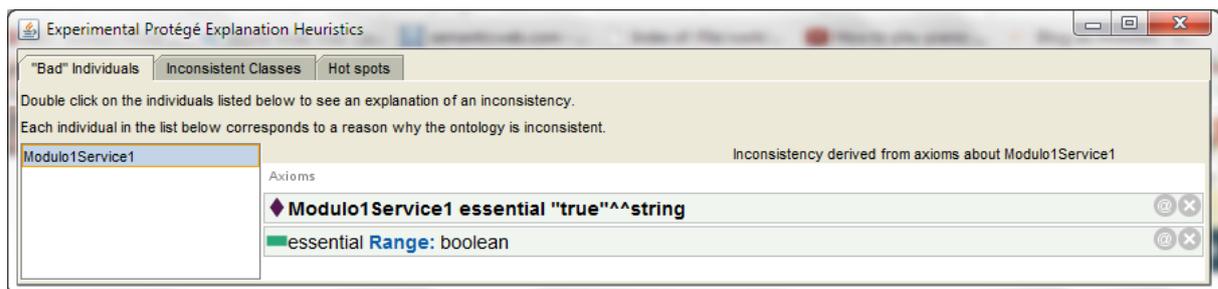
⁶Em inglês: Explain.

6.2.1 Tipo de Valor

No presente trabalho, quando se define uma propriedade de dado em uma ontologia, especificam-se o seu domínio (classe a qual pertence) e seu escopo (tipo de valor aceito). Dentre os tipos de valores podemos citar: string, boolean, integer, dateTime, entre outros.

Deste modo, quando um valor de propriedade for especificado com um tipo diferente do proposto pela ontologia, o motor de inferência acusará a inconsistência. A Figura 6.7 exemplifica tal situação, onde a propriedade de dado referente ao metadado Essential (número 4.10.4) foi assinalada com o tipo de dado string, quando o correto seria boolean.

Figura 6.7 – Exemplo de inconsistência por tipo de valor



Fonte: Elaborado pelo autor.

6.2.2 Espaço de Valores

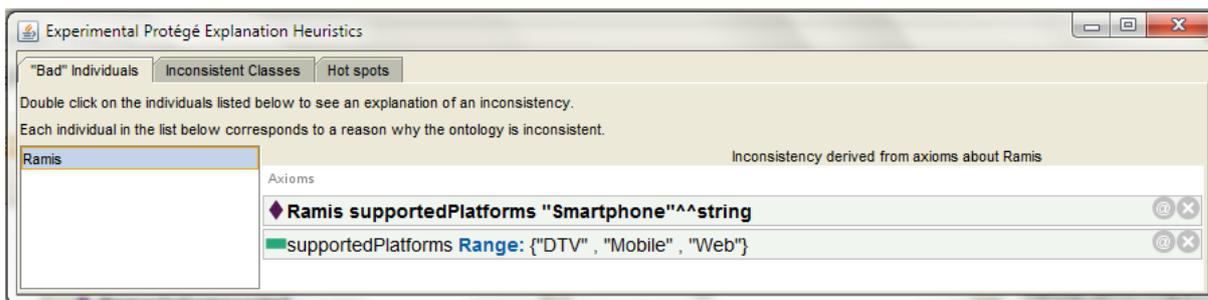
Além da definição de tipo de dado permitido para um escopo de uma propriedade de dado, pode-se delinear um espaço de valores aceitáveis. Utilizaram-se intervalos para valores inteiros, como para o metadado 1.8 Aggregation Level, que permite valores inteiros de 1 a 4. Além disso, valores máximos de caracteres de uma string também foram limitados, como em 5.11 Language que possui até 100 caracteres. Também foram definidos conjuntos de strings para espaço de valores, tais como em 4.8 Supported Platform, onde somente são aceitáveis os valores: Mobile, DTV e Web.

A Figura 6.8 exemplifica uma inconsistência obtida por um valor estar fora do espaço permitido. A partir da explanação podemos constatar que o valor “Smartphone” não está presente no espaço de valores que constitui o escopo da propriedade de dados.

6.2.3 Valor Condicional

Como abordado anteriormente, a lógica de valores condicional é realizada em nível de superclasse da classe correspondente a propriedade de dado. No seu escopo, todos os valores

Figura 6.8 – Exemplo de inconsistência por espaço de valor



Fonte: Elaborado pelo autor.

possíveis para a propriedade de dado são definidos.

Deste modo, mesmo com um valor dentro do espaço válido de escopo pode-se obter uma inconsistência, pois para ele ser totalmente válido depende de um valor de uma outra propriedade. Este fato é ilustrado pela Figura 6.9. Quando a propriedade de dado `specificAspectRatio` possuir o valor “4:3”, o espaço de valores possível válido para a propriedade de dado `specificStandardResolution` passa a ser somente: “1152x864, 1280x960, 320x240, 448x336, 480x360, 640x480, 768x576”. Note que o valor de `specificStandardResolution`, “960x640” será um valor válido quando o valor de `specificAspectRatio` for “3:2”.

6.2.4 Cardinalidade

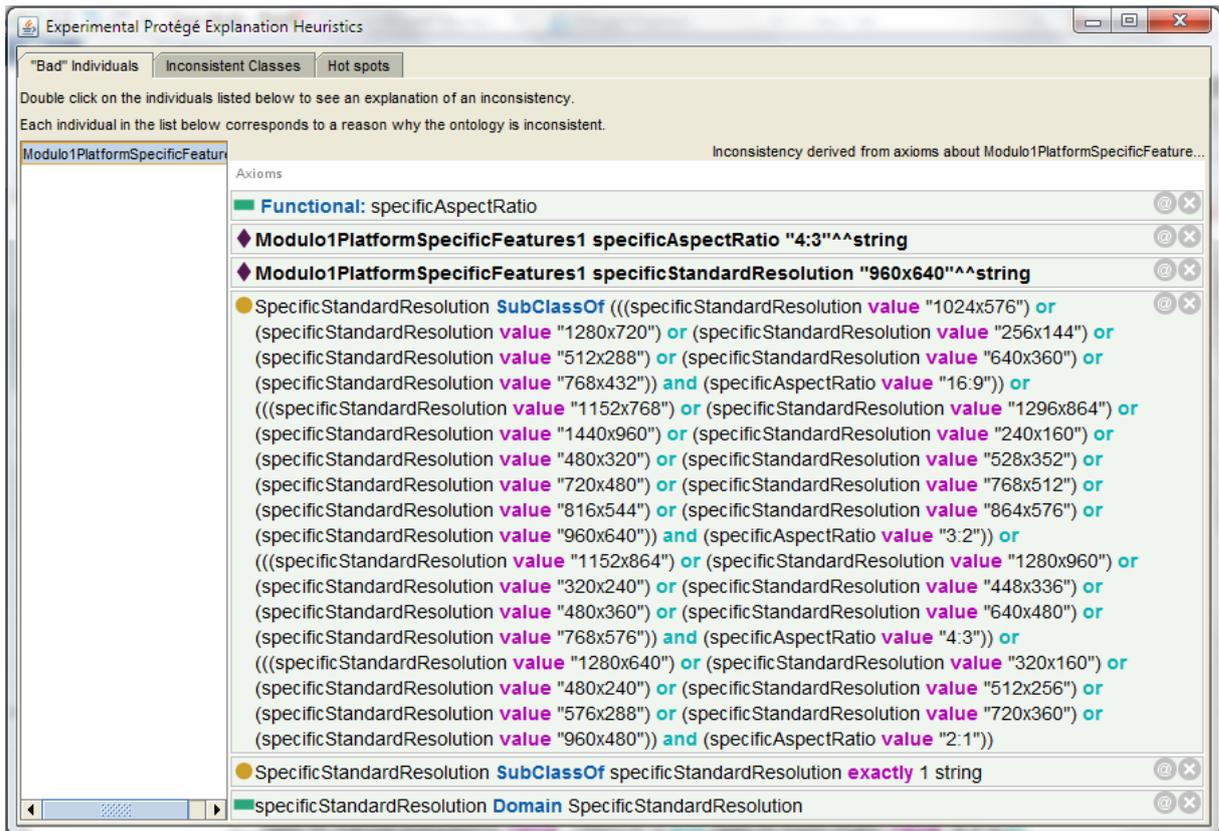
Da mesma forma que a lógica de valores condicionais, as cardinalidades também foram definidas em nível de superclasse. Assim, é possível verificar quando uma propriedade é associada mais vezes que o permitido a um indivíduo.

A Figura 6.10 exemplifica esta verificação. O metadado 5.7 `Typical Age Range` só pode ter até cinco ocorrências. Como no exemplo foram criadas seis propriedades, gerou-se uma inconsistência.

6.3 Processo de Inferência sobre Perfis de Aplicação

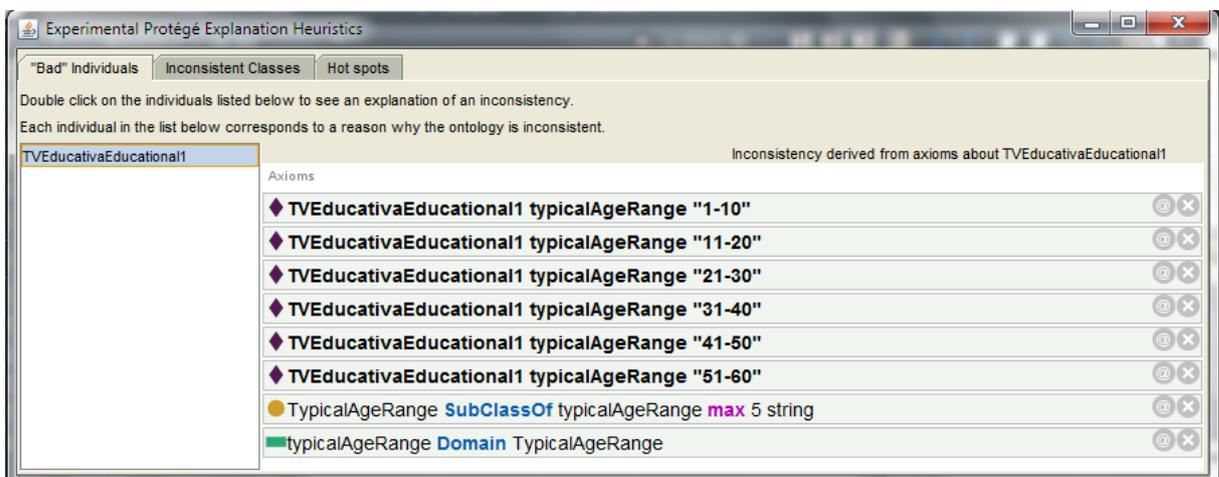
Através das ontologias dos OAs e dos perfis de aplicação importados em uma mesma ontologia, também é possível que o motor de inferência classifique os OAs como membros da classe de equivalência que representa o perfil de aplicação. Esta classificação se dá quando se infere que todos os conteúdos de metadados foram corretamente valorados.

Figura 6.9 – Exemplo de inconsistência por valor condicional



Fonte: Elaborado pelo autor.

Figura 6.10 – Exemplo de inconsistência por cardinalidade



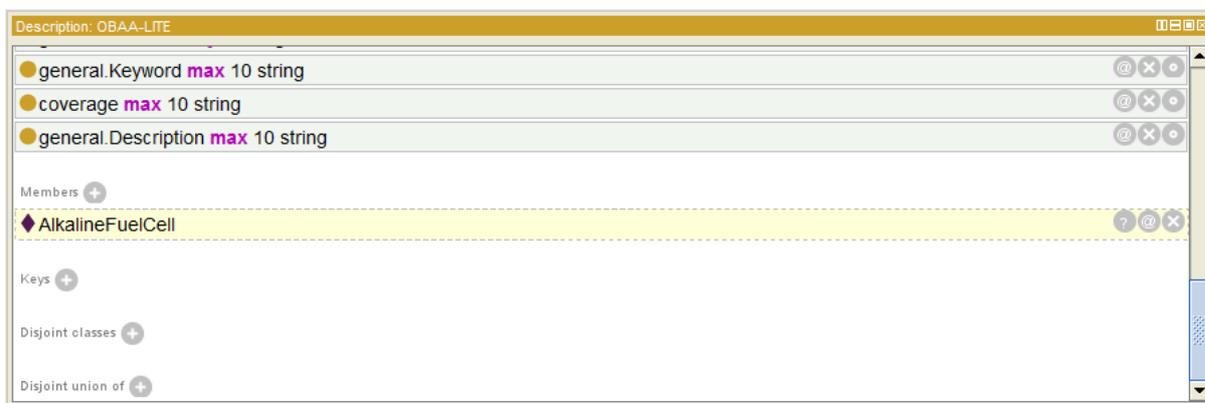
Fonte: Elaborado pelo autor.

É importante salientar que, por motivos de desempenho, nem sempre foi importado mais de um OA em uma mesma ontologia. Isto se deve pelo fato de que quanto maior o número de indivíduos em uma mesma ontologia, maior o tempo de processamento do motor de inferência. A seguir serão apresentados somente as ocorrências em que OAs (descritos anteriormente) são definidos como membros de perfis.

6.3.1 Perfil OBAA-LITE

Em conjunto com a ontologia do perfil OBAA-LITE foi importada a ontologia do OA Célula Combustível Alcalina. A Figura 6.11 ilustra a classificação deste OA como membro do perfil em questão ao final do processo de inferência.

Figura 6.11 – Exemplo de inferência sobre o perfil OBAA-LITE



Fonte: Elaborado pelo autor.

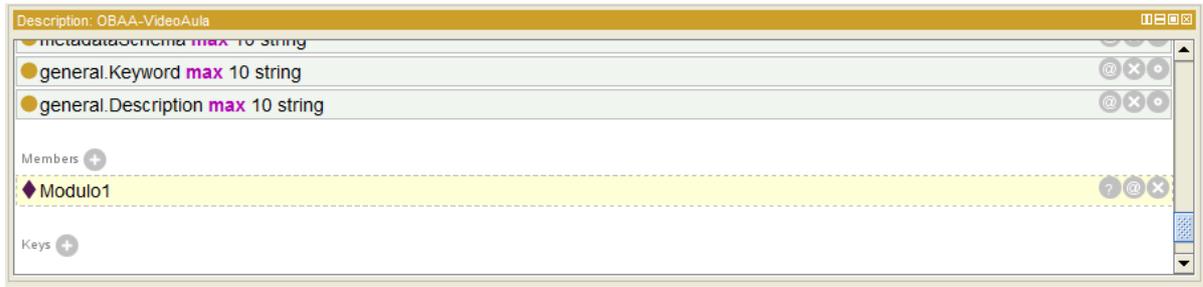
6.3.2 Perfil OBAA-VideoAula

A ontologia referente ao OA Apresentação do Serviço Videoaula@RNP: Módulo 1 foi importada com o perfil OBAA-VideoAula. O resultado após o processo de inferência sobre o perfil é apresentado na Figura 6.12.

6.3.3 Perfil OBAA-DTV

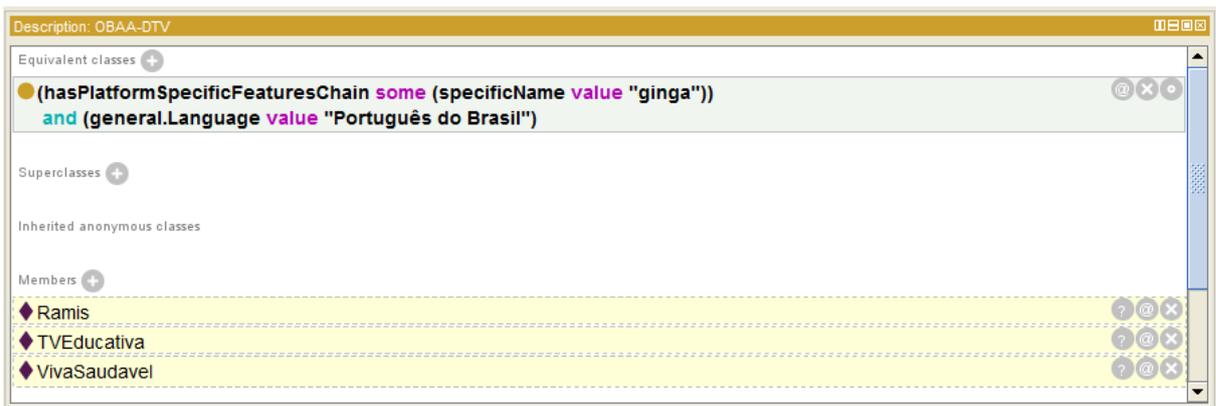
Os OAs Ramis, TV Educativa e Outras Infâncias são importados em conjunto com o perfil OBAA-DTV. Ao final do processo de inferência, e constatação de que os valores definidos foram preenchidos, os OAs são classificados como membros do perfil, como representado na Figura 6.13.

Figura 6.12 – Exemplo de inferência sobre o perfil OBAA-VideoAula



Fonte: Elaborado pelo autor.

Figura 6.13 – Exemplo de inferência sobre o perfil OBAA-DTV



Fonte: Elaborado pelo autor.

7 OBAA-LEME

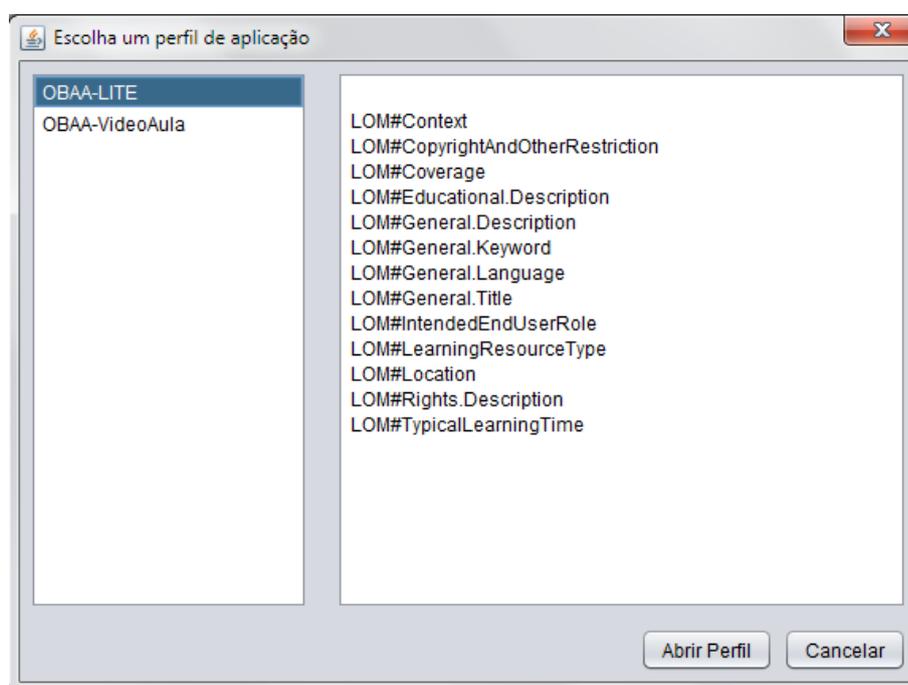
O OBAA-LEME é um editor de conteúdos de metadados para OAs compatível com a Web Semântica. Ele foi desenvolvido utilizando-se da linguagem de programação Java em conjunto com a OWLAPI. A ferramenta carrega as ontologias anteriormente criadas para obtenção da semântica dos metadados.

Ao iniciar-se a aplicação, o usuário deverá sinalizar a localização de seu objeto de aprendizagem. Esta localização poderá ser tanto local (no próprio computador do usuário) ou remota. Em caso de arquivo local, este será carregado no servidor. Por outro lado, se o arquivo for remoto, basta o fornecimento da URI do mesmo. Posteriormente, a ferramenta automaticamente preencherá o campo do metadado Location (número 4.3) com o caminho do arquivo.

7.1 Escolha do Perfil de Aplicação

Em um segundo momento, o usuário irá visualizar os perfis de aplicação existentes. Ao selecionar o perfil, o usuário obterá todos os metadados que compõem o perfil em ordem alfabética precedidos de seu padrão de metadados. A Figura 7.1 apresenta a escolha de perfis e a visualização dos metadados do perfil de aplicação OBAA-LITE.

Figura 7.1 – Janela de escolha de perfil de aplicação



Fonte: Elaborado pelo autor.

Como cada perfil de aplicação é carregado a partir de uma ontologia de perfis¹, para se carregar um novo perfil, este deverá ser definido como uma ontologia e em seguida importado a esta ontologia de perfis. Com isso, a ferramenta automaticamente carregará o novo perfil e adicionará uma opção de seleção do mesmo.

7.2 Preenchimento de Conteúdo

Em seguida a escolha do perfil de aplicação, todos os metadados que fazem parte do perfil são carregados para que seus conteúdos possam ser preenchidos. Também é possível adicionar grupos de metadados container que fazem parte do perfil, desde que tenham cardinalidade maior que um.

A Figura 7.2 apresenta a janela principal do editor de conteúdos com o perfil de aplicação OBAA-LITE selecionado. As abas do perfil são previamente carregadas de acordo com as cardinalidades dos metadados do tipo container. No caso do perfil OBAA-LITE serão duas abas: OBAA-LITE, que contém os metadados dos grupos de metadados container com cardinalidade um, e Educational1, que por possuir cardinalidade maior que um tem sua aba específica. As abas são sinalizadas com coloração vermelha quando ainda existam metadados a serem preenchidos e com verde se todos possuírem algum valor.

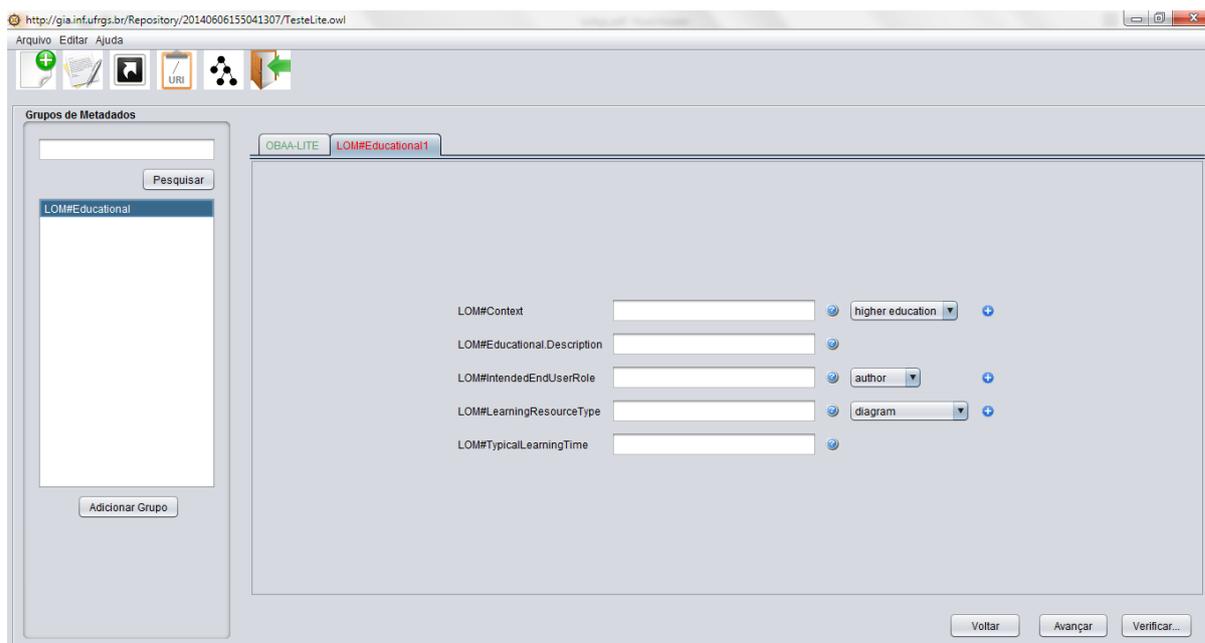
Um botão de informação (“?”) segue ao lado do campo de preenchimento. É possível acessar alguns campos de documentação inseridos nas ontologias, tais como descrição, tipo de dado, exemplo, cardinalidade e espaço de valores.

Quando um metadado possui um espaço de valores definido, uma caixa de seleção é adicionada. Se a cardinalidade do metadado é um, somente a caixa é apresentada. Em caso de cardinalidade maior que um, um campo de texto é precedido à caixa e o usuário pode adicionar valores pressionando o botão “+” ou simplesmente digitando valores separados por vírgulas.

Com isso, a ferramenta se utiliza simplesmente de separação por vírgulas para múltiplos valores de um metadado com cardinalidade maior que um. Por exemplo, “diagram, exercise” como preenchimento do metadado Learning Resource Type (número 5.2) resultará em duas propriedades de dados, uma com o valor “diagram” e outra com “exercise” na ontologia do OA.

¹<http://gia.inf.ufrgs.br/ontologies/Profiles.owl>

Figura 7.2 – Janela principal a partir do perfil OBAA-LITE



Fonte: Elaborado pelo autor.

7.3 Exportação

Até mesmo perfis de aplicação podem ter uma quantidade de metadados relativamente grande, resultando em uma demanda de tempo considerável para o seu total preenchimento. Assim, existe a opção do usuário preencher em parte o conteúdo dos metadados e exportar todos os conteúdos preenchidos para um arquivo local.

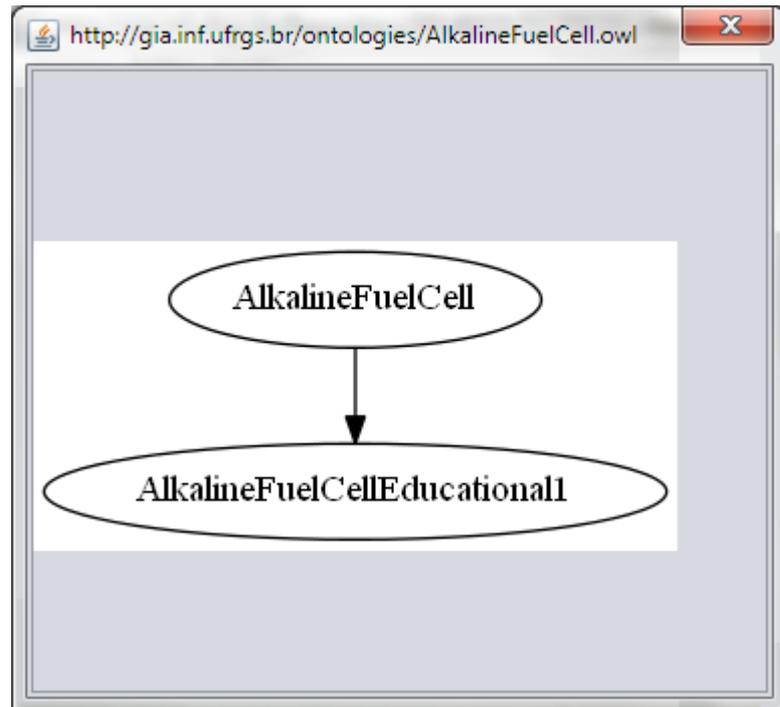
Este arquivo já estará em formato OWL. Logo, é possível que a ferramenta abra o arquivo local e carregue todos os conteúdos de metadados anteriormente preenchidos. Também é possível preencher todos os conteúdos de metadados e adiar a sua publicação.

7.4 Visualização

Ao acessar a opção de visualização do menu, é possível obter todas as relações entre as abas de preenchimento, representadas por elipses. Cada elipse resultará em um indivíduo na ontologia do OA e representará os metadados do tipo container maior que um.

O OA Cálula de Combustível Alcalina tem a sua representação geral ilustrada pela Figura 7.3. Este OA foi representado através do perfil de aplicação OBAA-LITE.

Figura 7.3 – Visualização do OA Célula Combustível Alcalina



Fonte: Elaborado pelo autor.

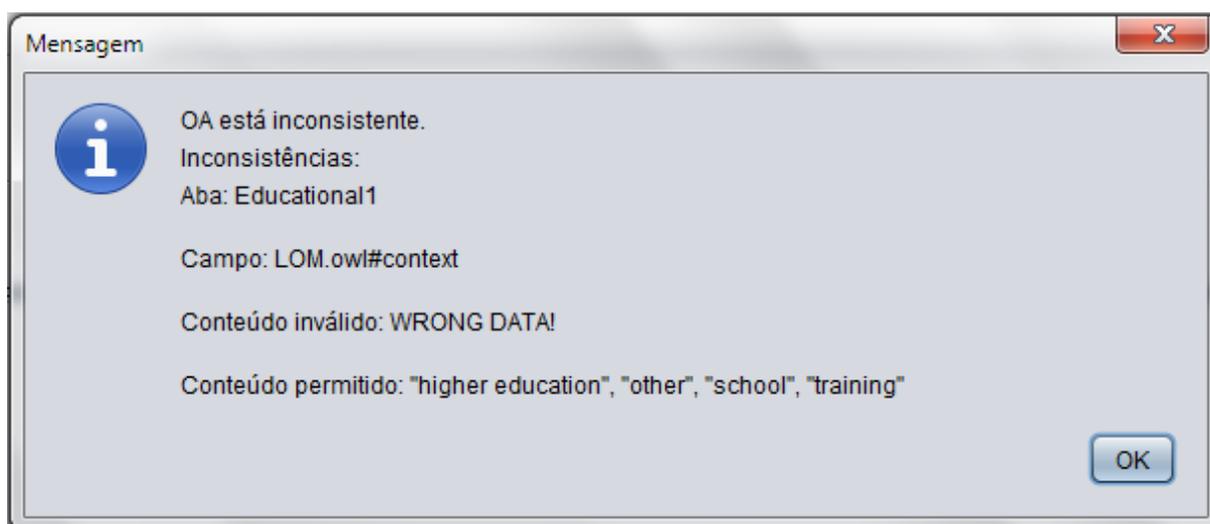
7.5 Verificação de Consistência

No momento em que o usuário desejar a verificação de consistência dos metadados do OA, ele será carregado no servidor para que o motor de inferência seja acionado. Após o tempo de processamento, a aplicação exibirá uma mensagem de resultado. Esta mensagem poderá conter as inconsistências que o OA apresenta ou sinalizar que o OA já está apto para publicação.

Um exemplo de verificação de consistência é referente a adequação ao espaço de valores proposto, se um conteúdo está ou não contido no espaço de valores definido. A Figura 7.4 ilustra uma mensagem de inconsistência do OA capturada pelo motor de inferência, onde o valor preenchido (“WRONG DATA”) não está contido no espaço de valores aceitável (“higher education, other, school, ou training”). A asserção com o valor incorreto é apresentada, em conjunto com o espaço de valores aceitável.

A verificação se todos os metadados do perfil foram preenchidos também pode ser realizada. Outros exemplos de verificação de consistência podem ser a verificação de cardinalidade máxima de um metadado ou valores condicionais.

Figura 7.4 – Exemplo de mensagem de inconsistência



Fonte: Elaborado pelo autor.

7.6 Publicação

Após a verificação de consistência ser concluída e não apresentar inconsistências, o usuário poderá publicar seu OA. Em caso positivo, a ontologia do OA permanecerá no servidor. Caso contrário, a ontologia será removida.

Todas as representações dos OAs são armazenadas em um servidor² aberto. Cada OA é armazenado em uma pasta nomeada por um valor de data e hora³, que objetiva ser uma chave onde a ontologia do OA e o arquivo local do usuário serão armazenados. Através deste armazenamento, esta data e hora permitem que um OA de mesmo nome possa futuramente ser publicado, como por exemplo, para futuro versionamento.

²<http://gia.inf.ufrgs.br/Repository>

³Em inglês: Timestamp.

8 RESULTADOS FINAIS

Neste capítulo serão apresentados alguns exemplos de metadados de OAs representados por ontologias por meio da utilização da ferramenta OBAA-LEME. Todas as representações podem ser acessadas pelo link correspondente.

8.1 Ábaco

A partir dos metadados do OA Ábaco (SILVA, 2011) e do perfil de aplicação OBAA-LITE, uma representação do OA em forma de ontologia foi publicado em: <http://gia.inf.ufrgs.br/Repository/20140402112725493/Abaco.owl>.

A Figura 8.1 ilustra uma captura de tela do preenchimento dos metadados em conjunto com a visualização dos vínculos entre as abas deste OA.

Figura 8.1 – Captura de tela do preenchimento dos metadados do OA Ábaco

The screenshot shows the OBAA-LITE interface with the LOM#Educational1 tab selected. A window in the foreground displays an ontology diagram with two nodes: 'Abaco' (top) and 'AbacoEducational1' (bottom), connected by a downward-pointing arrow. The main interface contains the following metadata fields:

LOM#CopyrightAndOtherRestriction	yes
LOM#Coverage	Brasil
LOM#General.Description	n sua tela inicial, A animação apresenta
LOM#General.Keyword	1a posicional decimal, Soma, Subtração
LOM#General.Language	PT-BR
LOM#General.Title	Ábaco
LOM#Location	acionais2.mec.gov.br/handle/mec/10855
LOM#Rights.Description	anto que atribuam o crédito aos autores.

Fonte: Elaborado pelo autor.

8.2 A Cartomante

A ontologia publicada do OA A Cartomante (SILVA, 2011) tendo como base o perfil de aplicação OBAA-LITE pode ser acessada em: <http://gia.inf.ufrgs.br/Repository/20140402095748993/ACartomante.owl>.

Uma exemplificação de uma dupla entrada para o campo Learning Resource Type é apresentada na Figura 8.2, onde “narrative text, figure” são duas ocorrências do metadado na descrição do OA.

Figura 8.2 – Captura de tela do preenchimento dos metadados do OA A Cartomante

LOM#Context	school	?	school	+
LOM#Educational.Description	lo a curiosidade em ler o texto completo.	?		
LOM#IntendedEndUserRole	learner	?	learner	+
LOM#LearningResourceType	narrative text, figure	?	figure	+
LOM#TypicalLearningTime	PT10M	?		

Fonte: Elaborado pelo autor.

8.3 Infraestrutura do Serviço Videoaula@RNP: Módulo 2

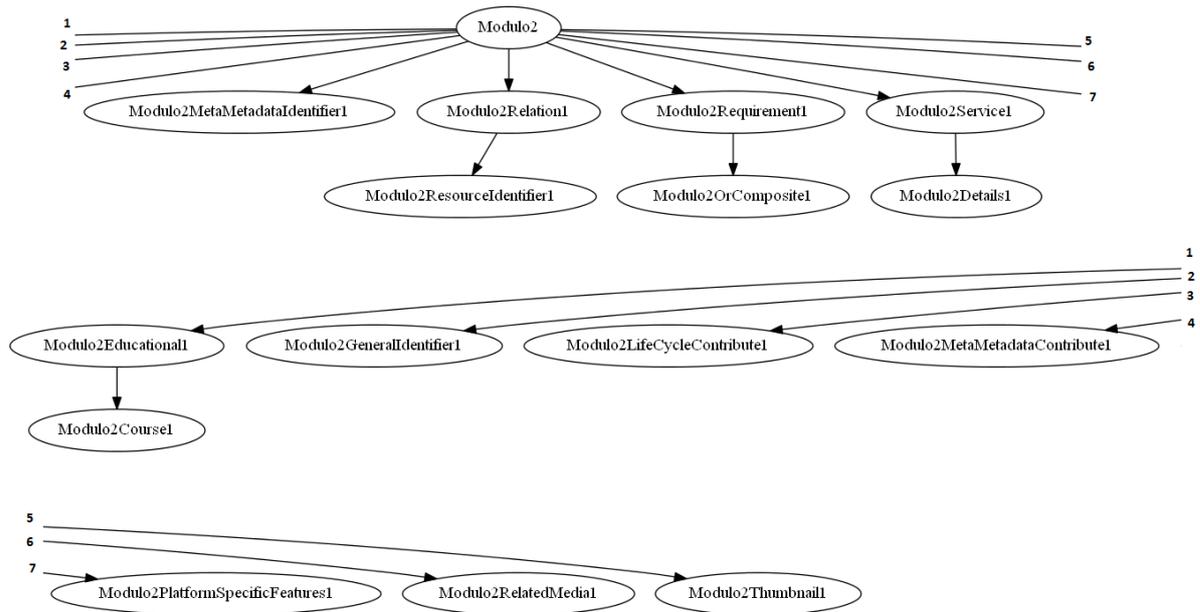
A representação dos metadados⁴ do OA Infraestrutura do Serviço Videoaula@RNP: Módulo 2⁵ em forma de ontologia pode ser acessado em: <http://gia.inf.ufrgs.br/Repository/20140403211232415/Modulo2.owl>.

A Figura 8.3 exibe a visualização do OA obtida através da ferramenta OBAA-LEME. Esta imagem foi editada para fins de ilustração. Cada número corresponde a continuação das linhas. Na ferramenta não existe tal divisão.

⁴http://va05-cps.rnp.br/riotransfer/rnp/treinamentos/videoaulas/modulo_2/modulo_2.xml

⁵http://videoaula.rnp.br/rioflashclient.php?xmlfile=/rnp/treinamentos/videoaulas/modulo_2/modulo_2.xml

Figura 8.3 – Visualização do OA Módulo 2



Fonte: Elaborado pelo autor.

9 CONCLUSÃO

A Web Semântica fornece um ambiente rico de desenvolvimento de aplicações que venham a realizar inferências, consistir e pesquisar dados. A utilização de padrões de formatos tem caráter essencial para que a interoperabilidade de sistemas aconteça na WS.

Até o desenvolvimento deste trabalho, ainda não havia sido propostas ontologias recomendadas para padrões de metadados. Assim, através das tecnologias da WS, o presente trabalho tem como objetivo cooperar com a produção de conteúdo em formato interoperável para descrição de objetos de aprendizagem na WS e sua utilização na educação.

O ciclo de desenvolvimento de uma ontologia é extremamente complexo. Assim, as ontologias aqui propostas não acarretam em versões finais para as mesmas.

Estas ontologias são utilizadas na ferramenta proposta para abstrair em parte o conhecimento em ontologias necessário ao usuário. Assim, ele pode se concentrar no correto preenchimento de conteúdo de metadados e contribuir com a publicação de materiais nesta nova Web.

Com estas publicações, é possível constituir um repositório aberto de objetos de aprendizagem. A partir dele, é possível a realização de consultas e o versionamento dos mesmos.

9.1 Trabalhos Futuros

Este trabalho não apresenta uma versão final das ontologias propostas, bem como uma versão final do editor de conteúdos. Portanto, a seguir são delineados alguns tópicos para o prosseguimento do trabalho.

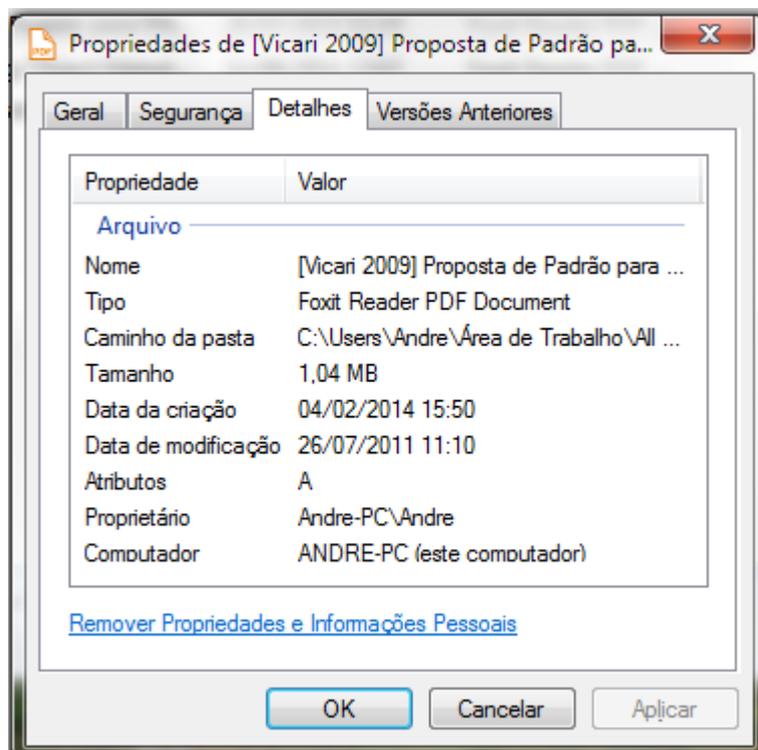
9.1.1 Autopreenchimento do Conteúdo dos Metadados

Um arquivo armazenado em um computador carrega consigo um conjunto de atributos. Alguns atributos são preenchidos automaticamente pela máquina. Além disso, o usuário tem a possibilidade de editar estes atributos.

A Figura 9.1 exemplifica o conjunto de atributos que pode ser obtido de um arquivo. A partir disso, cada atributo pode ser lido e o editor de conteúdos poderia auto preencher metadados correspondentes aos atributos. Por exemplo, poderíamos utilizar o conteúdo do

atributo “Nome” como sugestão de título (1.2. Title), o “Tipo” para o formato (4.1. Format) o “Tamanho” para o tamanho (4.2. Size), entre outros.

Figura 9.1 – Exemplo de conjunto de atributos de arquivo



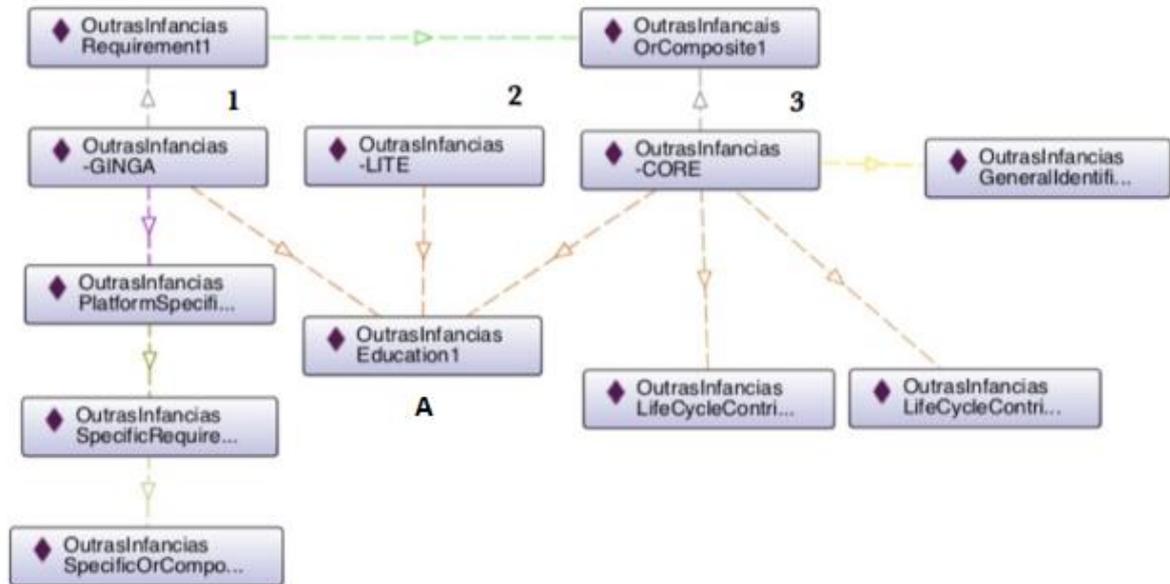
Fonte: Elaborado pelo autor.

9.1.2 Reuso e Versionamento de Objetos de Aprendizagem

Através dos estudos apresentados com este processo de descrição de indivíduos, torna-se possível o reuso de informações de OAs. Casos relacionados a versionamento de OAs não necessitam replicação de conteúdos, apenas relacionamentos através de propriedades. A Figura 9.2 ilustra tal característica, onde as marcas **1**, **2** e **3** apresentam indivíduos principais de um OA e as linhas tracejadas os relacionamentos entre indivíduos, com destaque para a marca **A** que ilustra um indivíduo compartilhado (PRIMO, 2013).

Além disso, os indivíduos principais de um OA, representando diferentes versões, podem ser relacionados através do metadado 7. Relation, representado pela propriedade de objeto hasRelation nas ontologias. Esta utilização é ilustrada pela Figura 9.3, onde as marcas **1**, **2** e **3** apresentam indivíduos principais de um OA e as linhas tracejadas a propriedade de objeto hasRelation, realizando relacionamentos entre indivíduos.

Figura 9.2 – Exemplo de reuso de indivíduos de OAs



Fonte: Primo (2013, p. 100).

O emprego das características acima mencionadas busca uma descrição mais completa de um OA e uma maior reutilização de OAs já descritos. Portanto, é almejado em futuras versões do presente trabalho que tais tópicos sejam implementados.

Figura 9.3 – Exemplo de versionamento de OAs



Fonte: Elaborado pelo autor.

REFERÊNCIAS

- ANTONIOU, G.; VAN HARMELEN, F. Web ontology language: owl. In: STAAB, S; STUDER, R. (Eds). **Handbook on ontologies**. [S.l.]: Springer, 2004. p. 67–92.
- BARGMEYER, B. E. et al. **Metadata standards and metadata registries: An Overview**. [S.l.:s.n.], 2000.
- BERNERS-LEE, T. WWW: past, present, and future. **IEEE Computer**, [S.l.], v. 29, n. 10, p. 69–77, 1996.
- BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. The semantic web. **Scientific American**, [S.l.], v. 284, n. 5, p. 34–43, 2001.
- COMMITTEE, IEEE L. T. S. **IEEE standard for learning object metadata**. [S.l.:s.n.], 2002.
- CONSORTIUM, I. G. L. **IMS accessforall meta-data information model**. [S.l.:s.n.], 2004.
- DUVAL, E. et al. Metadata principles and practicalities. **D-Lib Magazine**, [S.l.], 2002.
- GAROFALAKIS, J.; HATZILYGEROUDIS, I.; KYRIAKOU, P. **A tool for managing domain metadata in a web-based intelligent tutoring system**. [S.l.:s.n.], 2007.
- GHEBGHOUBA, O. et al. **LOIT: an indexing tool based on lom ontology**. [S.l.:s.n.], 2009.
- GRÜNINGER, M.; LEE, J. Ontology applications and design. **Communications of the ACM**, [S.l.]: ACM, v.45, n.2, p.39–41, 2002.
- HORRIDGE, M.; BECHHOFFER, S. The OWL API: a java api for owl ontologies. **Journal Semantic Web**, Amsterdam, v. 2, n. 1, p. 11–21, 2011.
- MIZOGUCHI, R.; IKEDA, M. **Towards ontology engineering**. [S.l.:s.n.], 1996.
- NOY, N. F.; MCGUINNESS, D. L. **Ontology development 101: a guide to creating your first ontology**. [S.l.:s.n.], 2001.
- PRIMO, T. T. **Método de representação de conhecimento baseado em ontologias para apoiar sistemas de recomendação educacionais**. 2013. 120 f. Dissertação (Doutorado em Ciência da Computação) – Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2013.
- PRUNUSKE, A. J. et al. **Using online lectures to make time for active learning**. [S.l.:s.n.], 2012.
- QIN, J.; FINNERAN, C. Ontological representation of learning objects. In: WORKSHOP ON DOCUMENT SEARCH INTERFACE DESIGN AND INTELLIGENT ACCESS IN LARGE-SCALE COLLECTIONS, 2002, [S. l.]: **Proceedings...** Springer, 2002. p. 52–57.

SHADBOLT, N.; BERNERS-LEE, T.; HALL, W. **The semantic web revisited**. IEEE Intelligent Systems, Piscataway: IEEE, v. 21, n. 3, p. 96–101, 2006.

SILVA, J. M. C. da. **Análise técnica e pedagógica de metadados para objetos de aprendizagem**. 2011. 189 f. Dissertação (Doutorado em Informática na Educação) – Centro Interdisciplinar de Novas Tecnologias na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2011.

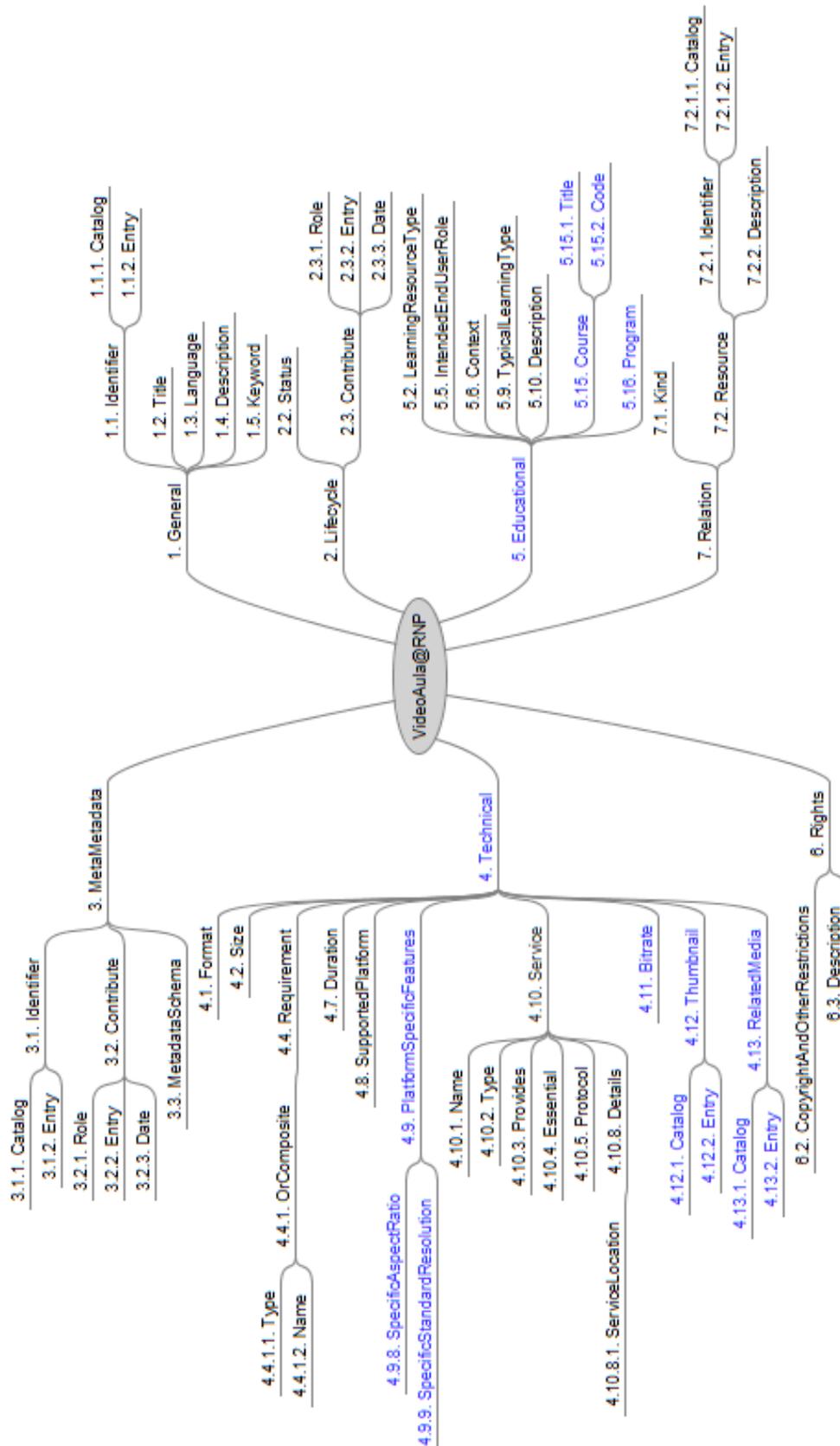
SIMIONI, A. et al. **Mecanismos para criação de conteúdo interoperável entre web, tv digital e móveis**. [S.l.:s.n], 2009.

STUDER, R.; BENJAMINS, R.; FENSEL, D. **Knowledge engineering: principles and methods**. Data and knowledge engineering, [S.l.]: Elsevier B.V., v. 25, p. 161–197, 1998.

VICARI, R. M. et al. **Proposta de padrão para metadados de objetos de aprendizagem multiplataforma**. [S.l.:s.n], 2009.

WARREN, P. **Ontology users survey summary of results**. [S.l.:s.n], 2013.

APÊNDICE A MAPA MENTAL DO PADRÃO DE METADADOS VIDEOAULA@RNP



ANEXO A MAPA MENTAL DO PADRÃO DE METADADOS OBAA

