



CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS

COMUNIDADE EVANGÉLICA LUTERANA "SÃO PAULO"
Recredenciado pela Portaria Ministerial nº 3.607 - D.O.U. nº 202 de 20/10/2005

Douglas Brito

**DEFINIÇÃO E IMPLEMENTAÇÃO DE OBJETOS DE APRENDIZAGEM
A PARTIR DE UMA ONTOLOGIA**

**Palmas
2012/1**



CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS

COMUNIDADE EVANGÉLICA LUTERANA "SÃO PAULO"
Recredenciado pela Portaria Ministerial nº 3.607 - D.O.U. nº 202 de 20/10/2005

Douglas Brito

DEFINIÇÃO E IMPLEMENTAÇÃO DE OBJETOS DE APRENDIZAGEM A PARTIR DE UMA ONTOLOGIA

Projeto apresentado como requisito parcial da disciplina de Trabalho de Conclusão de Curso II (TCC II) do curso de Sistemas de Informação, orientado pela Professora Mestre Parcilene Fernandes de Brito.

**Palmas
2012/1**



CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS

COMUNIDADE EVANGÉLICA LUTERANA "SÃO PAULO"
Recredenciado pela Portaria Ministerial nº 3.607 - D.O.U. nº 202 de 20/10/2005

DOUGLAS BRITO

DEFINIÇÃO E IMPLEMENTAÇÃO DE OBJETOS DE APRENDIZAGEM A PARTIR DE UMA ONTOLOGIA

Projeto apresentado como requisito parcial da disciplina de Trabalho de Conclusão de Curso II (TCC II) do curso de Sistemas de Informação, orientado pela Professora Mestre Parcilene Fernandes de Brito.

Aprovada em 21 de Junho de 2012.

BANCA EXAMINADORA

Prof. M.Sc. Parcilene Fernandes de Brito

Centro Universitário Luterano de Palmas

Prof. M.Sc. Fabiano Fagundes

Centro Universitário Luterano de Palmas

Prof. M.Sc. Edeilson Milhomem da Silva

Centro Universitário Luterano de Palmas

Palmas
2012/1

AGRADECIMENTOS

Primeiramente, venho a agradecer a Deus, por ter me concedido vida a cada novo amanhecer, por me dar saúde, paciência, força e apoio para conseguir enfrentar todos os obstáculos encontrados durante o processo de Graduação. Obrigado por tudo Deus!

Dedico este trabalho “in memoriam” ao meu avô paterno (Adelson Otávio de Brito) e aproveito também para agradecê-lo, por ter sido um avô presente e principalmente um grande amigo. Sempre nas despedidas me desejava saúde e sabedoria para conseguir alcançar meus objetivos. Um exemplo de ser humano a ser espelhado, pois “nos fez acreditar que a vida sempre vale a pena e nenhum sofrimento pode nos tirar a força para levantar todos os dias e continuar essa breve jornada que é nossa passagem nesse imenso mundo de Deus” (Parcilene, 2010).

É difícil agradecer a todas as pessoas que de certa forma, nos momentos apreensivos e aprazíveis, fizeram ou fazem parte da minha vida, por isso primeiramente agradeço à todos de coração. Agradecimento especial aos meus pais (Edilson e Luiza), meus irmãos (Tayrine, Renan e Lucas), a minha sobrinha (Ana Luiza), aos meus avós (Albertina, Joaquim e Jeronima), a minha tia/orientadora/mãe (Parcilene), a minha querida namorada (Tailla), aos tios/professores/pais (Fabiano, Jackson, Edeilson, Cristina, Madianita, Fernando, Andrez), aos amigos (Bruno, Cassimiro, Diogão, Fábio Castro, João Paulo, Marquim, Raphael, Vinícius) aos amigos de faculdade (Andrew, Bruno Vilar, Cleydiane, Diego, Douglas Neves, Elias, Fábio Poka, Luane, Lucas George, Naara, Nyl, Ricardo, Thearlismar, Warley, Valdirene), aos sogros/cunhada (Lisboa, Dinalva e Monique), aos tios/padrinhos (Godo, Rosa, Valdeci, Ivanilde, Edson, Pascual, Creusa).

SUMÁRIO

1. INTRODUÇÃO	11
1.1. Objetivos	14
1.1.1. Objetivo Geral.....	14
1.1.2. Objetivos Específicos.....	14
1.2. Justificativa.....	14
1.3. Problema.....	15
1.4. Hipótese.....	15
2. REFERENCIAL TEÓRICO	16
2.1. Objeto de Aprendizagem.....	16
2.1.1. Definição.....	16
2.1.2. Características	17
2.1.2.1. Reusabilidade	17
2.1.2.2. Adaptabilidade.....	18
2.1.2.3. Granularidade	19
2.1.2.4. Acessibilidade.....	22
2.1.2.5. Durabilidade	23
2.1.2.6. Interoperabilidade.....	24
2.1.3. LOM.....	25
2.1.3.1. Geral	26
2.1.3.2. Ciclo de Vida.....	28
2.1.3.3. Meta-metadados.....	29
2.1.3.4. Técnico	29
2.1.3.5. Educacional	31
2.1.3.6. Direito.....	33
2.1.3.7. Relação	33
2.1.3.8. Anotação.....	34
2.1.3.9. Classificação.....	34
2.2. Ontologia.....	35
2.2.1. Características	40
2.3. Trabalhos Correlatos	43
2.3.1. Ontologia para Objetos de Aprendizagem (OntoLo).....	43

2.3.2. Java Learning Object Ontology – JLOO	48
3. MATERIAIS E MÉTODOS	53
3.1. Local e Período.....	53
3.2. Materiais.....	53
3.2.1. Hardware.....	53
3.2.2. Software	53
3.2.3. Fontes de Referências Bibliográficas.....	54
3.3. Metodologia	55
4. RESULTADOS E DISCUSSÃO	59
4.1. Arquitetura do Konnen.....	59
4.2. Arquitetura do Módulo <i>Content</i>	62
4.3. Representação de Conteúdos como Objetos de Aprendizagem	70
4.3.1. Geral.....	72
4.3.2. Anotação	76
4.3.3. Ciclo de Vida	78
4.3.4. Técnico e Direito.....	80
4.3.5. Educacional.....	81
4.4 Ontologia no modelo LOM+OA	82
4.4.1. Axioma.....	85
5. CONSIDERAÇÕES FINAIS.....	93
REFERÊNCIAS BIBLIOGRÁFICAS	95
ANEXOS	99

RESUMO

A busca por meios que auxiliam o processo de ensino-aprendizagem têm impulsionado os especialistas a utilizarem/criarem ferramentas capazes de facilitar o aprendizado. A implantação de uma rede social acadêmica em uma instituição de ensino tende a contribuir para esse processo. O presente trabalho consiste em apresentar a definição e implementação de Objetos de Aprendizagem para o módulo “Material Didático” de uma Rede Social Acadêmica (Konnen), tendo como base a utilização de um conjunto de metadados que formarão a ontologia do domínio de forma a contextualizar seus objetos.

PALAVRAS CHAVE: Ontologia, Objeto de Aprendizagem, Redes Sociais Acadêmica.

LISTA DE FIGURAS

Figura 1 - Diagrama de Pacotes do Konnen: Sistema Principal e Subsistemas	12
Figura 2 - Exemplo de Reusabilidade	18
Figura 3 - Exemplo de Adaptabilidade.....	19
Figura 4 - Reutilização versus tamanho do componente (Adaptada PRIETO-DIÁZ, 1996, p. 5).....	20
Figura 5 – Exemplo de Granularidade.....	20
Figura 6 - Níveis de Granularidade	21
Figura 7 - Acessibilidade.....	23
Figura 8 – Estrutura do esquema LOM v1.0	25
Figura 9 - Exemplo de Axioma aplicado na Rede Social Acadêmica.....	36
Figura 10 - Exemplo de Classes, Relações e Instâncias no Domínio “Vinho” (NOY e MCGUINNESS, 2002, p. 4).	38
Figura 11 - Ontologia para o domínio material didático de uma rede social acadêmica.	39
Figura 12 - Processo de Evolução da Ontologia (Adaptada STOJANOVIC, 2004, p. 76) ..	41
Figura 13 - Hierarquia de classes de ontologias (Adaptada de WANG, FANG e FAN, 2008, p. 473).....	44
Figura 14 - Formação de Acorde (Nível Fundamental)	45
Figura 15 - Vídeo aula sobre ritmo de música - Nível combinado fechado (YOUTUBE, 2008, online, mim 0:15).....	46
Figura 16 - Letra de música + Acordes + Vídeo Aula integrada - Nível Combinado Aberto (CIFRACLUB, 2011, online).	46
Figura 17 – Exemplo de Instruções de Declaração	49
Figura 18 - Exemplo de <i>if e else</i>	49

Figura 19 - Exemplo de <i>switch</i>	49
Figura 20 - Exemplo de Recursividade	50
Figura 21 - Hierarquia de subclasse do "modelo de dados" (LEE, 2005, p. 3).....	50
Figura 22 - Hierarquia de subclasse de "Teste e depuração" (LEE, 2005, p. 4).	51
Figura 23 - Metodologia do Projeto	55
Figura 24 - Aplicativos (subsistemas) do Konnen (Modificado de SOUZA, <i>et al</i> , 2012, p.5).60	
Figura 25 - Estrutura de Pastas para a criação de um módulo.....	60
Figura 26 - Níveis de Aplicativos Fundamentais para o funcionamento do Konnen (SOUZA, <i>et al</i> , 2012).	61
Figura 27 - Arquitetura do Gerenciador de Conteúdo.....	62
Figura 28 – Parte da View ManagerContent, Método CreateContent e Método Compartilhar (BRITO, 2011 p. 49).....	64
Figura 29 - Action Save do Controller Content.....	65
Figura 30 - Action Save do Controller Content.....	66
Figura 31 - Conteúdo publicado via Grupo	67
Figura 32 - Action Share do Controller Content	68
Figura 33 - Tela da lista de amigos e grupos de um usuário	69
Figura 34 - Estrutura do Metadado de Objeto de Aprendizagem baseado no LOM.	70
Figura 35 - Modelo Relacional do Metadado de Objeto de Aprendizagem.....	71
Figura 36 - Metadado de Objeto de Aprendizagem (Geral - A).....	72
Figura 37 - Action StructureSave do Controller Content	72
Figura 38 - Parte da Tela do Gerenciador de Conteúdo (Geral).....	74
Figura 39 - Estrutura (Em Rede)	74
Figura 40 - Estrutura (Hierárquica)	75
Figura 41 - Estrutura (Linear).....	76

Figura 42 - Metadado de Objeto de Aprendizagem (Anotação)	76
Figura 43 - Action Anotacao do Controller Content	77
Figura 44 - Parte da Tela do Gerenciador de Conteúdo (Anotação)	77
Figura 45 - Metadado de Objeto de Aprendizagem (Ciclo de Vida).....	78
Figura 46 - Método Save do Manager Content	79
Figura 47 - Parte da Tela do Editor de Conteúdo Completo (Ciclo de Vida)	79
Figura 48 - Metadado de Objeto de Aprendizagem (Técnico).....	80
Figura 49 - Metadado de Objeto de Aprendizagem (Educativo).....	81
Figura 50- Parte da Tela do Editor de Conteúdo Completo (Educativo).....	82
Figura 51 - Estrutura.....	83
Figura 52 - SELECT * FROM na Tabela Group (1).....	84
Figura 53 - SELECT * FROM na Tabela Group (2).....	84
Figura 54 - SELECT * FROM na Tabela Content	85
Figura 55 - Diagrama que define a relação Interdisciplinaridade	86
Figura 56 - View Axioma (Interdisciplinaridade)	87
Figura 57 - Diagrama que define a relação Multidisciplinaridade	88
Figura 58 - View Axioma (Multidisciplinaridade).....	90
Figura 59 - Diagrama que define a relação Transversalidade	91
Figura 60 - View Axioma (Transversalidade).....	92
Figura 61 - Parte do Modelo Relacional (Banco de Dados).....	105

LISTA DE TABELA

Tabela 1 - Categoria Geral - LOM	26
Tabela 2 - Categoria Ciclo de Vida - LOM	28
Tabela 3 - Categoria Meta-metadados - LOM.....	29
Tabela 4 - Categoria Técnico - LOM	30
Tabela 5 - Categoria Educacional - LOM	31
Tabela 6 - Categoria Direito - LOM.....	33
Tabela 7 - Categoria Relação - LOM	33
Tabela 8 - Categoria Anotação - LOM.....	34
Tabela 9 - Categoria Classificação - LOM.....	34
Tabela 10 - Hierarquia de objetos de aprendizagem – Adaptada de Wiley (2000, p. 77).....	44
Tabela 11 - Uma instância e parte de suas propriedades – Adaptada de Wang (2008, p. 474)	47
Tabela 12 - Requisito Gerenciar Conteúdo.	99
Tabela 13 - Requisitos Funcionais- Criar Conteúdo.	100
Tabela 14 - Requisitos Não Funcionais - Criar Conteúdo.....	100
Tabela 15 - Requisitos Funcionais - Inicial Gerenciador de Conteúdos	101
Tabela 16 - Requisito Gerenciar Galeria.	102
Tabela 17 - Requisito Gerenciar Coleções.	102
Tabela 18 - Requisito Gerenciar Permissões.....	102
Tabela 19 - Requisito Gerenciar Compartilhamento.	103
Tabela 20 - Dicionário de Dados da Entidade Content	105
Tabela 21 - Dicionário de Dados da Entidade file_on_disk.....	107
Tabela 22 - Dicionário de Dados da Entidade User	107
Tabela 23 - Dicionário de Dados da Entidade User_relationships	108

1. INTRODUÇÃO

Com as inovações ocorridas no contexto das Tecnologias da Informação e Comunicação, cada vez mais a internet e as redes sociais fazem parte do cotidiano das pessoas. Esse fato pode ser evidenciado, segundo Goulart (2011), em uma pesquisa realizada pelo Ibope, a qual apontou que 87% dos usuários de internet do Brasil utilizam uma rede social e, destes, 83% utilizam esses serviços para finalidades pessoais. Este número vem crescendo exponencialmente, pois inovações tecnológicas estão facilitando o acesso à internet e, conseqüentemente, a diversas redes sociais. Além disso, podem-se citar os três elementos tecnológicos que a internet integra: processadores, capacidade de armazenamento e largura de banda.

Conforme Anderson (2009) a cada dois anos o preço da unidade de processamento em um computador cai pela metade e o preço da largura de banda e armazenagem está caindo em um intervalo de tempo menor. Com a deflação desses elementos, adquirir computadores mais potentes, softwares com maior qualidade e acesso mais rápido à internet vem se tornando cada vez mais comum. Este desenvolvimento propiciou um grande crescimento da quantidade de informações e conhecimentos disponíveis na web. Como consequência desse fato, um dos maiores desafios dos profissionais da informação é gerenciar e recuperar esta gama de informações de forma precisa. Para facilitar o gerenciamento das informações é importante estruturá-las, logo as ontologias conceituadas por Staab e Stuber (2004, apud DAVIES, 2006, p. 4) como conceitos, relações, instâncias e axiomas de um determinado domínio, podem descrever os relacionamentos entre objetos e armazenar informações semânticas dos mesmos para automatizar e facilitar o processo de gerenciamento da informação.

O aumento significativo de usuários da internet tem popularizado a utilização de novas tecnologias que apoiam as ações em ambientes acadêmicos e possibilitam novas formas de ensino-aprendizagem. Associando os recursos tecnológicos que as TICs disponibilizam com o conceito de Objetos de Aprendizagem (IEEE (2001, p.4) define que “um Objeto de Aprendizagem é qualquer entidade, digital ou não digital, que pode ser usada para o aprendizado, educação ou treinamento”.) é possível desenvolver materiais educacionais com objetivos pedagógicos que apoiem o processo de ensino-aprendizagem por meio da web. Assim, a internet pode maximizar esse processo e ser um meio eficiente para disseminação de técnicas didáticas.

Para que exista uma representação da compreensão do significado semântico das informações, tanto para o usuário final, quanto para o sistema, faz-se necessário o uso de Ontologia. Este recurso pode ser aplicado de diversas maneiras, como, por exemplo, para realizar a representação semântica dos dados ou criar axiomas a partir de informações contidas no sistema. Desta forma, a partir da agregação desse conceito (4-tupla <C,R,I,A> conceito, relação, instancia e axioma), é possível criar estratégias computacionais para a identificação de grupos de metadados de objetos de aprendizagem, dando-lhes sentido de acordo com seu contexto.

Este projeto faz parte da rede social acadêmica Konnen, na qual é estruturada em uma arquitetura baseada em módulos, ou seja, um módulo seria uma unidade de um sistema, desenvolvido independentemente. A Figura 1 apresenta os principais módulos que compõem a rede social acadêmica, sendo eles: Content, Profile, Social, ActivityStream e Security. O módulo em destaque, Content, é o foco deste trabalho.

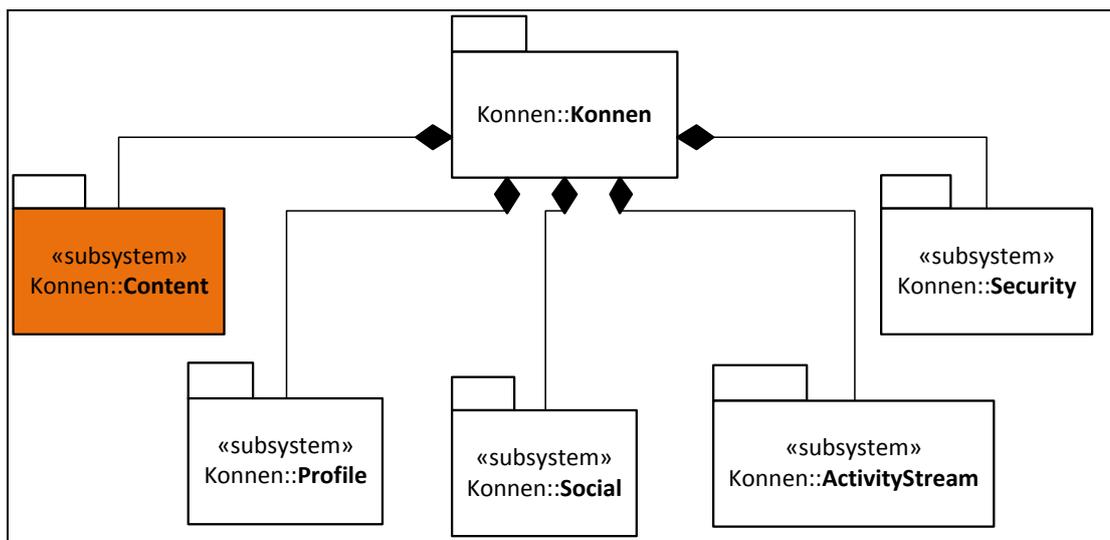


Figura 1 - Diagrama de Pacotes do Konnen: Sistema Principal e Subsistemas

O módulo de Conteúdo é responsável por permitir que os usuários criem, editem, pesquisem e compartilhem conteúdos do tipo: texto, vídeo, áudio, imagens, entre outras mídias com outros usuários da rede. Contudo, esses conteúdos que são cadastrados na rede não seguem uma padronização de estruturação e também não é levada em consideração a relevância da compreensão dos elementos que compõem o domínio e suas relações. A não padronização, o cadastramento incorreto das características relevantes e as relações de um determinado conteúdo podem influenciar na recuperação da informação, ou seja, os resultados retornados podem não condizer com a expectativa do usuário final.

Segundo Vilar (2006, p. 43) ao analisar este problema, por mais que a instituição estabeleça um conjunto de regras e aplicações para que as descrições desses conteúdos cadastrados possam ser relacionadas corretamente ao conteúdo, ainda existem vários problemas quanto ao reconhecimento deste material à sua forma de utilização, como:

A incapacidade de estabelecer a co-relação entre o valor de uma característica de um material com a característica de outros materiais, a falta de flexibilidade que permita trabalhar um conjunto de informações a diferentes recursos que estão presentes em um domínio, a inabilidade de estabelecer a relação que um determinado recurso possui com outros e também a incapacidade em manter um histórico de participações no processo de construção de um material. (VILAR, 2006, p. 43 e 44)

Com a descrição dos objetos e sua contextualização, o sistema poderá sugerir um determinado conteúdo para um aluno que esteja com dificuldade em aprender uma determinada disciplina, fazendo com que ele tenha facilidade em encontrar conteúdos que possam contribuir para o seu estudo. Assim, no contexto apresentado, ter informações sobre os dados de cada objeto (metadados) é um passo fundamental para a atribuição do fator “aprendizagem” ao objeto. Com base nisso, este trabalho consiste em definir as estruturas de metadados que descreverão os objetos de aprendizagem e, conseqüentemente, comporão a ontologia.

O trabalho foi estruturado através das seções: o capítulo 2 (Referencial Teórico) apresenta na seção 2.1 (Objetos de Aprendizagem) os principais conceitos relacionados a OA, introduzindo-os, apresentando as características e, por fim, a descrição detalhada do LOM (*Learning Object Metadata*). A seguir (Seção 2.2) é apresentada a definição de Ontologia, aplicação do conceito em um domínio, razões para o desenvolvimento de uma Ontologia, como também suas características, classificação e o processo de evolução de uma ontologia. Na seção 2.3 são discutidos os Trabalhos Correlatos, onde são apresentados a explicação e o funcionamento de Ontologias para Objetos de Aprendizagem (OntoLo) e JAVA ontologia de Objetos de Aprendizagem (JLOO).

O Capítulo 3 descreve a metodologia adotada para o desenvolvimento deste projeto; e no seguinte capítulo, Capítulo 4, é apresentado o Cronograma, evidenciando o planejamento das etapas a serem cumpridas para a realização da pesquisa com o período de início e uma

estimativa de tempo gasto em sua execução. Por fim, no Capítulo 5, são apresentadas as referências bibliográficas utilizadas neste projeto.

1.1. Objetivos

1.1.1. Objetivo Geral

O objetivo geral deste projeto consiste em definir os metadados que estabelecerão os objetos de aprendizagem, de forma a alicerçar a definição da ontologia dos materiais didáticos de uma rede social acadêmica.

1.1.2. Objetivos Específicos

- Estudar os conceitos de Objetos de Aprendizagem, LOM e Ontologia;
- Definir os elementos que comporão os metadados dos Objetos de Aprendizagem da Rede Social Acadêmica (Konnen);
- Criar mecanismos para agregar os metadados dos Objetos de Aprendizagem de forma a estabelecer a Ontologia;
- Definir, com auxílio de especialistas do domínio, alguns axiomas para a ontologia;
- Implementar, baseado nos objetivos anteriores, o módulo de conteúdos da rede social acadêmica Konnen.

1.2. Justificativa

Associar o potencial que a tecnologia proporciona às novas formas de gerir o conhecimento é algo comum entre as Instituições de Ensino. A utilização de conteúdos educacionais digitais torna-se cada vez mais fundamental para estimular o estudo autônomo dos alunos e a forma como os conteúdos são disponibilizados tem um importante papel nesse contexto. Isso porque com a utilização dos Objetos de Aprendizagem e ontologias, o aluno poderá ter informações contextualizadas do material que é postado pelo professor, o que facilitará a organização de seus estudos, desde o entendimento da relação entre o material e os tópicos que compõem o plano de ensino, até a possibilidade de vincular ao objeto anotações de estudo que nortearão seu processo de aprendizagem.

No Centro Universitário Luterano de Palmas (CEULP/ULBRA) está sendo desenvolvida uma rede social acadêmica (Konnen). O projeto em questão é um dos módulos dessa rede e está relacionado mais especificamente ao módulo de gerenciamento de conteúdo. Com a utilização de objetos de aprendizagem a partir de uma ontologia de materiais didáticos, será possível ter um entendimento amplo das propriedades e características das classes e relacionamentos existentes no contexto acadêmico. Isso propiciará ao sistema a realização de inferências sobre conceitos preponderantes na concepção dos cursos, como a questão da interdisciplinaridade, da multidisciplinaridade etc, além de propiciar a modularização dos conteúdos, o relacionamento do plano de ensino com o material disponibilizado pelo professor, a sistematização das informações e a verificação automática da atualização dos materiais, tendo como base as versões dos objetos.

1.3. Problema

Objetos de Aprendizagem aliados ao conceito de Ontologia podem contribuir para tornar mais eficiente o processo de definição, contextualização e manipulação do material didático de uma Rede Social Acadêmica?

1.4. Hipótese

Para a resolução do problema apresentado na seção anterior, pensou-se na seguinte hipótese:

- Se agregar ao elemento Material Didático de uma Rede Social Acadêmica o conceito de Objeto de Aprendizagem, tendo como base a implementação de um conjunto de metadados, então será possível construir uma ontologia do domínio de forma a contextualizar seus objetos.

2. REFERENCIAL TEÓRICO

A presente seção abordará conceitos referentes a Objetos de Aprendizagem (OA), bem como o Metadado de Objetos de Aprendizagem (LOM). Apresentará também conceitos de Ontologia e, por fim, a aplicação de Ontologia para Objetos de Aprendizagem.

2.1. Objeto de Aprendizagem

O aumento significativo e a popularização da internet vêm propiciando cada vez mais a utilização de novas tecnologias, estratégias e ferramentas que apoiam a aprendizagem e possibilitam novas formas de ensino-aprendizagem. Considerando que a demanda por essas tecnologias aumentam gradativamente, é indispensável obter novas ideias que minimizem tais esforços. De acordo com Zanette (2006, p. 2) as Tecnologias da Informação e Comunicação (TICs) são usadas com bastante frequência na educação. O uso dos recursos tecnológicos das TICs permite que sejam realizadas pesquisas, antecipações e simulações, assim como a criação de soluções e construção de novas formas de representação mental.

Associando os recursos tecnológicos que as TICs disponibilizam com o conceito de Objetos de Aprendizagem (OA) é possível desenvolver materiais educacionais com objetivos pedagógicos que apoiem o processo de ensino-aprendizagem por meio da web.

2.1.1. Definição

Atualmente existe uma grande quantidade de softwares e conteúdos educacionais disponíveis na web de forma gratuita que auxiliam no processo de aprendizagem. Porém, na maioria das vezes, a forma como são produzidos os materiais pedagógicos impossibilita a reutilização dos mesmos. Esse é um dos grandes desafios encontrados nessa área, pois a reutilização de um conteúdo auxilia na aplicação em diferentes contextos, assim uma das soluções em evidência é a utilização de Objetos de Aprendizagem reutilizáveis, adaptáveis, acessíveis e interoperáveis. A seguir serão apresentadas algumas definições, as características, exemplos de utilização e uma contextualização do conceito.

Não há um consenso na definição do termo “Objeto de Aprendizagem”. Assim, enquanto L'Allier, (1997, online) define Objeto de Aprendizagem “como a menor experiência de estrutura autônoma que contém um objetivo, uma atividade de aprendizagem e uma avaliação”, a definição proposta pela IEEE (2001, p.4) estabelece que “um Objeto de Aprendizagem é qualquer entidade, digital ou não digital, que pode ser usada para o aprendizado, educação ou treinamento”. A primeira definição apoia-se no conceito da autonomia a partir do tripé “objetivo, atividade e avaliação”, e a segunda fornece uma abrangência maior ao conceito de OA, desvinculando-o do meio (digital ou não) e indicando os contextos para sua aplicação.

Wiley (2002, p.7) conceitua OA como “qualquer recurso digital que pode ser reutilizado para apoiar a aprendizagem.”. Esta definição inclui qualquer objeto que possa ser disponibilizado através da internet cuja sistematização ocorre em categorias:

- pequenos recursos digitais reutilizáveis, que são entidades mais simples como uma imagem, um vídeo, um texto;
- grandes recursos digitais reutilizáveis, que podem ser objetos mais elaborados como uma página (contendo vários tipos de mídias), um portal de ensino, dentre outros.

2.1.2. Características

Para que os recursos digitais disponibilizados na internet possam ser considerados OA e possam ser inseridos em um ambiente de aprendizagem devem apresentar diversas características, como: a reusabilidade, adaptabilidade, granularidade, acessibilidade, durabilidade e interoperabilidade. As seções a seguir apresentarão como são definidas essas características.

2.1.2.1. Reusabilidade

A Reusabilidade traz à tona um dos principais benefícios de objetos de aprendizagem, que é proporcionar um apoio ao design instrucional e ao processo de desenvolvimento, afirma Kim (2009, p. 32). Segundo Filatro (2003, p. 64 e 65) design instrucional é compreendido como:

“a ação institucional e sistemática de ensino, que envolve o planejamento, o desenvolvimento e a utilização de métodos, técnicas, atividades, materiais, eventos e produtos educacionais em situações didáticas específicas, a fim de

facilitar a aprendizagem humana a partir dos princípios de aprendizagem e instrução conhecidos”.

Um OA bem elaborado deve permitir que os usuários estabeleçam várias aplicações do mesmo sem dificuldades, ou seja, deve possibilitar que os usuários realizem tarefas sem complicações adicionais. A seguir (Figura 2) será apresentado um exemplo de reusabilidade.

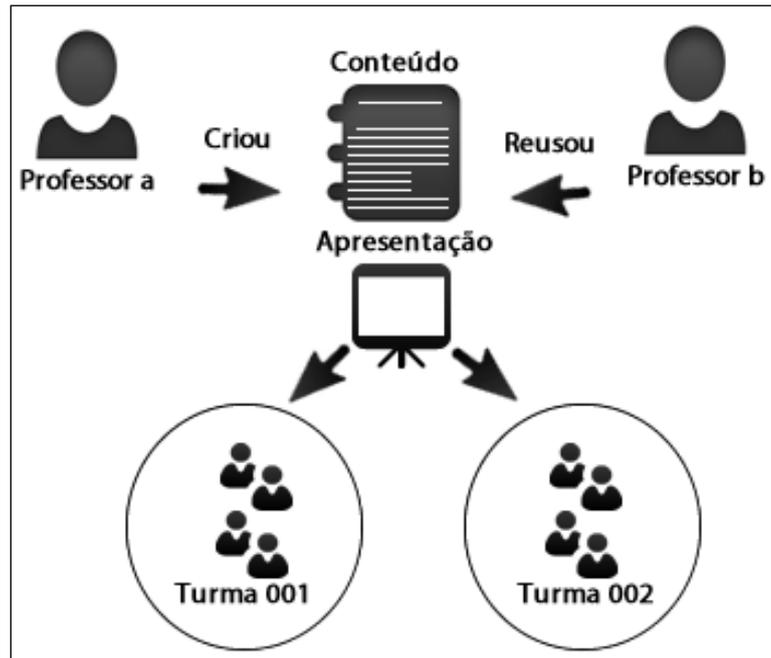


Figura 2 - Exemplo de Reusabilidade

Como pode ser observado na Figura 2, um professor de uma determinada disciplina (Professor A) criou um OA como conteúdo de estudo para a “Turma 001”, em seguida o Professor B procurava por um assunto cujo assunto era o mesmo que o Professor A tinha compartilhado em sua turma. O Professor B reutilizou o OA em sua turma (Turma 002), mudando a sua forma de apresentação.

A reusabilidade deve permitir que um OA seja criado de forma que possa ser aplicado em diferentes contextos, por exemplo, mudando apenas a camada de apresentação.

2.1.2.2. Adaptabilidade

As constantes evoluções das tecnologias põem em evidência a adaptabilidade, pois quando surgem novos ambientes e ferramentas tecnológicas, novas formas e práticas devem ser

desenvolvidas com as condições propostas por essas inovações. Diante disso, sistemas que tenham como característica a adaptabilidade inserem-se mais rapidamente nesses novos cenários, cuja mudança é uma constante. Assim Mendes (2004, online) afirma que um OA adequado deve ser “adaptável a qualquer ambiente de ensino”. A Figura 3 exemplifica esta característica.

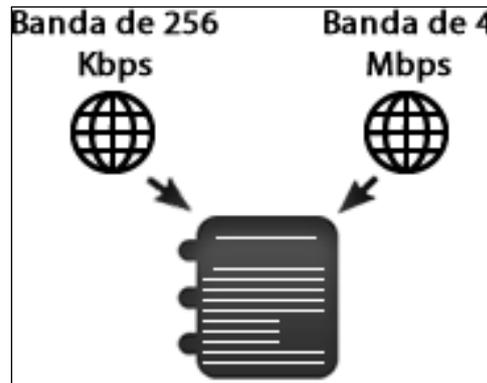


Figura 3 - Exemplo de Adaptabilidade

De acordo com a Figura 3, um exemplo de adaptabilidade é a apresentação de objetos/conteúdos independente da banda larga da instituição de ensino, seja ela de 8Mbps ou 256Kbps, pois a apresentação do conteúdo do OA deve se adaptar a banda, ou seja, se a banda for pequena deverá ser apresentado um *layout* mais simples sem perder informações ou a estrutura.

2.1.2.3. Granularidade

A granularidade tem sido um dos temas mais predominantes na aplicação de OA na educação. Vários pesquisadores tentam defini-la, porém não há um consenso a respeito do nível de granularidade ideal, pois ainda existe uma confusão sobre a quantidade de níveis existente dentro desta característica e o que cada nível realmente representa. A granularidade é a característica mais importante na criação, no uso e reuso de um OA.

Segundo Silveira (2007, p. 151 e 152) a granularidade dos objetos de aprendizagem refere-se ao “grau de detalhe ou de precisão contido em um OA, bem como seu tamanho, decomposição e potencial de reuso”. Ela poderá ter várias dimensões, como, por exemplo, espaço e tempo. A figura a seguir apresenta um gráfico referente à granularidade.

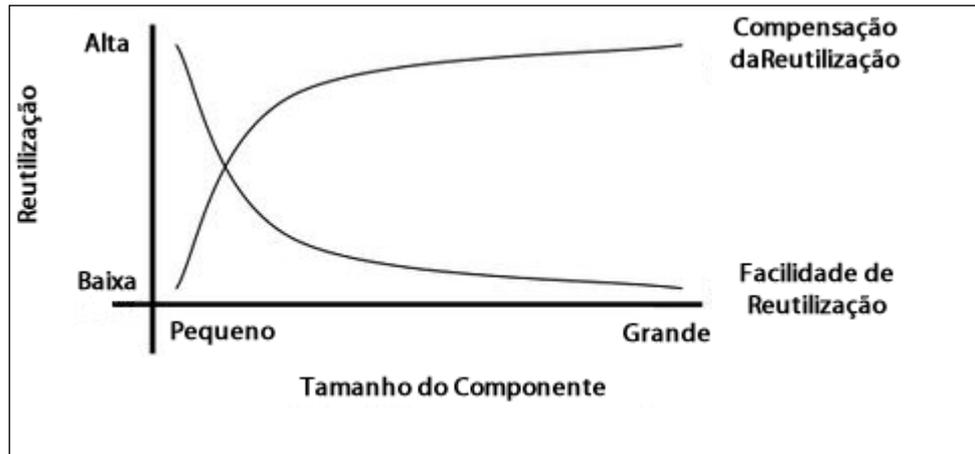


Figura 4 - Reutilização versus tamanho do componente (Adaptada PRIETO-DIÁZ, 1996, p. 5).

De acordo com a Figura 4 pode-se observar que, quanto menor o componente, maior a possibilidade de reutilização. Quanto maior o componente, menor a possibilidade de reutilização. Pode-se correlacionar esta explicação com o conceito de programação em módulo, ou seja, quando mais modulado for um programa, maior a possibilidade de reutilização em outros contextos. A imagem a seguir (Figura 5) exemplifica como funciona a granularidade.

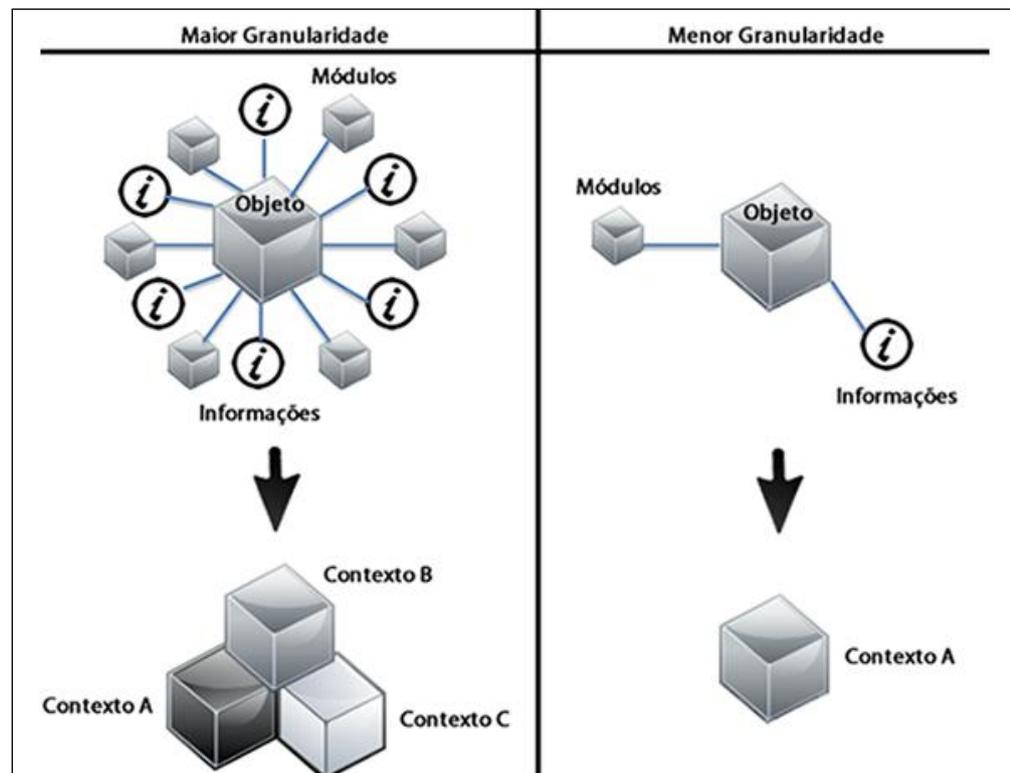


Figura 5 – Exemplo de Granularidade

Conforme demonstrado na Figura 5, um OA que contém um número maior de módulos pode ser inserido em mais contextos do que outro objeto com menos módulos, então um OA que possui mais módulo poderá ter uma maior granularidade. Granularidade é conhecida também pelo termo Modularidade, que por sua vez é inspirada na ideia de grãos materiais. Segundo Garcia (2006, p. 4) quanto maior a granularidade, maior será a flexibilidade na reutilização. Pode-se concluir que, quanto mais um OA pode ser utilizado em diferentes contextos, maior é sua granularidade.

Wiley (2000) considera a comunicação como base para agregar OA reutilizáveis em unidades cada vez maiores. Quando são agregadas informações aos objetos é iniciado um nível para este recurso (nível 1); quando há uma agregação dos recursos de nível 1, automaticamente é criado o recurso de nível 2; e por fim alguns dos recursos de nível 2 são agregados no curso completo que representará o tamanho maior. Esta visão pode ser observada na maioria das organizações que trabalha com OA, incluindo IMS Globo de Aprendizagem *Consortium*, a ADL e o Comitê de Padrões de Tecnologia de Aprendizagem (IEEE). Kim (2009, p. 25) realizou um estudo de caso com os níveis de granularidade definidos pelas organizações citadas e por outros autores. A partir deste estudo ele atribuiu cinco níveis para representar os níveis definidos por essas organizações. A imagem a seguir representa esses níveis.

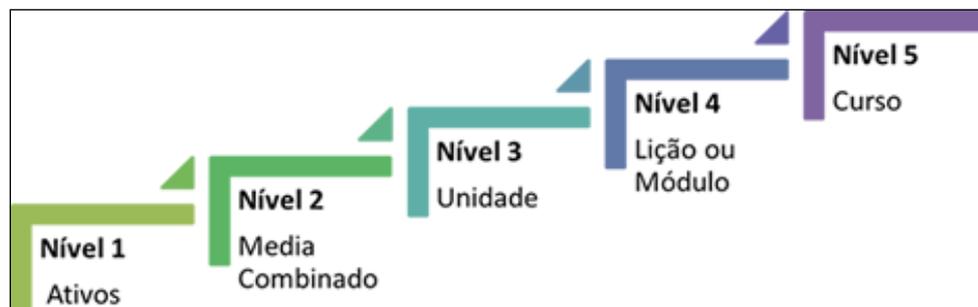


Figura 6 - Níveis de Granularidade

Como apresentado na Figura 6, os níveis de granularidade podem ser entendidos da seguinte forma:

- Nível 1 representa os “ativos”, única unidade discreta de informação, como texto, áudio, imagem etc.;

- Nível 2 - mídia combinado - consiste em um texto plano e uma combinação opcional de mídias (imagem, áudio, vídeo etc.);
- Nível 3 - unidade - consiste em vários meios de comunicação combinados, pode ter alguns componentes, como objetos de aprendizagem, conteúdos, prática e avaliação com a mídia;
- Nível 4 - lição ou módulo - consiste em diversas unidades (Nível 3);
- Nível 5 – curso - consiste em várias lições ou módulos.

A determinação do tamanho dos módulos básicos dos objetos de aprendizagem tem sido uma das questões mais importantes na concepção desses objetos. Segundo Wiley (2002), muitas tentativas têm sido feitas para definir o tamanho ideal da unidade do objeto de aprendizagem. Percebe-se que quanto maior a quantidade de módulos, maior é a complexidade para organizá-los, logo, com a ausência de uma organização adequada, o que poderia contribuir para a melhoria do processo de implementação pode se tornar algo extremamente complexo e de difícil manuseio.

2.1.2.4. Acessibilidade

Acessibilidade na web diz respeito à facilidade de acesso, por qualquer pessoa, independente das condições físicas, técnicas ou dispositivos, define Ferraz (2009, p. 18). Para que um Objeto de Aprendizagem se torne acessível, ele deve ser projetado de modo que atenda várias necessidades, dentre elas, por exemplo, as necessidades dos deficientes visuais, auditivos e motores. A imagem a seguir demonstra esta característica.

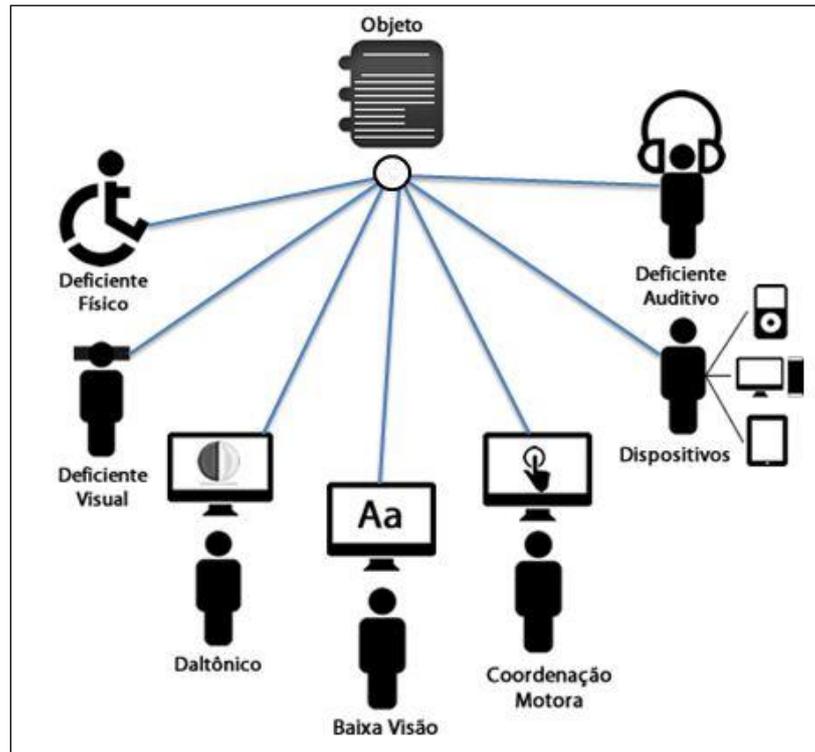


Figura 7 - Acessibilidade

Como apresentado na Figura 7, para que um objeto tenha esta característica, deve atender, por exemplo, as necessidades dos cegos (os OAs devem permitir que softwares especializados façam a leitura dele), daltônicos (os elementos da interface web devem ser ter contrastes bem definidos), pessoas com baixa visão (o OA deve permitir que softwares especializados ampliadores de tela ou de caracteres aumentam o tamanho da fonte e das imagens na tela do computador), deficientes auditivos (ter uma forma para que estes consigam acompanhar vídeos, por exemplo, legendas), deficientes de coordenação motora (permitir que softwares especializados faça o acionamento do sistema por meio de comandos falados através de um microfone).

2.1.2.5. Durabilidade

A característica durabilidade consiste na possibilidade de utilizar um Objeto de Aprendizagem sem ter que projetá-lo novamente ou recodificá-lo quando a base tecnológica for mudada. Para garantir uma maior segurança na mudança de plataforma/tecnologia é importante que a documentação do software esteja bem detalhada, clara e concisa, seguindo algum padrão de

desenvolvimento de software. É importante, também, que a equipe responsável pela mudança tenha um certo entrosamento, pois é necessário que a troca de informações entre estes membros esteja de acordo com a estratégia traçada pelo responsável.

Uma exemplificação desta característica é quando o Sistema onde estão armazenados os OAs passa por um processo de mudança de tecnologia ou plataforma [de desenvolvimento de] software. Deve ser mantida a integridade dos dados quando estes forem realocados em uma nova estrutura.

2.1.2.6. Interoperabilidade

Para serem reutilizáveis em diferentes contextos, os objetos de aprendizagem precisam do recurso de interoperabilidade, que representa a capacidade de ser funcional em sistemas diferentes, sem dificuldades técnicas, afirma Kim (2009, p. 33), ou seja, por exemplo, quando existe a transferência de um objeto de aprendizagem de um sistema de gerenciamento de conteúdo para outro, a integridade do objeto deve ser preservada. Desta forma, usar um modelo interoperável durante a construção de um objeto de aprendizagem é um importante fator a se considerar.

Como exemplos de tecnologia de suporte à aprendizagem podem-se citar os sistemas de treinamento baseado em computador, ambientes de aprendizagem interativos, sistemas de ensino a distância e ambientes de aprendizagem colaborativa. Exemplos de Objetos de Aprendizagem incluem conteúdos multimídia, conteúdos instrucionais, objetivos de aprendizagem, softwares instrucionais, ferramentas de software, pessoas e organizações (LOM, 2000, online).

De acordo com Wiley (2000, p.2) um dos principais objetivos do OA é que designers instrucionais possam construir objetos que, na maioria das vezes, sejam reutilizados em diferentes contextos. Além disso, os OA são entendidos como entidades digitais disponibilizadas na internet, ou seja, várias pessoas podem ter o acesso ao objeto simultaneamente. Agrega-se a isso o fato de que, com o aumento da incorporação de um dado objeto de aprendizagem, aumenta a possibilidade de um número maior de pessoas contribuírem para melhorar as novas versões.

Assim, no contexto apresentado, ter informações sobre os dados de cada objeto (metadados) é um passo fundamental para que os objetos de aprendizagem possam ser descritos corretamente. Pensando nesses e outros problemas o Comitê de Padrões de Tecnologia de Aprendizagem (*Learning technology standards committee - IEEE*) criou um

padrão de Metadados de Objetos de Aprendizagem (*Learning Object Metadata - LOM*). Esse padrão será descrito na seção seguinte.

2.1.3. LOM

Diante das características apresentadas, é importante ressaltar o uso de metadados para os objetos de aprendizagem, pois os metadados têm como função descrever e estruturar o objeto de maneira que o acesso, extração e compreensão dos dados que o compõem sejam facilitados. O LOM (Figura 8) é definido como “um elemento de dados para o qual o nome, a explicação, o tamanho, ordenação, espaço de valor e tipo de dados são definidos nesta norma” (IEEE, 2002, p. 5). A imagem a seguir (Figura 8) apresenta as categorias que formam o LOM.

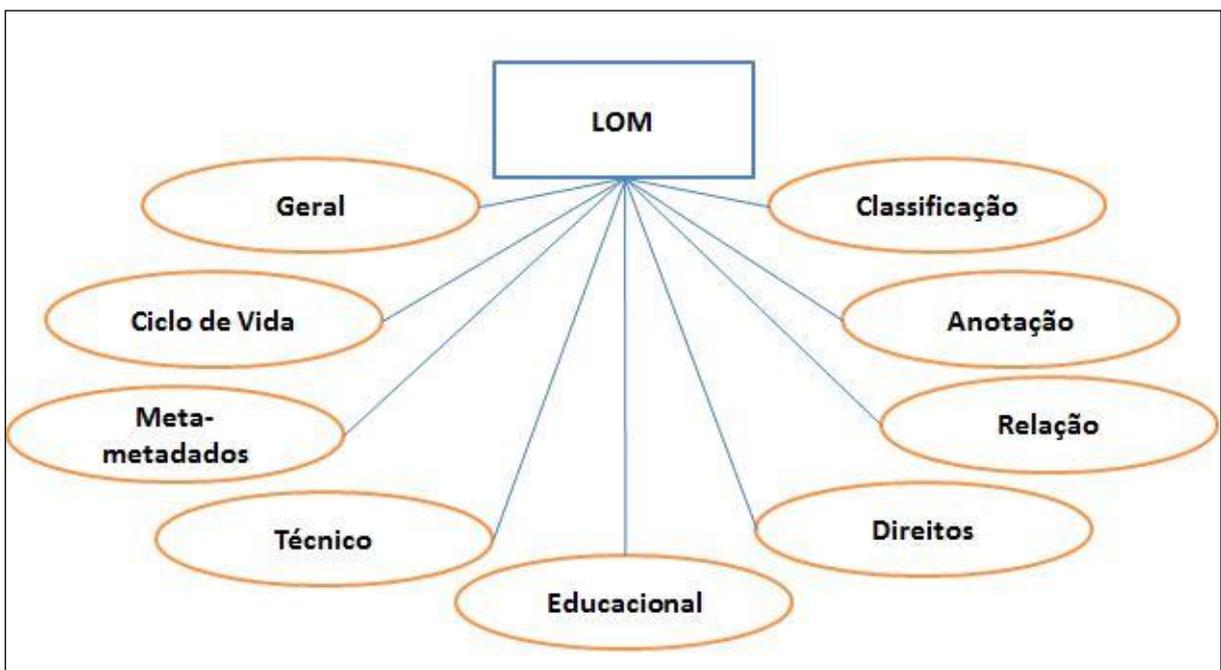


Figura 8 – Estrutura do esquema LOM v1.0

Na Figura 8 são apresentadas as categorias que compõem o LOM: Geral, Ciclo de Vida, Meta-metadados, Técnico, Educativo, Direitos, Relação, Anotação e Classificação. Cada categoria carrega consigo outros elementos, que vão desde valores (tipo, valor, data) até outros sub-elementos. A seguir serão apresentadas estas categorias de acordo com a estrutura definida pela IEEE (2002, p. 10-36).

2.1.3.1. Geral

Esta categoria tem como função agrupar as informações gerais que descrevem o Objeto de Aprendizagem. A Tabela 1 ilustra os elementos que compõem a categoria.

Tabela 1 - Categoria Geral - LOM

Geral
- Identificador
- Título
- Idioma
- Descrição
- Palavra-chave
- Cobertura
- Estrutura
- Nível de Agregação

Como pode ser observado na Tabela 1, os elementos desta categoria podem ser descritos da seguinte forma. O primeiro, **Identificador**, seria o rótulo exclusivo para identificar o Objeto de Aprendizagem. Este elemento apresenta dois valores, Catálogo, que informa o nome ou designação da identificação, por exemplo, URI (Identificador Uniforme de Recursos) e Entrada, que é o valor do identificador, no caso de usar o URI, um valor seria, por exemplo, <http://www.konnen.com/video?id=123456>, estes números finais (123456) identificam um determinado OA.

O segundo elemento, o **Título**, corresponde ao nome atribuído ao Objeto de Aprendizagem, por exemplo, “Modelo de maturidade CMMI e MPS-BR”. O próximo elemento, **Idioma**, refere-se ao idioma utilizado para se comunicar com o usuário.

O elemento **Descrição** consiste na descrição textual do conteúdo do OA. Esta descrição deve ser feita de forma apropriada para atingir o grau de entendimento do usuário sobre determinado assunto, por exemplo, se o OA de aprendizagem é voltado para um usuário que cursa os últimos períodos de um curso de Bacharelado, logo pode ser considerado que o seu grau de conhecimento sobre um assunto seja elevado, então pode-se utilizar termos técnicos na descrição do conteúdo do OA.

O quinto elemento desta categoria são as **Palavras-chave**, que podem ser usadas para formar a folksonomia (hierarquização/organização das informações feitas seguindo um

modelo pré-definido e controlado por especialistas da área (GOLDEN e HUNBERMAN, 2006, p. 198)) do sistema. A **Cobertura** é a região geográfica, o tipo de cultura, o período ou jurisdição do OA, por exemplo, tendo como título do OA “As inovações tecnológicas brasileiras do século XXI”, o assunto pode ser descrito da seguinte forma: Palavra-chave: “inovações tecnológicas” e Cobertura: “Século XXI Brasil”.

A **Estrutura** consiste na forma de organização do OA e é definida pelos seguintes valores:

- Atômica: um objeto que é indivisível, sem relacionamentos;
- Coleção: um conjunto de objetos que possuem alguma característica em comum, mas que não têm relação de hierarquia;
- Em rede: um conjunto de objetos interligados entre si;
- Hierárquica: Um conjunto de objetos cujos relacionamentos podem ser representados por uma estrutura de árvore;
- Linear: um conjunto de objetos ordenados.

E, por fim, o elemento **Nível de Agregação**, que mostra a granularidade funcional do OA. Este elemento é caracterizado por quatro níveis:

- Nível 1: O menor nível de agregação;
- Nível 2: Uma coleção de Objetos de Aprendizagem de Nível 1;
- Nível 3: Uma coleção de Objetos de Aprendizagem de Nível 2;
- Nível 4: O maior nível de granularidade, pode conter Objetos de Aprendizagem de Nível 3 ou 4 de agregação.

Levando em consideração os dois últimos elementos, Estrutura e Nível de Agregação, pode-se dizer que, por exemplo, se o objeto de aprendizagem for uma imagem de um diagrama de UML, a estrutura é considerada atômica e o nível de agregação é 1. A partir do momento que o Objeto de Aprendizagem é a imagem do diagrama UML e um texto com uma explicação sobre o tema, sua estrutura passa a ser de Coleção ou Em rede e o seu nível de agregação passa a ser 2. No caso da criação de um curso sobre UML, a estrutura passa a ser Linear, se os documentos forem vistos de forma Linear, ou seja, seguindo uma sequência direta (anterior/próximo) e o seu Nível de Agregação passa a ser 3. Se o Objeto de Aprendizagem for um conjunto de lições sobre os diagramas de UML a partir de diferentes fontes, a sua estrutura é uma Coleção e seu Nível de Agregação é 3. Por fim, se o Objeto de

Aprendizagem é um conjunto de cursos contendo a definição, descrição e interpretação dos diagramas de UML em sua totalidade (ou próximo a isso), a estrutura desse Objeto passa a ser Linear ou Hierárquica e o seu Nível de Agregação 4.

2.1.3.2. Ciclo de Vida

Esta categoria descreve as características relacionadas ao histórico e estado atual dos Objetos de Aprendizagem e todos os usuários que contribuíram para sua evolução. A Tabela 2 apresenta os elementos que fazem parte desta categoria.

Tabela 2 - Categoria Ciclo de Vida - LOM

Ciclo de Vida
- Versão
- Status
- Contribuição

Esta categoria é composta por três elementos, conforme pode ser visto na Tabela 2. O elemento **Versão** descreve a edição em que o Objeto de Aprendizagem se encontra, por exemplo, versão 1.0.beta. Já o elemento **Status** informa o status em que o Objeto se encontra, como possíveis valores: final, revisado, indisponível.

E, por último, o elemento **Contribuição**, que indica as entidades (pessoas, organizações) que contribuíram para a evolução do Objeto de Aprendizagem durante seu ciclo de vida, como forma de contribuição pode-se considerar um contexto amplo, ou seja, todas ações que afetam o estado do OA, por exemplo, a criação, edição e publicação. Para a descrição completa deste elemento é necessário informar o Papel que o responsável tem diante de um determinado OA, como, autor, editor, especialista no assunto, validadores educacionais etc. É importante também identificar as informações da Entidade responsável pela contribuição, seja ela uma organização ou pessoa, como também a Data da contribuição.

2.1.3.3. Meta-metadados

Esta categoria relata as informações referentes ao metadado utilizado para descrever um Objeto de Aprendizagem. Na Tabela 3 são listados os elementos que fazem parte desta categoria.

Tabela 3 - Categoria Meta-metadados - LOM

Meta-metadata
<ul style="list-style-type: none"> - Identificador - Contribuição - Esquema de Metadados - Linguagem

Conforme apresentado na Tabela 3, o **Identificador** é responsável por armazenar o rótulo exclusivo que identifica esse registro de metadados. Este elemento apresenta dois valores: Catálogo, que informa o nome ou designação da identificação, por exemplo, URI e Entrada, que é o valor do identificador, sendo um possível valor, por exemplo, <http://www.konnen.com/metadado/lom?id=123>.

O elemento **Contribuição** descreve as entidades (pessoas e organizações) responsáveis em contribuir com o metadado (criação, validação), semelhante ao elemento contribuição da seção anterior (2.1.3.2 Ciclo de Vida) que descreve as entidades responsáveis por contribuir com o Objeto de Aprendizagem.

Já o elemento **Esquema de Metadado** descreve o nome e a versão da autoridade de especificação usada para criar estes metadados, por exemplo, pode-se citar o LOM v1.0. E, por fim, a **Linguagem** utilizada para a especificação dos metadados.

2.1.3.4. Técnico

A categoria Técnico descreve os requisitos necessários e as características do Objeto de Aprendizagem. A tabela 4 apresenta os dados desta categoria.

Tabela 4 - Categoria Técnico - LOM

Técnico
<ul style="list-style-type: none"> - Formato - Tamanho - Localização - Requisitos - Procedimentos de Instalação - Outros Requisitos - Duração

De acordo com a Tabela 4, o elemento **Formato** é usado para descrever o tipo e o formato do Objeto de Aprendizagem. De acordo com a IEEE é importante que seja seguido o padrão definido pela RFC2048:1996, tipo MIME. Por exemplo, no caso do OA ser um vídeo, seu formato será da seguinte forma: video/avi.

O segundo elemento, **Tamanho**, é utilizado para descrever o tamanho real em *bytes* do OA. Já o elemento **Localização** é destinado para a descrição do lugar de armazenamento de um Objeto de Aprendizagem, neste caso é indicado que seja realizado por meio de uma URL ou URI, por exemplo, <http://localhost/sites/konnen/imagem/id=1234>.

Nesta categoria também encontra-se o elemento **Requisitos**, que indica a capacidade técnica necessária para a utilização de um determinado Objeto de Aprendizagem. Pode-se criar grupos de requisitos. Cada grupo é armazenado no sub-elemento Ou Composto, por exemplo, requisitos de software, hardware(dentro de hardware, outros grupos, como capacidade de armazenamento, processamento, entre outros). Para o sub-elemento Ou composto é interessante que sejam apresentadas as seguintes informações:

- Tipo: requisito necessário para a utilização do Objeto, por exemplo, hardware, software, entre outros.
- Nome: a tecnologia necessária para utilizar o OA (Levando em consideração que seja Navegador, por exemplo, teria como alternativa *Chrome* e *FireFox*)
- Versão Mínima: descreve a versão mínima da tecnologia.
- Versão Máxima: descreve a versão máxima da tecnologia.

No elemento **Procedimentos de Instalação** é feita uma descrição de como instalar e/ou utilizar AO como, por exemplo, “descompactar um arquivo zip, acessar uma determinada

pasta que contém o arquivo *iniciar.html*". Já o elemento **Outro Requisitos** contém informações sobre outros softwares e requisitos de hardware como, por exemplo, placa de som, tempo de execução, entre outros. E, por fim, a **Duração**, que descreve o tempo de execução dos OA como vídeos e áudios.

2.1.3.5. Educacional

A categoria Educacional descreve as características pedagógicas do Objeto de Aprendizagem. A tabela a seguir contém o grupo de elementos que forma esta categoria.

Tabela 5 - Categoria Educacional - LOM

Educacional
<ul style="list-style-type: none"> - Tipo de Interatividade - Tipo de Recurso de Aprendizagem - Nível de Interatividade - Densidade Semântica - Papel Destinado - Contexto - Idade - Dificuldade - Tempo de Aprendizado - Descrição - Idioma

Como pode ser observado na Tabela 5, o primeiro elemento desta categoria é **Tipo de Interatividade**, que é definido por um dos três tipos de recursos: ativo, expositivo e misto. O recurso é determinado no momento em que o usuário interage com o OA, conforme pode ser observado a seguir:

- Ativo: induz o usuário a uma ação produtiva ou de decisão, por exemplo, faz com que ele realize simulados, questionários e exercícios;
- Expositivo: induz o usuário a uma ação de observação sobre um OA, ou seja, ocorre quando o usuário deve absorver o conteúdo apresentado a ele, na maioria das vezes esses conteúdos são áudios, vídeos, apresentações etc;

- Misto: quando um Objeto de Aprendizagem utiliza os dois recursos, Ativo e Expositivo.

No elemento **Tipo de Recurso de Aprendizagem** é especificada a metodologia utilizada pelo OA. Um mesmo recurso pode apresentar mais de uma metodologia, por exemplo, *slide*, tabela, imagem, questionário, entre outros.

O **Nível de Interatividade** descreve o grau em que o aluno pode influenciar o aspecto ou comportamento de um OA. Este elemento pode ser classificado em cinco níveis: muito baixo, baixo, médio, alto e muito alto. Já o elemento **Densidade Semântica** armazena o grau de concisão de um Objeto de Aprendizagem. A densidade semântica é independente da dificuldade de aprendizado. Na maioria das vezes, documentos concisos podem ter uma alta densidade semântica se agregarem muita informação. Podem ser atribuídos cinco níveis para esse elemento: muito baixo, baixo, médio, alto e muito alto.

O quinto elemento, o **Papel Destinado** informa o papel do usuário no qual o Objeto de Aprendizagem foi projetado, neste caso tem-se estes quatro possíveis papéis:

- Professor: responsável em apresentar os conceitos e mostrar um OA para o aprendiz, por exemplo;
- Autor: cria ou publica um Objeto de Aprendizagem;
- Aprendiz: o aluno que acessa o OA a fim de aprender algum assunto;
- Gerente: administra o compartilhamento do OA, por exemplo, um órgão ou universidade.

O próximo elemento é o **Contexto**, que descreve o universo do Objeto de Aprendizagem. São sugeridos alguns contextos, como: escola, ensino superior, treinamento, entre outros. Já a **Idade** é destinada a armazenar a faixa etária à qual o Objeto de Aprendizagem é destinado. Também existe o elemento **Dificuldade**, que informa o quão difícil é trabalhar com um determinado OA para o destino pretendido, no caso a faixa etária. Este dificuldade pode ser classificada em cinco níveis: muito fácil, fácil, médio, difícil, muito difícil.

O elemento **Tempo de Aprendizado** é destinado a descrever o tempo aproximado necessário para compreender e trabalhar algum conceito, atividade de um OA com um público alvo. O elemento **Descrição** conterà comentários sobre a forma como os Objetos de

Aprendizagem deverão ser usados. E, por fim, o **Idioma** usado pelo típico usuário alvo do OA.

2.1.3.6. Direito

Esta categoria descreve os direitos de propriedades intelectuais e condições de uso de um Objeto de Aprendizagem. O grupo de elementos a seguir (Tabela 6) apresenta esta categoria.

Tabela 6 - Categoria Direito - LOM

Direito
<ul style="list-style-type: none"> - Custo - Copyright e Outras Restrições - Descrição

Como apresentado na Tabela 6, o elemento **Custo** indica o valor referente à utilização do OA (o custo pode ser zero). Os elementos **Copyright** (licenças para softwares livres que poderão ser usadas, como a GPL – Licença Pública Geral; MIT – Licença criada pelo Massachusetts Institute of Technology; Creative Commons – várias licenças de direitos autorais) e **Outras Restrições** apresentam as questões de direitos autorais e restrições na manipulação/alteração de um OA. A **Descrição** é formada pelas informações sobre o termo de utilização do Objeto de Aprendizagem, por exemplo, a informação “só é permitido o uso do OA após a realização do cadastro no sistema”.

2.1.3.7. Relação

Esta categoria define a relação entre Objetos de Aprendizagem. Caso existam múltiplos relacionamentos, devem-se criar instâncias do relacionamento mais de uma vez, respeitando o limite de no máximo cem instâncias definido pelo padrão LOM (IEEE, 2002, p. 31). A tabela a seguir mostra os elementos que compõem esta categoria:

Tabela 7 - Categoria Relação - LOM

Relação
<ul style="list-style-type: none"> - Tipo - Recurso

De acordo com a Tabela 7, o elemento **Tipo** armazena a natureza da relação entre os Objetos de Aprendizagem. Logo, é possível criar uma sequência de estudo sobre um determinado assunto, enriquecendo ainda mais o aprendizado do usuário. Depois de determinar o tipo é importante estabelecer o **Recurso** que será utilizado. Esse elemento é formado pelos atributos Catálogo, Entrada e Descrição. Estes atributos armazenam os mesmos valores das demais categorias, mudando apenas a Descrição que, nesse caso, apresenta informações sobre o Objeto de Aprendizagem alvo.

2.1.3.8. Anotação

Esta categoria fornece comentários sobre o uso educacional do Objeto de Aprendizagem, como também informações a respeito de quando e por quem os comentários foram criados. Permite, ainda, que sejam compartilhados os comentários, as sugestões e avaliações de OA entre os educadores, o que possibilita a interação e a colaboração entre eles. A seguir é apresentado o grupo dos elementos desta categoria.

Tabela 8 - Categoria Anotação - LOM

Anotação
- Entidade
- Data
- Descrição

O primeiro elemento desta categoria é a **Entidade**, que armazena a pessoa ou organização que criou a anotação, seguida da **Data** na qual ela foi criada e da sua **Descrição**.

2.1.3.9. Classificação

Esta categoria descreve o contexto em que o Objeto de Aprendizagem será classificado. Para a definição de múltiplas classificações, podem ser criadas várias instâncias de uma categoria. A seguir (Tabela 9) são apresentados os elementos que compõem essa categoria.

Tabela 9 - Categoria Classificação - LOM

Classificação
- Finalidade

- Taxonomia
- Descrição
- Palavras-chave

Como pode ser observado na Tabela 9, o primeiro elemento desta categoria é a **Finalidade**, que descreve o propósito da classificação de um Objeto de Aprendizagem. Tem como valores padrão: disciplina, ideia, requisito prévio, objetivo educacional, restrições, entre outros.

O elemento **Taxonomia** armazena os níveis da classificação, onde cada nível é um aprimoramento na definição de um nível anterior. Podem existir caminhos distintos nas classificações que descrevem uma mesma característica. Existem dois sub-elementos que compõem a Taxonomia: a Fonte, que representa o nome do sistema de classificação e o Táxon, que indica um termo particular dentro de uma taxonomia. O Táxon guarda também o Id, que é o seu identificador, e a Entrada, que é o rótulo textual do Táxon. Tem, ainda, a **Descrição** e as **Palavras-chave** do Objeto de Aprendizagem.

2.2. Ontologia

Em função do advento da internet e da popularização dos computadores o ambiente informacional de hoje é mais amplo do que há alguns anos. Com a grande quantidade de informações disponíveis no meio digital, é recomendada a utilização das ontologias com o propósito de organizar informações em um domínio do conhecimento.

De acordo com Staab e Stuber (2004, apud DAVIES, 2006, p. 4) Ontologias consistem em:

“conceitos (classes), relações (propriedades), instâncias e axiomas. Assim, de uma forma sucinta, uma ontologia pode ser considerada uma 4-tupla $\langle C, R, I, A \rangle$, onde C é um conjunto de conceitos, R um conjunto de relações, I um conjunto de instâncias e A um conjunto de axiomas.”

Aplicando este conceito em um domínio específico, por exemplo, material didático de uma rede social acadêmica, tem-se o seguinte modelo:

- **C** - o conjunto de Conceitos, que seriam as classes (Curso, Disciplina, Turma, Plano de Ensino, Programa, Tópico);

- **R** - o conjunto de Relações existente entre as classes (Curso \supset Disciplina, Disciplina \supset Turma, Turma \supset Plano de Ensino, Plano de Ensino \supset Programa e Programa \supset Tópico);
- **I** - o conjunto de instâncias, como exemplo, um Material Didático (vídeo);
- **A** - o conjunto de Axiomas.

Logo essa 4-tupla forma a Ontologia do domínio e, nesse contexto, os axiomas são as verdades estabelecidas. Ao considerar como contexto o material didático de uma rede social acadêmica, têm-se definições como interdisciplinaridade e transversalidade. Assim, é necessário criar meios computacionais capazes de representar essas definições, de forma que quando um conjunto de ações for realizada, o sistema possa inferi-las. A Figura 9 define um exemplo de axioma.

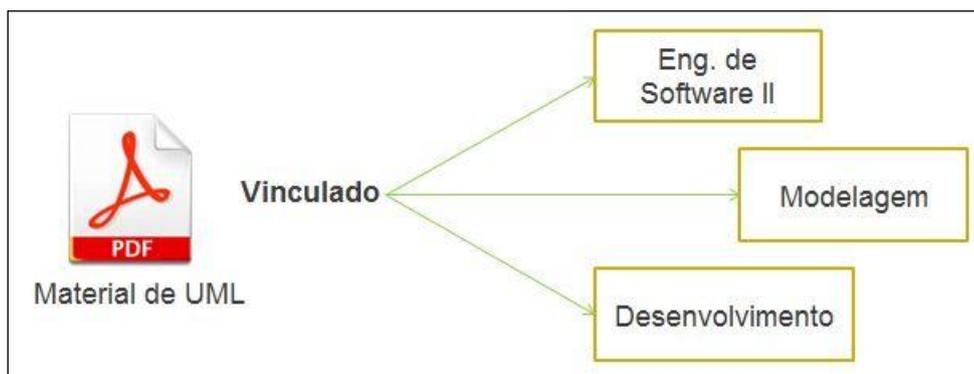


Figura 9 - Exemplo de Axioma aplicado na Rede Social Acadêmica

De acordo com a Figura 9, no momento em que um material é vinculado a mais de uma disciplina, é inferida a relação de “interdisciplinaridade”, pois na ontologia, há o seguinte axioma que define essa relação: todo objeto relacionado a mais de uma disciplina é equivalente a um objeto interdisciplinar. Assim, ao serem definidos os axiomas, grupos de domínio podem ser construídos e analisados pelo sistema.

Noy e McGuinness (2002, p.1) apresentam algumas razões para o desenvolvimento de uma ontologia:

- Compartilhar o entendimento da estrutura de informações entre pessoas ou agentes de software: supondo que existam vários sites de vendas de produtos acadêmicos e uma rede social acadêmica e que todos esses sistemas utilizam ontologias aplicadas em

seus domínios. A partir do momento que houver um mecanismo de inferência que equipare os termos com significado semelhante, mas sintaxe distinta nessas diversas ontologias, os agentes de software poderão extrair e agregar mais informações, assim como tornar o processo de recomendação mais eficiente. Desta forma, tendo como base essas informações, os agentes poderão responder consultas de usuários ou até mesmo usá-las como dados de entrada para outras aplicações.

- Permitir a reutilização do conhecimento do domínio: por exemplo, supondo que foi implantada uma rede social acadêmica em duas instituições de ensino, X e Y, e foi desenvolvida uma ontologia para a rede da instituição X. A instituição Y achou interessante a utilização da ontologia aplicada ao domínio de X, e resolveu reutilizar a mesma ontologia da rede social em sua instituição.
- Fazer inferências explícitas do domínio: assim, se houver a necessidade de mudar seus pressupostos, há um maior entendimento dos conceitos envolvidos e das suas relações.
- Separar o conhecimento do domínio do conhecimento operacional: por exemplo, definiu-se uma ontologia para uma rede social acadêmica, onde existem situações como curso, turma, professor, aluno (conhecimento do domínio). Em seguida surgiu a oportunidade de desenvolver uma ontologia para uma rede social de apreciadores de músicas alternativas. Logo, usa-se a mesma tecnologia (algoritmos, plataformas e afins) que serviu como base para o desenvolvimento da outra ontologia (conhecimento operacional) e muda-se o domínio (passa a ser banda, cantor, música, gênero musical).
- Analisar o conhecimento do domínio: um exemplo para que seja possível realizar esta análise é a existência de uma especificação declarativa dos termos. A partir desta especificação pode-se verificar se é possível entender as ontologias existentes e/ou reutilizá-las.

A relação entre essas razões e o conceito de ontologia apresentado indica, por exemplo, que ao definir metadados que permitem a associação entre objetos de um domínio, é possível estabelecer as relações que esses objetos possuem com as classes e propriedades de um dado domínio, assim como a inserção de relações entre esses metadados podem fazer com que haja a inferência de algumas verdades estabelecidas no contexto.

Segundo Noy e McGuinness (2002, p. 3 e 4), o desenvolvimento de uma ontologia resume-se em:

- definir as Classes;
- organizar as Classes em uma Taxonomia;

- definir os *slots* (propriedade de cada conceito/classe) e descrever seus valores permitidos;
- preencher os valores para os *slots* e para as instâncias.

A partir disso, pode-se criar uma base de conhecimentos através da definição de instâncias individuais dessas classes, preencher as informações específicas e restrições dos *slots* adicionais. A figura a seguir mostra esta aplicação no domínio do vinho.

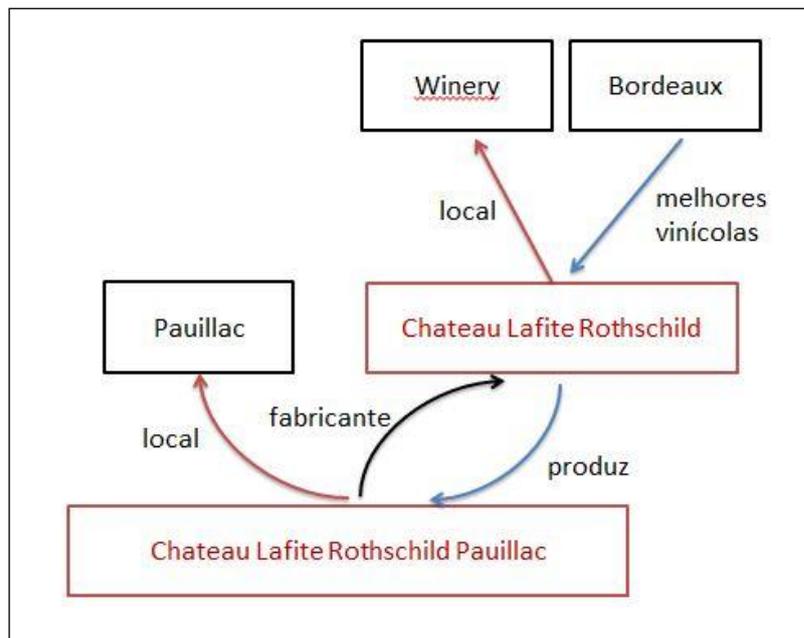


Figura 10 - Exemplo de Classes, Relações e Instâncias no Domínio “Vinho” (NOY e MCGUINNESS, 2002, p. 4).

A Figura 10 apresenta algumas classes, instâncias e relações existentes no domínio “Vinho”, onde os quadros pretos e vermelhos representam as classes de instâncias e os *links* diretos apontam ligações internas (relações).

No seguinte domínio, material didático de uma rede social acadêmica, o desenvolvimento da ontologia pode ser estruturado da seguinte forma.

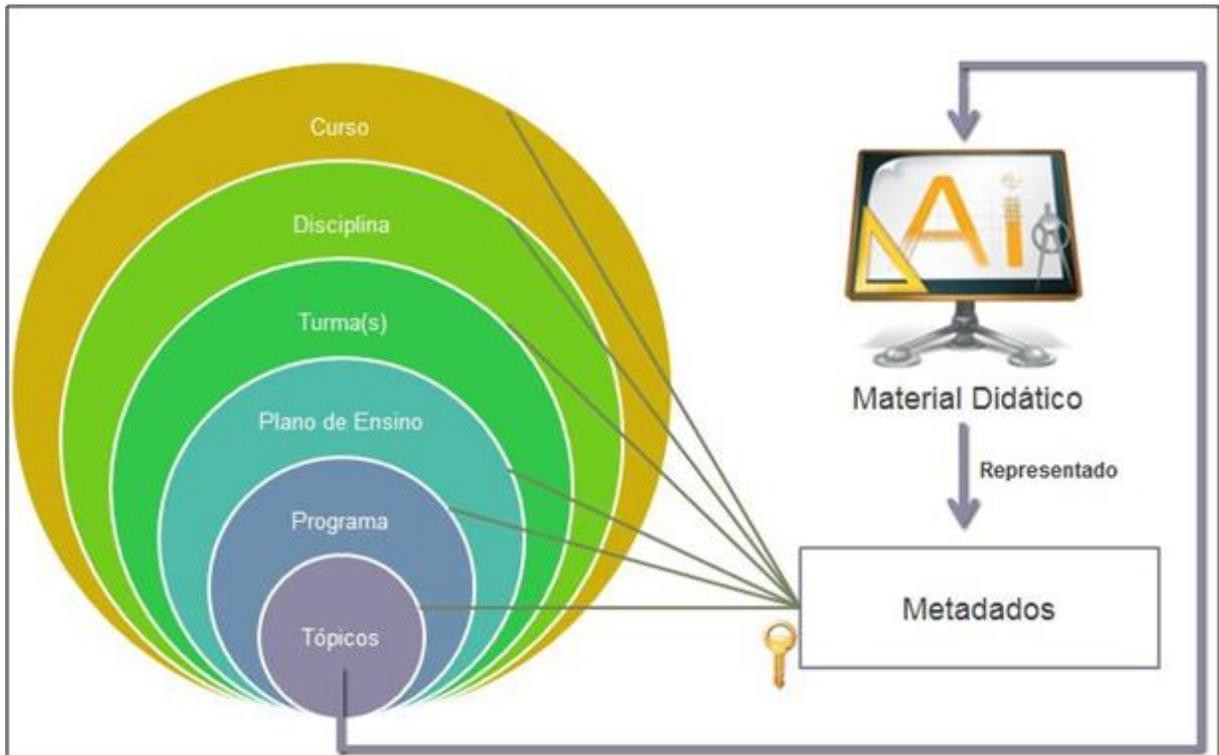


Figura 11 - Ontologia para o domínio material didático de uma rede social acadêmica.

Após a definição das classes (Curso, Disciplina, Turma, Plano de Ensino etc.), faz-se a organização em uma taxonomia, onde $\ll \text{Curso} \supset \text{Disciplina}, \text{Disciplina} \supset \text{Turma}, \text{Turma} \supset \text{Plano de Ensino}, \text{Plano de Ensino} \supset \text{Programa} \text{ e } \text{Programa} \supset \text{Tópico} \gg$. A próxima etapa é a definição dos *slots* e a descrição dos valores permitidos (tipos de valores, cardinalidade). Por exemplo, a classe Curso pode ter os seguintes *slots*: área, nível de aprovação, nível de formandos, quantidade de alunos. Seguindo a exemplificação, no *slot* área da classe Curso é atribuído o valor da área à qual o curso pertence. O *slot* material, da classe Disciplina, pode assumir o valor “um” ou “mais de um” se a Disciplina for ministrada por um ou por mais de um professor. Por fim, são preenchidos os valores para os *slots* e para as instâncias. Por exemplo, a instância material didático representa um tipo de material da classe Disciplina. Esta instância tem os seguintes valores de *slots* definidos:

- Nome: O Futuro dos Preços
- Disciplina: Gestão Tecnológica II
- Tipo: arquivo de texto
- Autor: Chris Anderson

2.2.1. Características

Para o desenvolvimento de uma Ontologia é necessário considerar algumas características, como: clareza, coerência, extensibilidade, codificação independente, entre outras. Segundo Gruber (1993, p. 2 e 3) estas características podem ser descritas da seguinte forma:

- **Clareza** – Uma ontologia deve repassar o significado pretendido dos termos definidos. As definições devem ser objetivas e devem ser independentes do contexto social ou computacional. É importante que todas as definições sejam documentadas em linguagem natural;
- **Coerência** – Uma ontologia deve ser coerente, ou seja, elas devem garantir que as inferências sejam consistentes de acordo com as definições. Por exemplo, se uma sentença pode ser inferida a partir de axiomas contraditórios, então esta Ontologia está incoerente;
- **Extensibilidade** – Uma ontologia deve ser projetada de modo que seja possível adicionar novos termos e definições, sem que tenha a necessidade de refazê-la ou modificá-la completamente;
- **Codificação independente** – A conceituação deve ser especificada sem depender de uma codificação. A importância de minimizar a codificação é que o compartilhamento de agentes de conhecimento podem ser implementados em sistemas diferentes, então, quanto menor for a dependência de código, mais fácil será a reutilização da ontologia em outros contextos;
- **Menor número de afirmações na ontologia** – Uma ontologia deve conceber o mínimo de afirmações e definições específicas. Para que seja suportado o compartilhamento e comunicação consistente do conhecimento.

Ontologias são normalmente classificadas de acordo com a generalidade do processo de conceituação, da sua abrangência e finalidade (DAVIES, 2006, p. 181):

- **De nível superior:** são as ontologias que representam um modelo geral do mundo, adequado para uma grande variedade de tarefas, domínios e áreas de aplicação;
- **Ontologias de domínio:** representam uma conceituação de um domínio específico, por exemplo, departamento pessoal ou biblioteca de uma instituição de ensino;

- **Ontologias de aplicação e tarefa:** são adequadas para intervalos específicos de aplicações e tarefas, ou seja, fornecem um vocabulário sistematizado de termos relacionados à execução de uma tarefa específica, independente do domínio, como exemplo, tem-se o KM Proton¹.

Segundo Davies (2006, p. 52 - 60), a Ontologia passa por um processo de evolução durante o seu desenvolvimento, o qual é definido por uma série de mudanças em cada uma das suas etapas, desde a etapa de captura dos requisitos, passando pela representação e implementação, até a etapa de validação, quando o processo é iniciado novamente. A Figura 12 apresenta o funcionamento do processo de evolução da ontologia.



Figura 12 - Processo de Evolução da Ontologia (Adaptada STOJANOVIC, 2004, p. 76).

De acordo com Davies (2006, p. 52 - 60) o processo representado pela Figura 12 funciona da seguinte forma:

- **Mudança de Captura** – O processo de evolução da ontologia é iniciado pela mudança de captura de requisitos explícitos (são gerados pelos Engenheiros de Ontologia que fazem adaptações das novas exigências ou pelos utilizadores finais que fornecem uma resposta explícita sobre a usabilidade da ontologia) ou por meio dos requisitos implícitos (é realizada através da análise do comportamento do sistema). As mudanças resultantes de tais requisitos são chamadas de *top-down* (requisitos explícitos) e *bottom-up* (requisitos implícitos);
- **Mudança de Representação** – Esta mudança inicia o componente central da Figura 12. É necessário que sejam identificadas e representadas essas mudanças em um formato adequado, ou seja, esta representação precisa ser definida para um modelo de ontologia específica. Esta mudança pode ser representada em vários níveis de granularidade, por exemplo, mudanças elementares (como a adição ou remoção de algum elemento da ontologia – um conceito, uma propriedade) compostas (como

¹ Mais informações sobre o projeto podem ser obtidas no site <http://proton.semanticweb.org/>

mover um conceito X para um outro ponto na hierarquia de conceitos) e complexa (é uma mudança de ontologia feita por combinações de pelo menos duas mudanças elementares ou compostas);

- **Mudança de Semântica** – Esta etapa do processo de evolução da ontologia descreve a mudança de semântica considerando a consistência estrutural (garante que a ontologia obedeça as restrições no que tange à forma de como são utilizadas as construções de linguagem da ontologia), consistência lógica (diz respeito à semântica formal da ontologia, ou seja, a visualização da ontologia como uma teoria lógica) e consistência definida pelo usuário (deve seguir as condições definidas pelo usuário para que se torne consistente; estas definições de consistência não são capturadas pela linguagem da ontologia em si);
- **Mudança de Propagação** – Esta fase tem como objetivo assegurar a consistência de artefatos dependentes (ontologias dependentes, instâncias e programas de aplicação usando a ontologia) após uma atualização da ontologia;
- **Mudança de Implementação** – O papel da fase de mudança de implementação é dividido em três etapas: (I) informar o engenheiro de ontologia sobre solicitações de alteração (é enviada uma lista de todas as implicações a serem realizadas, onde o engenheiro decide se aceita ou cancela as solicitações), (II) aplicar todas as mudanças (para aplicar uma mudança é necessário analisar se a mudança tem propriedades transacionais, ou seja, se garante atomicidade, consistência, isolamento e durabilidade) e, por fim, (III) acompanhamento das alterações realizadas (são guardadas em um registro de mudança, para que se possa acompanhar através dos *logs* de evolução);
- **Mudança de Validação** – A tarefa da mudança de validação é recuperar situações como: (I) o Engenheiro de Ontologias pode não compreender os efeitos reais que uma mudança pode causar e aprovar uma alteração que não poderia ser realizada; (II) quando existem diversos Engenheiros trabalhando em uma mesma ontologia de forma colaborativa, pode ocorrer de eles terem ideias diferenciadas sobre como a ontologia deve ser alterada. De acordo com estas situações apresentadas, esta mudança deve permitir justificar as alterações realizadas ou desfazê-las.

2.3. Trabalhos Correlatos

Com o desenvolvimento das Tecnologias de Informação e Comunicação, um novo cenário vem sendo explorado no meio educacional, a educação baseada na *web*. Torna-se cada vez mais comum o uso de Conteúdos Educacionais *online* para auxiliar o processo de ensino-aprendizagem. Utilizando apenas o padrão de descrição de Objeto de Aprendizagem, o LOM, apresentado na seção anterior (6.1.3, LOM - *Learning Object Metadata*), não será suficiente para que os OA apresentem todas as características de forma automatizada e eficaz, como reusabilidade, granularidade, entre outras. Diante desta situação, faz-se necessário o uso de Ontologias para os Objetos de Aprendizagem, que aumentarão, por exemplo, a capacidade de compartilhar, reutilizar e recuperar os OA em diversos contextos.

Para que a Ontologia seja estabelecida no domínio, será levada em consideração a ideia do conceito apresentado anteriormente, a 4-tupla $\langle C, R, I, A \rangle$. As seções a seguir apresentarão dois trabalhos relacionados à criação da ontologia para objetos de aprendizagem.

2.3.1. Ontologia para Objetos de Aprendizagem (OntoLo)

O termo “OntoLo” foi utilizado pelos autores Wang, Fang e Fan (2008) para definir uma Ontologia para Objetos de Aprendizagem. O principal objetivo do OntoLo é propiciar aos Objetos de Aprendizagem uma melhoria na característica de reusabilidade, de forma a facilitar o processo de compartilhamento. Para isso eles pensaram na combinação de Metadados de OA e Ontologia com o objetivo de melhor reutilizar e compartilhar os Objetos de Aprendizagem. O trabalho desenvolvido por eles fornece uma Ontologia de Objetos de Aprendizagem desenvolvida com o auxílio da ferramenta Protégé (*software* utilizado para modelagem de ontologias), conforme pode ser observado na descrição de alguns elementos a seguir.

Classes

Segundo Wang, Fang e Fan (2008, p. 473) existem diversas abordagens no desenvolvimento de uma hierarquia de classes, por exemplo, *top-down* (de cima para baixo), *bottom-up* (de baixo para cima) e combinações de processos de desenvolvimento. Foram criadas três classificações básicas para as abordagens citadas acima, que são: Objetos de Aprendizagem,

Objetivos de Aprendizagem e Categoria. A Figura a seguir representa um exemplo destas classificações.

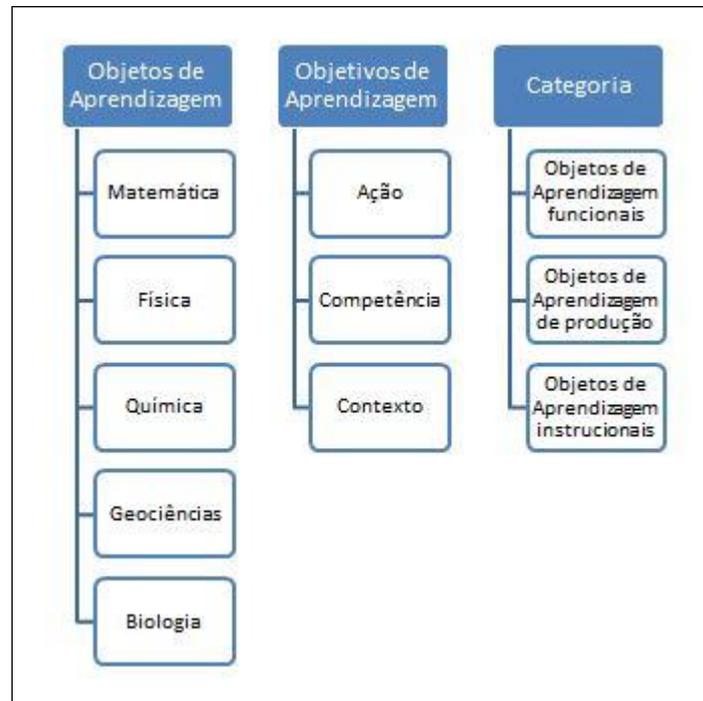


Figura 13 - Hierarquia de classes de ontologias (Adaptada de WANG, FANG e FAN, 2008, p. 473).

De acordo com a Figura 13, a classe Objeto de Aprendizagem é usada para descrever os assuntos do domínio que será representado (Matemática, Física, Química, Geociências e Biologia). Já a classe Objetivos de Aprendizagem descreve os diferentes níveis de aprendizagem, definidos nesse trabalho como: competência, ação e contexto. E, por fim, a classe Categoria, que abrange diversos tipos de recursos, dependendo da estrutura, formato etc. Os itens abordados na classe Categoria, segundo Wang, Fang e Fan (2008, p. 474) têm relação com a Hierarquia de Objetos de Aprendizagem apresentada por Wiley (2000, p. 77). A Tabela 10 apresenta essa hierarquia.

Tabela 10 - Hierarquia de objetos de aprendizagem – Adaptada de Wiley (2000, p. 77)

Taxonomia	Explicação	Exemplo
Fundamental	Único recurso digital.	Uma imagem JPEG
Combinado fechado	Combinação de elementos fundamentais que não podem ser restituídos.	Um vídeo

Combinado aberto	Combinação de recursos digitais abundantes que podem ser diretamente adquiridos e restituídos.	A web
Apresentação Generativa	Baixo nível de raciocínio lógico e da estrutura de objetos de aprendizagem.	Um aplicativo JAVA
Instrução Generativa	Raciocínio lógico e estrutura interativa de diferentes objetos de aprendizagem	Uma instrução EXECUTE transação de shell.

Esta hierarquia também pode ser entendida da seguinte forma: começaria do nível fundamental, por exemplo, uma imagem de uma mão formando um acorde em um teclado.

A imagem a seguir apresenta uma contextualização do nível fundamental.



Figura 14 - Formação de Acorde (Nível Fundamental)

Após o entendimento do nível fundamental (Figura 14), o próximo nível, combinado fechado, leva em consideração que o objetivo de montar os acordes foi alcançado e que próximo passo é aprender o ritmo da música, assim é apresentado um vídeo para que sejam acompanhados os acordes de uma música de fundo, conforme apresentado na Figura 15.



Figura 15 - Vídeo aula sobre ritmo de música - Nível combinado fechado (YOUTUBE, 2008, online, mim 0:15).

O nível combinado aberto pode ser exemplificado da seguinte forma, uma página web que mostra a letra de uma música junto aos acordes e uma vídeo-aula integrada à página ensinando a tocar a música apresentada. A Figura 16 apresenta esta exemplificação.

Figura 16 - Letra de música + Acordes + Vídeo Aula integrada - Nível Combinado Aberto (CIFRA CLUB, 2011, online).

O próximo nível seria a Apresentação Generativa. Uma exemplificação seguindo o contexto anterior teria, por exemplo, um sistema capaz de identificar o posicionamento inadequado na formação dos acordes no teclado. Já no nível mais avançado (Instrução Generativa), será considerado o aprendizado do usuário, assim o mesmo teria acesso a um aplicativo que permitiria testar os conhecimentos adquiridos.

Relacionamentos

De acordo com Wang, Fang e Fan (2008, p. 474) existem dois tipos de relações básicas: aquelas que são comuns a qualquer ontologia, como *Is-a*, *Kind-of*, *Instance-of*, e as relações criadas para atender o propósito educacional de uma ontologia baseada em um domínio, por exemplo, *Composed-of* e *Target-of*.

Propriedades

Os autores Wang, Fang e Fan (2008, p. 474 e 475) definem que, as propriedades (mencionadas através do termo *slot*, devido à utilização do Protegé) são baseadas no padrão CELTS-42 (*Metadata Application Specification of Basic Education Resource*) (YONG-HE, 2009, p. 449). Segundo Wang, Fang e Fan (2008, p. 475), nota-se que todas as subclasses de uma classe herdam os *slots* de sua superclasse, mas as subclasses podem ter propriedades próprias, pois os *slots* podem ter diferentes facetas descrevendo tipo, valores permitidos e cardinalidade.

Instâncias

As instâncias correspondem ao preenchimento dos *slots*. Ou seja, ao considerar uma instância individual *TesteLiteraturaModerna001*, que representa o tipo específico *Literatura_Moderna*, subclasse de *Literatura*, que, por sua vez, é subclasse de *Objetos de Aprendizagem*, tem-se a estrutura apresentada na Tabela 11.

Tabela 11 - Uma instância e parte de suas propriedades – Adaptada de Wang (2008, p. 474)

TesteLiteraturaModerna001	
Dificuldade:	2
Criador:	João da Silva
Assunto:	Teste sobre Literatura Moderna
Formato	Comentários
Objetivo:	Educação Fundamental
	Definição
	Descrição

Título:	Unidade 1, Teste 1
Tamanho:	2049 bytes
Data:	07/07/2011
Descrição:	Teste para unidade 1 do Livro 2, Nível Junior.

Conforme apresentado na Tabela 11, uma possível estrutura de uma instância e suas propriedades pode ser assim definidas: nível de dificuldade do objeto de aprendizagem, criador, assunto, formato, objetivo, título, tamanho, data e descrição.

A seção a seguir apresentará uma possível descrição de uma Ontologia para Objetos de Aprendizagem feita pelo autor Lee (2005), a Java Ontologia de Objetos de Aprendizagem – JLOO.

2.3.2. Java Learning Object Ontology – JLOO

Segundo Gagne (1985, online), o termo “Objeto de Aprendizagem” pode ser definido como a Representação do Conhecimento Declarativo, que por sua vez seriam conhecimentos de “fatos”, “conceitos”, “princípios” e “modelo mental”. Diante desta definição o autor Lee (2005 p.2 - 3) relaciona a descrição destes quatro tipos de conhecimentos com a Linguagem de Programação Java:

- Fatos ou “Sabe o que”, seria uma declaração explicando a relação entre objetos ou eventos, por exemplo, em Java, os operadores aritméticos:

Adição	$A = B + C$
Subtração	$A = B - C$
Multiplicação	$A = B * C$
Divisão	$A = B / C$
Resto da Divisão (módulo)	$A = B \% C$

- Conceitos ou “Saiba que”, seriam classes de itens, palavras ou ideias que são distinguidas por um nome comum e compartilham algumas características comuns, incluindo vários exemplos. Em Java, pode-se correlacionar este conhecimento com as “declaração e atribuição” como apresentado na Figura 17.

```

1  int numero = 10;
2  boolean estaPronto = false;
3  String mensagem = "Olá!";

```

Figura 17 – Exemplo de Instruções de Declaração

- Princípios ou “Saber por que”, são regras relacionadas que podem ser usadas como “se-então” ou “causa-efeito”. Em Java, um exemplo de princípios seriam os condicionais e estrutura de dupla-seleção (*if/else*) (Figura 18) e seleção de múltipla-estrutura (*switch*) (Figura 19).

```

1  int month = 2;
2  if (month == 1) {
3      System.out.println("January");
4  } else if (month == 2) {
5      System.out.println("February");
6  }

```

Figura 18 - Exemplo de *if* e *else*

```

1  int month = 2;
2  String monthString;
3  switch (month) {
4      case 1: monthString = "January"; break;
5      case 2: monthString = "February"; break;
6  }
7  System.out.println(monthString);
8  }

```

Figura 19 - Exemplo de *switch*

- Modelo Mental é uma rede construída a partir de fatos, conceitos e princípios. Em Java, um exemplo é a “recursão” ou a “iteração”, conforme apresentado na Figura 20.

```

1 public int soma(int n) {
2     if (n <= 0)
3         return 0;
4     else
5         return n + soma(n-1);
6 }
7
8 System.out.println("Soma dos 10 primeiros numeros = " + soma(10));

```

Figura 20 - Exemplo de Recursividade

As classes são determinadas a partir da definição e explicação da relação entre Objetos de Aprendizagem e a Linguagem de Programação Java. Para uma visão geral desse trabalho, serão apresentados a sistematização das classes e hierarquia do modelo de dados e o modelo de teste de depuração do JLOO desenvolvidos no software Protegé 3.0.

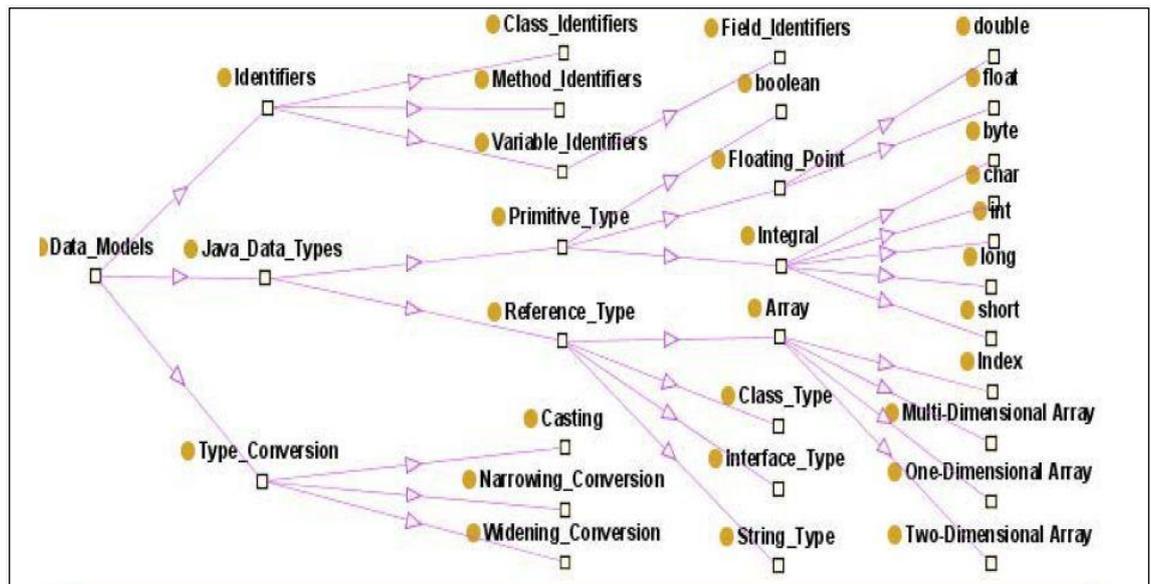


Figura 21 - Hierarquia de subclasse do "modelo de dados" (LEE, 2005, p. 3)

De acordo com a Figura 21, os nós folha são os “fatos”, que por sua vez são as definições do modelo do objeto de aprendizagem atômico, podendo ter diversos conteúdos de aprendizagem para os “fatos”. Um nó não folha representa a concretização de “conceitos”, “princípios” ou “modelo mental”. Uma instância de um nó não folha deverá se integrar com as instâncias criadas dos nós folhas que são descendentes diretos.

O modelo de dados apresentado anteriormente utiliza alguns conceitos de paradigmas de Orientação a Objetos e abrange alguns tipos de dados utilizados em Java. Para a descrição

original do modelo de dados foi utilizado o CC2001 (Diretrizes Curriculares para os cursos de graduação em Ciência da Computação da IEEE e ACM). Este modelo segue dois padrões para representar as estruturas dos dados:

- Abstratos (Descritos por um modelo);
- Concretos (Descritos por uma implementação).

Na Figura 22 é apresentada a hierarquia de subclasse de teste e depuração do JLOO, esta subclasse é uma das mais importantes, pois apontará e classificará os erros e a forma como resolvê-los.

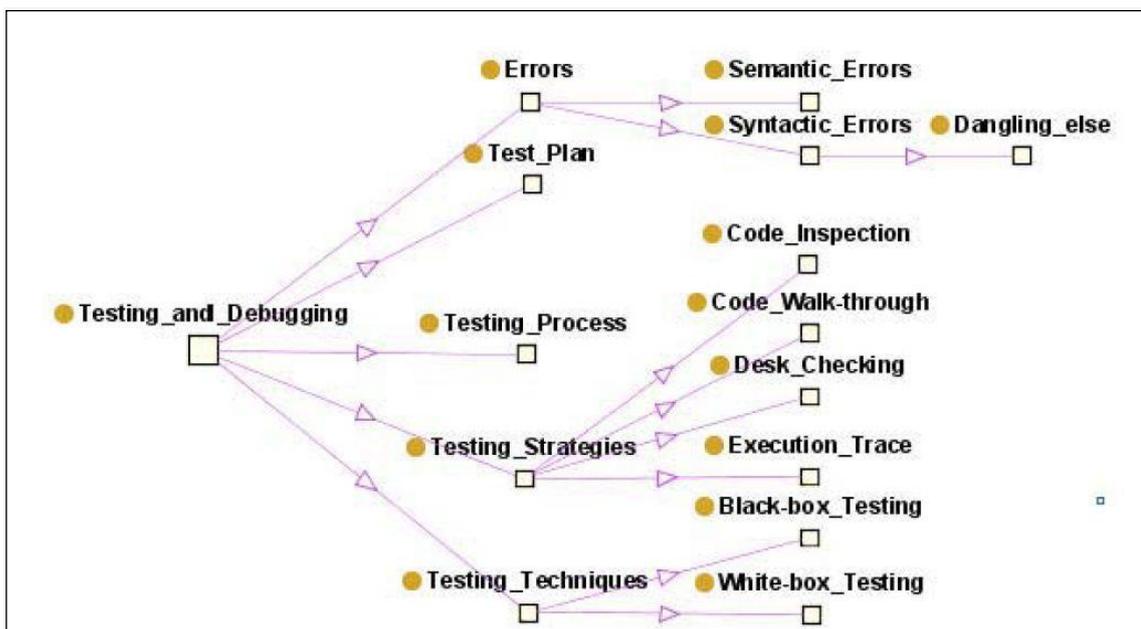


Figura 22 - Hierarquia de subclasse de "Teste e depuração" (LEE, 2005, p. 4).

A depuração ou *debugger* consiste em encontrar os defeitos do programa, e os testes servem para evitar estes defeitos. O JLOO classifica estes termos como *error types*, *testing*, *strategies*, *testing techniques*, como pode ser observado na Figura 22.

Lee (2005, p. 4) acrescenta, ainda, que são definidos três *slots* específicos para cada conceito:

- Pré-Requisitos – Objetos de Aprendizagem que são pré-requisitos para o entendimento de outro Objeto de Aprendizagem.
- Obrigatórios – Registra os Objetos de Aprendizagem obrigatórios para compreender determinado assunto. Logo, há uma condição para que se passe de um OA para outro.

- Opcionais – Registra os Objetos de Aprendizagem opcionais que podem melhorar a compreensão do objeto alvo.

Após a explicação e exemplificações de parte dos elementos que compõem o JLOO, pode-se considerar que a ontologia em questão pode ser utilizada como um guia para a apresentação de estratégias de aprendizagem. Assim, tendo como exemplo a definição de elementos que podem ser modelados para o entendimento da Linguagem de Programação Java, tem-se a exemplificação de alguns conceitos que podem facilitar a aprendizagem em outros domínios de conhecimento. Segundo Lee (2005, p.5), essa ontologia fornece uma maneira mais prática para construir uma ontologia de domínios específicos.

3. MATERIAIS E MÉTODOS

Nesta seção serão apresentados os detalhes referentes ao desenvolvimento deste trabalho, tais como o local e período de realização, o conjunto de hardware e software utilizados e a forma de trabalho adotada.

3.1. Local e Período

O desenvolvimento do presente trabalho deu-se inicialmente na Fábrica de Software, antigo PORTAL do Centro Universitário Luterano de Palmas (CEULP/ULBRA), bem como na residência do orientando em questão, Douglas Brito, localizada em Paraíso do Tocantins.

O período de desenvolvimento deste trabalho ocorreu no segundo semestre de 2011 e no primeiro semestre do ano de 2012, como um projeto desenvolvido para as disciplinas “Trabalho de Conclusão de Curso I” e “Trabalho de Conclusão de Curso II” do Curso de Sistemas de Informação. Desta forma, o desenvolvimento deste projeto iniciou no mês de Julho e tem como término o mês de Junho.

3.2. Materiais

O material utilizado pode ser dividido em três categorias: hardware, software e fontes bibliográficas. Esses materiais foram fornecidos pelo Centro Universitário Luterano de Palmas ou adquiridos gratuitamente pela Internet.

3.2.1. Hardware

O servidor utilizado para hospedar o módulo desenvolvido consiste em um processador Intel(R) Core(TM) 2 Quad Q8300 CPU 2.50 GHz, 4,00 GB de memória RAM e um HD com capacidade de 250 GB.

3.2.2. Software

Os softwares utilizados durante o trabalho constituíram-se na construção do referencial teórico, apresentações, edição de imagens, gerenciamento do projeto, modelagem e desenvolvimento do Módulo *Content* da Rede Social Acadêmica Konnen. Deste modo, dentre os softwares encontram-se aplicativos como processadores de texto, visualizadores de documentos, ferramentas para a edição de imagens, ferramenta de modelagem, gerenciador de

banco de dados e IDE para desenvolvimento. Tais softwares utilizados integram ferramentas gratuitas ou licenciadas pela própria Instituição, tal como o sistema operacional onde foram executados: Microsoft Windows 7 Professional 64 Bits.

Ferramentas utilizadas para a elaboração e edição de imagens:

- Adobe Photoshop CS5 12.0;
- Macromedia Fireworks CS5 11.0;
- CorelDRAW x5 15.0

Ferramenta utilizada na modelagem do sistema:

- Microsoft Office Visio 2010.

Para o gerenciamento do Projeto:

- Redmine versão 1.3.1 stable (MySQL).

Para o gerenciamento do banco de dados:

- MySQL Workbench versão 5.2.37 CE.

Para a implementação do projeto:

- IDE NetBeans versão 7.1 em PHP

Ferramenta para gerenciamento do código:

- Tortoise SVN 1.7.

Frameworks utilizados na implementação do projeto:

- Framework Kohana - é um *framework open-source* desenvolvido em PHP 5 para aplicações web. Ele usa o padrão de design HMVC (*Hierarchical Model-View-Controller*). Tem como características ser seguro, leve e fácil de usar. Uma de suas vantagens é que não limita o acesso a variáveis globais (GET, POST, COOKIE e SESSION), oferece recursos em cascata, módulos e heranças. – (<http://kohanaframework.org/>)
- JQuery - é um *framework javascript* voltado para o desenvolvimento web. Permite a navegação da árvore DOM, manipulação dos elementos DOM, manipulação de CSS, Gerenciamentos de eventos de forma simplificada, entre outras características – (<http://jquery.com/>)

3.2.3. Fontes de Referências Bibliográficas

O desenvolvimento deste trabalho se deu a partir da realização de pesquisas bibliográficas referentes aos conceitos, tecnologias e ferramentas que foram aplicados no Trabalho de Conclusão de Curso II, com o intuito de obter um maior embasamento teórico. Os diversos materiais adquiridos foram encontrados na “biblioteca” dos professores, em materiais de colegas e, principalmente, por recursos disponíveis na Internet. Dentre eles:

- Dissertações de Mestrado;
- Teses de Doutorado;
- Trabalhos de Conclusão de Curso;
- Publicações Científicas;
- Artigos;
- Livros;
- Sites.

3.3. Metodologia

A metodologia está relacionada à consecução das seguintes etapas, como apresentado na Figura 23:



Figura 23 - Metodologia do Projeto

- Estudo dos conceitos envolvidos (objeto de aprendizagem, LOM e ontologia). Estudo de algumas linguagens e tecnologias como: Linguagem de Programação PHP, Framework Kohana, Banco de Dados MySQL, CSS3 e o Framework JQuery;

- Entrevista com os professores pesquisadores responsáveis pelo projeto relacionado à Rede Social Acadêmica (Konnen), que é o contexto deste trabalho;
- Definição, a partir de entrevistas com os professores pesquisadores do projeto, dos elementos que estão compondo os metadados de OA;
- Definições, a partir de entrevistas com especialistas do domínio (Professores: Fabiano, Jackson e Parcilene), dos elementos relacionados ao conceito de Objeto de Aprendizagem que foram adicionados aos materiais didáticos, de alguns axiomas para a Ontologia, como também os mecanismos que agregam os metadados dos OA para estabelecer a Ontologia;
- Reuniões com os responsáveis pela parte técnica do Konnen para a definição do processo de desenvolvimento do módulo:
 - **Definição da Metodologia de Desenvolvimento;**
 - **Concepção e Levantamento de Requisitos:** a princípio o primeiro passo foi o entendimento do módulo e a descrição dos requisitos funcionais e não funcionais. Estes requisitos foram coletados por meio de reuniões e entrevistas com os responsáveis pelo Konnen. Na coleta de requisitos utilizou-se um formato mais explicativo textualmente (modelo de Caso de Uso Expandido ou *User Stories*);
 - **Gerência:** o responsável pelo projeto determinou um plano de ação, no qual definiu-se as atividades a serem realizadas, tendo um tempo estipulado para cada uma. Essas funcionalidades foram descritas utilizando a ferramenta de gerenciamento de projeto, *Redmine*;
 - **Modelagem:** foram definidos e criados os artefatos seguindo a metodologia de desenvolvimento de software;
 - **Prototipação:** nesta etapa foram criados protótipos de interface do software, bem como testes de uso do software. O desenvolvimento do protótipo foi pensado de maneira a deixar mais clara a interação entre usuário e sistema;
 - **Reimplementação do Gerenciador de Conteúdo:** devido à mudança de tecnologia, precisou reimplementar todo o gerenciador, antes feito em C# agora utilizando a linguagem PHP. Para a consecução dessa etapa realizou-se os seguintes processos:

- **Modelagem:** Foram adicionadas novas funcionalidades e com isso houve a necessidade de fazer uma nova modelagem do Gerenciador de Conteúdo, os artefatos utilizados foram:
 - *User Stories* (Descrição de telas e funcionalidades);
 - *Features* (Avaliação da complexidade, necessidade e prioridade de alguns requisitos);
 - Planejamento de Iterações (Depois da avaliação dos requisitos foram divididos por iterações);
 - *Checklist* (Após a implementação foi feito um *checklist* com intuito de identificar *bugs*).
 - **Prototipação:** Devido à nova proposta de interface para o projeto, foi necessário adaptar um novo *layout*;
 - **Implementação:** A estrutura do modelo relacional teve que ser refeita, pois passou de SQL para MySQL. A implementação do gerenciador foi feita utilizando a IDE *NetBeans* junto ao *Framework Kohana* (PHP). Devido o *kohana* trabalhar com o padrão MVC, o gerenciador de conteúdo foi implementado por partes, seguindo o modelo (*Model, View e Controller*);
 - **Testes:** Testes de unidade na lógica de negócio utilizando a técnica TDD - *Test Driven Development*. Também foram realizados Testes de usabilidade com os responsáveis pela implementação.
- **Implementação:** após o termino dos artefatos e a aprovação dos protótipos, iniciou-se a implementação do modulo *Content*, foi adotado o padrão de arquitetura de software baseado HMVC (*Hierarquical Model-View-Controller*). Nesse padrão a aplicação é sistematizada em três componentes principais: *Model* (modelo), *View* (visão) e *Controller* (controlador) (MVC):
 - Primeiramente realizou-se a criação do banco de dados relacional;
 - Logo em seguida foi feita a codificação dos *models*, que é parte da aplicação que implementa a lógica de negócio para o domínio de dados do aplicativo, em que os objetos do *model* recuperam e armazenam o estado do *model* no banco de dados;
 - Depois da implementação dessa camada, partiu-se para a implementação dos *controllers*, que tem como objetivo fazer o uso da

View e do *Model*, permitindo uma organização das requisições feitas entre eles e selecionando o que será exibido nas *Views*;

- E em paralelo a implementação dos *controllers*, foi realizada a codificação das *Views*, que são componentes que exibem a interface de aplicativo para o usuário, desenvolvida seguindo os dados do *Model*.
- E, por fim, a implantação e integração com os demais módulos existentes no projeto;
- **Testes:** foram realizados os seguintes testes:
 - Testes de unidade na lógica de negócio utilizando a técnica TDD - *Test Driven Development* (realizados na própria IDE de desenvolvimento com o auxílio da ferramenta *xdebug*);
 - Testes de usabilidade com os usuários e responsáveis, onde todos envolvidos relataram os *bugs* e sugestões no *Redmine*;
 - Teste da ferramenta como um todo para verificar e corrigir as inconsistências e erros.

4. RESULTADOS E DISCUSSÃO

A presente seção tem como objetivo apresentar os resultados obtidos com o desenvolvimento do sistema proposto neste trabalho, que permitirá a criação de conteúdos de diferentes naturezas (ex.: imagem, vídeo, áudio e texto). O sistema proporcionará também aos usuários uma maior interação, pelo fato de permitir o compartilhamento dos conteúdos entre usuários, grupos de usuário e turmas. A partir do compartilhamento de conteúdos e da possibilidade de colaboração entre os usuários, o conhecimento passa a ser representado e disseminado. Agrega-se a isso, o fato de que com a utilização de objetos de aprendizagem a partir de uma ontologia dos conteúdos será possível ter um entendimento amplo das propriedades e características das classes e relacionamentos existentes em um contexto, podendo realizar inferências sobre aspectos relevantes que auxiliarão no processo de ensino-aprendizagem.

As próximas seções apresentarão a arquitetura do Konnen, a arquitetura do módulo *Content*, a representação de Conteúdos como Objetos de Aprendizagem e por fim a Ontologia no modelo LOM+OA.

4.1. Arquitetura do Konnen

De acordo com o Projeto de Pesquisa “Aprendizagem Organizacional Através de uma Rede de Gestão de Conhecimento”, do curso de Sistemas de Informação do CEULP/ULBRA, a Rede Social Acadêmica Konnen tem uma arquitetura definida em múltiplas camadas e, com base no conceito de *plugin*, permite que sejam criados aplicativos com o objetivo de fornecer um modelo de extensões capaz de adicionar funcionalidade sob demanda para o sistema, usuários e grupos de usuários (SOUZA, *et al*, 2012, p. 5). A figura a seguir (Figura 24) apresenta os Aplicativos (subsistemas) do Konnen.

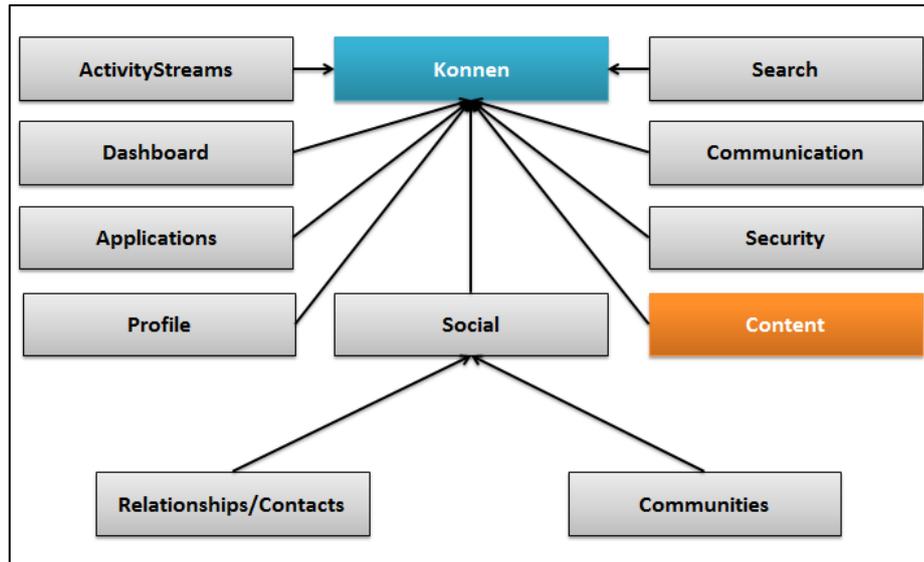


Figura 24 - Aplicativos (subsistemas) do Konnen (Modificado de SOUZA, *et al*, 2012, p.5).

Como apresentado na Figura 24, o Konnen é constituído por onze módulos (*ActivityStreams*, *Dashboard*, *Applications*, *Profile*, *Search*, *Communication*, *Security*, *Content*, *Social*, *Relationship* e *Communities*) e o módulo em destaque, *Content*, é o foco deste trabalho. Devido a utilização do *framework Kohana* (uma estrutura em PHP 5 que trabalha com o conceito HMVC (*Hierarchical Model-View-Controller*), para o funcionamento de um módulo são necessárias algumas configurações no projeto, como pode ser observado na figura a seguir (Figura 25).

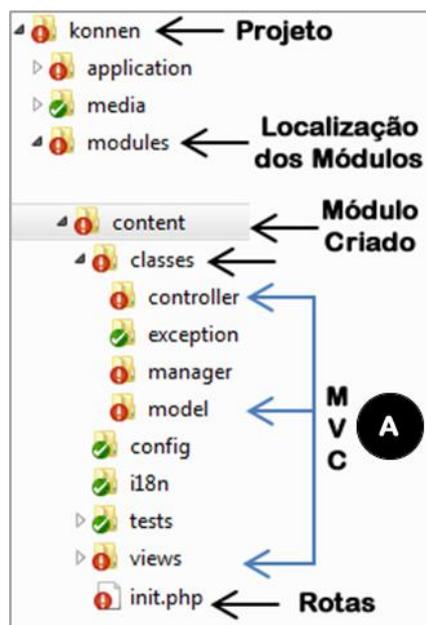


Figura 25 - Estrutura de Pastas para a criação de um módulo

Conforme apresentado na Figura 25, tem-se uma estrutura de pastas e o diretório que armazenará o módulo a ser criado está localizado dentro da pasta 'modules'. Devido o projeto ser implementado em camadas, é importante que sua organização seja separada por categorias de forma que, para cada parte do módulo (*Model*, *View* e *Controller*), exista um local específico para a criação das classes, métodos e interfaces gráfica, assim como pode ser visto na imagem anterior (Figura 25- A). O arquivo 'ini.php' é responsável por registrar todas as rotas relacionadas ao módulo e a sua implantação foi feita através do arquivo 'application/bootstrap.php', adicionando-se a seguinte linha de código no método Kohana::modules() :

```
'content' => MODPATH.'content';
```

A figura a seguir (Figura 26) dá uma visão da arquitetura do Sistema, no sentido de demonstrar os níveis dos aplicativos fundamentais para o funcionamento do Konnen (aplicativos de sistema) e extensões (aplicativos de usuário).

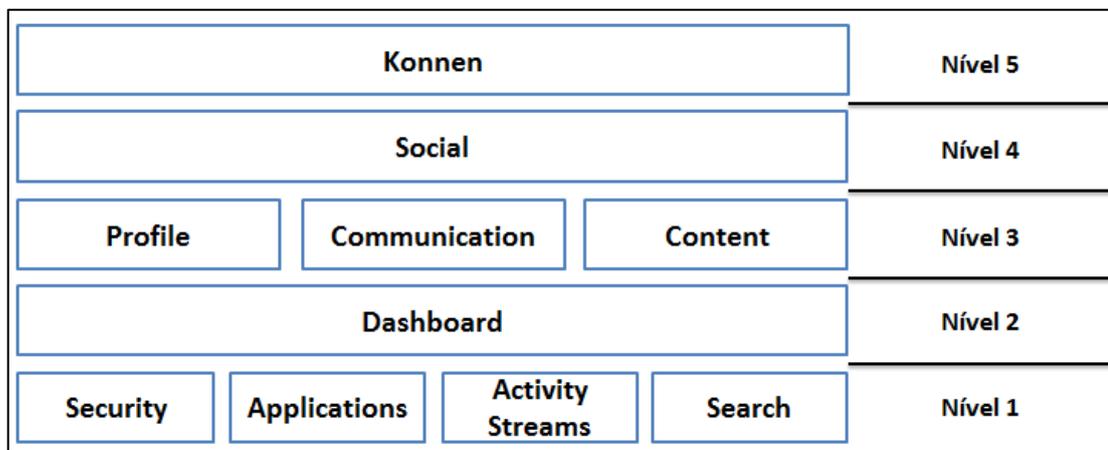


Figura 26 - Níveis de Aplicativos Fundamentais para o funcionamento do Konnen (SOUZA, *et al*, 2012).

Conforme evidenciado no Projeto de Pesquisa, Souza (*et al*, 2012) relata que os aplicativos de nível 1 são os aplicativos de sistema: *Security*, *Applications*, *ActivityStreams*, *Search*. Os de nível 1 podem ou não ter interface gráfica e são utilizados por outros aplicativos. No Nível 2 tem-se os aplicativos intermediários, por exemplo, o *Dashboard*, que é fundamental para o sistema e encontra-se entre o sistema e o usuário, pois representa a área de trabalho que reúne informações de outros aplicativos. Nos Níveis 3 e 4 estão localizados os aplicativos de usuários: *Profile*, *Communication*, *Content* e *Social*. De acordo com Souza (*et*

al, 2012), o fluxo de dados ocorre no sentido dos aplicativos de nível mais alto para os de nível mais baixo. Por exemplo, o aplicativo *Social* (nível 4) consulta o aplicativo *Security* (nível 1) por informações sobre privilégios de acesso de um usuário, mas o aplicativo *Security* (nível 1) não poderá extrair informações do aplicativo *Social* (nível 4). Os aplicativos de sistema (Nível 1) servem aos demais. O aplicativo intermediário *Dashboard* (nível 2) consulta informações dos demais níveis. Aplicativos de mesmo nível também podem trocar informações entre si.

4.2. Arquitetura do Módulo *Content*

O trabalho em questão faz parte do módulo *Content*, representa um aplicativo de usuário e está inserido no terceiro nível de aplicativos definidos no *Konnen*. Para facilitar o entendimento desse módulo, foi desenvolvida sua estrutura principal, que destaca o funcionamento do sistema de gerenciamento de conteúdo, apresentado na Figura 27.

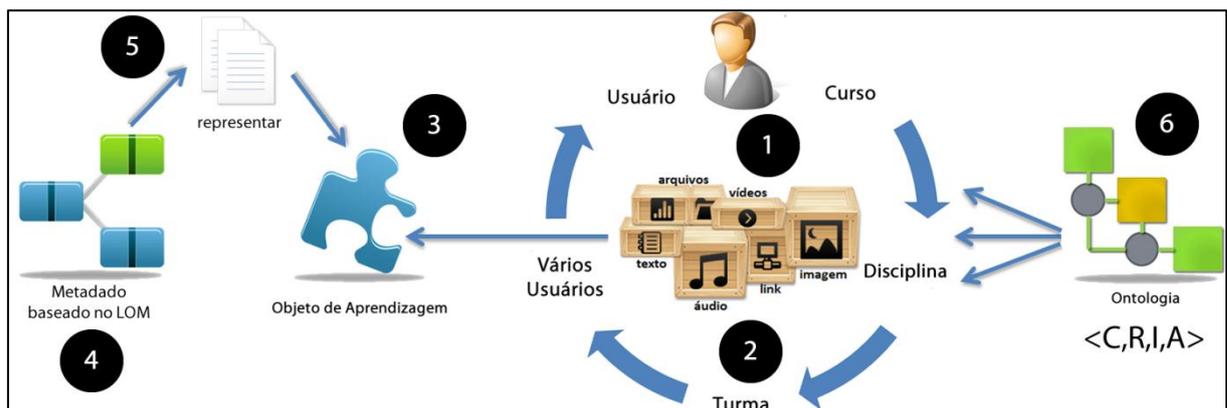


Figura 27 - Arquitetura do Gerenciador de Conteúdo

De acordo com a Figura 27, os conteúdos (1) podem ser disponibilizados em diferentes contextos (2), por exemplo, em um ou mais cursos, disciplinas, turmas, para um usuário ou vários usuários, entre outros. A partir do momento em que estes conteúdos (entidades digitais) são usados para o aprendizado, educação ou treinamento e possuem algumas características, como acessibilidade (conteúdo esteja em um formato que o usuário tenha condições de visualizá-lo, por exemplo, por meio de um dispositivo móvel), reusabilidade (ser reutilizado em diferentes contextos, por exemplo, curso, disciplina etc), entre outras, eles podem ser considerados como um Objeto de Aprendizagem (3).

Para a descrição destes OAs, foram utilizados conceitos baseados no LOM - *Learning*

Object Metadado (4). Estes metadados serão responsáveis por descrever um OA (3) para o usuário, ou seja, informar condições para que ele seja reproduzido, a sua versão, tipo de interatividade, entre outros, por exemplo, no caso do vídeo, tem-se o formato do vídeo, a duração, codificação etc.

Com a descrição dos objetos e utilizando o conceito da 4-tupla $\langle C,R,I,A \rangle$ (Seção 2.2 - Ontologia), a ontologia (6) contextualizará o domínio a partir da identificação das classes, das relações e instâncias. Logo, um conjunto de axiomas poderá gerar determinadas inferências como, por exemplo, no momento em que um material é vinculado a mais de uma disciplina pode-se inferir a relação de “interdisciplinaridade”. Assim, ao serem definidos os axiomas, grupos de domínio podem ser construídos e analisados pelo sistema, permitindo que os gestores da instituição possam verificar onde existem relações (interdisciplinaridade, multidisciplinaridade e transversalidade) entre os conceitos (curso, disciplina e turma) e a partir disso criar estratégias de aprendizagem de acordo com a necessidade de cada contexto.

Algumas mudanças ocorreram no decorrer do desenvolvimento desse projeto de forma a produzir determinadas melhorias. Uma mudança que pode ser evidenciada e que resultou na reimplementação do Gerenciador de Conteúdo foi a troca de tecnologia, que passou da plataforma .NET sob a linguagem C# para PHP (*framework Kohana*). Segundo os responsáveis pela parte técnica do projeto (Coordenação da Fábrica de Software do CEULP/ULBRA), o motivo da mudança de tecnologia deve-se ao fato dessas novas ferramentas permitirem a criação de softwares sobre uma plataforma escalável, com foco no alto desempenho.

A versão anterior do Gerenciador de Conteúdo definida em Brito (2011, p.49 e 50) apresentava algumas limitações no gerenciamento do conteúdo, pois nem todos os tipos de conteúdos necessários ao contexto estavam disponibilizados. A adição de novos tipos de dados possibilita que o usuário ofereça conteúdos mais elaborados, pois diminui as limitações de documentos sugeridos.

Para exemplificar essa melhoria será apresentado o seguinte cenário: um educador do curso de engenharia civil que utilizasse a versão anterior do gerenciador de conteúdo não teria opção de disponibilizar conteúdo do tipo arquivo. Com a atualização do gerenciador, o educador pode, por exemplo, adicionar uma planta baixa da arquitetura de um edifício (arquivo em formato dwg), os alunos então poderão ter acesso a este material de referência.

Os trechos de código a seguir (Figura 28) serão apresentados com o intuito de demonstrar as limitações existentes na primeira versão do gerenciador no que diz respeito aos processos de cadastro e compartilhamento de conteúdo, assim pode-se ter um entendimento

melhor da importância das funcionalidades acrescentadas na nova versão.

<pre> var conteudos = new Array(); @using (Html.BeginForm("Pesquisar", "Content")) { <h2> Lista de Conteúdos</h2> Novo Texto
 Novo Vídeo/Imagem
 <input type="page" name="pesquisar" id="pesquisar" value="@ViewBag.P <input type="submit" value="pesquisar" />
 Home <table class="tabelaListarConteudos"> </h2> </pre>	<div style="background-color: black; color: white; border-radius: 50%; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 0 auto;">1</div>
<pre> public ActionResult CreateContent(ContentModel contentModel) { Content content = null; content = database.Get(c => c.Id == contentModel.Id); bool create = false; if (content == null) { content = new Content(); content.UserId = databaseUser.Get(u => u.Name.Equals(User.Identity.Name)).Id; create = true; } if (create) { content.Date = DateTime.Now; database.Add(content); } database.Save(); } </pre>	<div style="background-color: black; color: white; border-radius: 50%; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 0 auto;">2</div>
<pre> public ActionResult Compartilhar(long Id) { var IdUsuario = databaseUser.Get(u => u.Name.Equals(User.Identity.Name)).Id; var lista_usuario = databaseUser.Table.Where(u => u.Id != IdUsuario).ToList(); ViewData["IdConteudo"] = Id; return View(lista_usuario); } </pre>	<div style="background-color: black; color: white; border-radius: 50%; width: 40px; height: 40px; display: flex; align-items: center; justify-content: center; margin: 0 auto;">3</div>

Figura 28 – Parte da View ManagerContent, Método CreateContent e Método Compartilhar (BRITO, 2011 p. 49)

De acordo com a Figura 28, essa implementação possibilita criar apenas conteúdos do tipo texto, vídeo e imagem, conforme apresentado na *View* (1). Na *action createContent* do *controller* (2) tem-se a possibilidade de realizar o compartilhamento dos conteúdos, entretanto este compartilhamento era realizado somente entre os usuários (*action compartilhar* do *controller* (3)), pois o módulo de Grupo ainda não tinha sido implementado. No presente trabalho foram acrescentadas outras funcionalidades, como:

- Cadastro de conteúdo do tipo arquivo (por meio de *upload* ou através de um *link*), *link*, imagem (*upload* ou através de um *link*), texto e vídeo;
- Criação de coleções de conteúdos;
- Postagem de conteúdos dentro de grupo, turmas, disciplinas, cursos etc;
- Compartilhamento de conteúdos com usuários e grupos de usuários (turmas, disciplinas e cursos);
- Avaliação de um conteúdo;
- Realização de anotações de um conteúdo;
- Organização dos conteúdos por meio de estruturas (em rede, hierárquica ou linear).
- Versionamento dos conteúdos.

A seguinte imagem (Figura 29) ilustra parte da reimplementação do gerenciador de conteúdo, o trecho mostra a criação de um conteúdo do tipo imagem.

```

1  if ($type == 'file'){
2      if (isset($_FILES['file']['name']) && $_FILES['file']['name']) {
3          $file = $_FILES['file'];
4          $file = Manager_Content::save_file($user, $file);
5          $file_id = $file['id'];
6      } else {
7          if (isset($_POST['link'])) {
8              $url = $_POST['link'];
9              $file_name = pathinfo($url);
10             $valid_types_file = Kohana::$config->load('content')
11                 ->get('valid_types_file');
12             if (!in_array($file_name['extension'], $valid_types_file)) {
13                 throw new Exception_Media('Upload::UnsupportedType', 415);
14             }
15             $file_info = Media::save_file_from_url($url, 'uploads/content/');
16             $file_info['complete_file_path'] = str_replace(Kohana::$config
17                 ->load('media')->get('upload_relative_path'),
18                 Upload::$default_directory, $file_info['directory']) .
19                 $file_info['file_name'] . '.' . $file_info['ext'];
20             $file_info['file_on_disk_row'] = ORM::factory('fileondisk');
21             $file_info['file_on_disk_row']
22                 ->original_file_name = $file_info['original_name'];
23             $file_info['file_on_disk_row']
24                 ->path = $file_info['file_name'] . '.' . $file_info['ext'];
25             $file_info['file_on_disk_row']->save();
26         }
27     }
28 }

```

Figura 29 - Action Save do Controller Content.

O método apresentado na Figura 29 é iniciado com uma verificação do tipo de

conteúdo a ser cadastrado (linha 1). Se o tipo escolhido pelo usuário for o tipo "file", indica a criação de um arquivo (.rar, .pdf, .doc, entre outros). Na Linha 2, o usuário informa o modo como esse arquivo será carregado (*upload* ou através de um *link*), então é verificado (linha 2) se a variável global `$_FILES` possui alguma instância, ou seja, se as posições "file" e "name" do *array* `$_FILES` não estão vazias. Se não tiverem vazias, a opção escolhida pelos usuários foi a de *upload*, que por consequência o *controller* invoca o método `save_file` do *manager* e o arquivo será salvo (linha 4), assim é criado um novo registro no banco utilizando o *model* relacionado ao método (`Model_Content`). Se as posições "file" e "name" do *array* `$_FILES` estiverem vazias, indica que o usuário escolheu a opção de cadastrar um conteúdo por meio de um *link* e é verificado (linha 7) se a posição "link" do *array* `$_POST` não está vazia; se isso for verdadeiro, a variável `$url` (linha 8) recebe como valor uma *string* contendo o *link* informado pelo usuário. Na linha 9, a variável `$file_name` recebe as informações (nome e extensão do arquivo) relacionadas ao arquivo que está sendo postado. A variável da linha 10 armazena os tipos das extensões dos arquivos que são aceitos e, caso o usuário queira postar um arquivo cuja extensão não esteja contida na lista, é disparada uma exceção (linha 13), e se a extensão do arquivo estiver relacionado a lista o mesmo é salvo no banco por meio do método `save_file_from_url` invocado pelo *controller*.

Para postar um conteúdo dentro de um grupo realiza-se o mesmo processo de um *post* na área de um usuário, mudando apenas a forma de apresentação. O trecho de código a seguir mostra a parte que identifica que um conteúdo esta sendo postando dentro de um grupo (Figura 30).

```

1  $group_id = NULL;
2  if (isset($_POST['group_id'])) {
3      $group_id = $_POST['group_id'] == 0 ? NULL : $_POST['group_id'];
4  }
```

Figura 30 - Action Save do Controller Content

Conforme a Figura 30, na Linha 1 é instanciada a variável `$group_id`, utilizada para verificar se o conteúdo está sendo postado no contexto grupo ou no perfil do usuário, pois se a posição "group_id" do *array* `$_POST` estiver vazia o conteúdo é criado no perfil do usuário, caso contrário, é criado no perfil do grupo (linha 2). Quando um conteúdo é postado no perfil de um grupo é indicado no perfil do usuário que um conteúdo foi publicado via um

grupo, conforme pode ser observado na Figura 31.



 **parcilene** **Via** Lógica de Predicados
4 days ago [Ementa](#)

Formalização de argumentos. Estudos sobre a Lógica Proposicional e de Predicados. Práticas de resolução de fórmulas através de métodos de refutação e de dedução natural.

[Ver mais](#)

Você e outra(s) 1 pessoa(s) gostaram deste conteúdo

[Compartilhar](#)

Comentários
Nenhum comentário até o momento. Seja o primeiro a comentar!

Figura 31 - Conteúdo publicado via Grupo

Como pode ser observado na Figura 31, o usuário (Parcilene) publicou um conteúdo (tipo texto) no grupo da disciplina de Lógica de Predicados. Este conteúdo será apresentado desta maneira na tela inicial dos demais usuários que pertencem ao grupo, por exemplo, o nome do usuário que postou “via” nome do grupo que foi postado seguido do título e do próprio conteúdo. O usuário vai poder comentar sobre este conteúdo e até mesmo compartilhar com outros usuários ou grupos. A seguir será apresentado o funcionamento do compartilhamento de um conteúdo.

```

1 public function action_share() {
2     $user = auth::instance()->get_user();
3     if ($_POST) {
4         $content_id = $_POST['content_id'];
5
6         if (isset($_POST['ids'])) {
7             $recipients = array();
8             for ($i = 0; $i < count($_POST['ids']); $i++) {
9                 $recipients[] = array(
10                    'type' => $_POST['types'][$i],
11                    'id' => $_POST['ids'][$i]
12                );
13            }
14        }
15        Manager_Content::share($user->id, $content_id, $recipients);
16    }
17 }
18 $request_url = $this->request->referrer();
19 $this->request->redirect($request_url);
20 }

```

Figura 32 - Action Share do Controller Content

Conforme pode ser observado na Figura 32, na Linha 2 a variável `$user` armazena o usuário que está *logado* no momento do compartilhamento. No instante em que o usuário clica para compartilhar um conteúdo, ele escolherá o(os) destinatário(os) (usuário(os) ou grupo(os)). A partir dessa ação, os dados do destinatário são armazenados no *array* `$recipients` nas posições “type” (usuário ou grupo) e “id” (Linha 7 à 14) . Após definir o(os) destinatário(os), o *controller* invoca o método `Share` e, conseqüentemente, o conteúdo será compartilhado. A Figura 33 apresenta a tela que lista os possíveis destinatários.



Figura 33 - Tela da lista de amigos e grupos de um usuário

Ao clicar em “compartilhar” (Figura 32), o sistema informará ao usuário uma tela contendo a lista de amigos e grupos que ele faz parte, conforme apresentado na figura a cima (Figura 33). Após selecionar os usuários que ele pretende compartilhar e confirmar a ação, será criado um registro na tabela `content_recipients` contendo o identificador do conteúdo compartilhado, o tipo do destinatário (`user` ou `group`), identificador do destinatário, identificador do remetente e a data do compartilhamento.

A reimplementação do Gerenciador de Conteúdo trouxe novidades no que diz respeito à descrição de um conteúdo, pois com a utilização de metadados que descrevam os conteúdos, será possível, por exemplo, aumentar o valor de um conhecimento, ou seja, a partir do momento que um conteúdo passa a ser compartilhado em diferentes contextos ele pode ser melhorado cada vez mais fazendo com que sua consolidação cresça de forma espontânea. A seção a seguir (4.3) apresenta os metadados dos objetos de aprendizagem que descreverão um conteúdo.

4.3. Representação de Conteúdos como Objetos de Aprendizagem

A definição dos Metadados que representarão os Objetos de Aprendizagem foi baseada nas categorias existentes do LOM. Após reuniões com os responsáveis pelo projeto a fim de definir as categorias preponderantes para o contexto do trabalho, chegou-se à estrutura apresentada na imagem a seguir (Figura 34), na qual estão definidas as categorias selecionadas bem como quem será responsável por preencher seus elementos (Usuário ou Sistema).

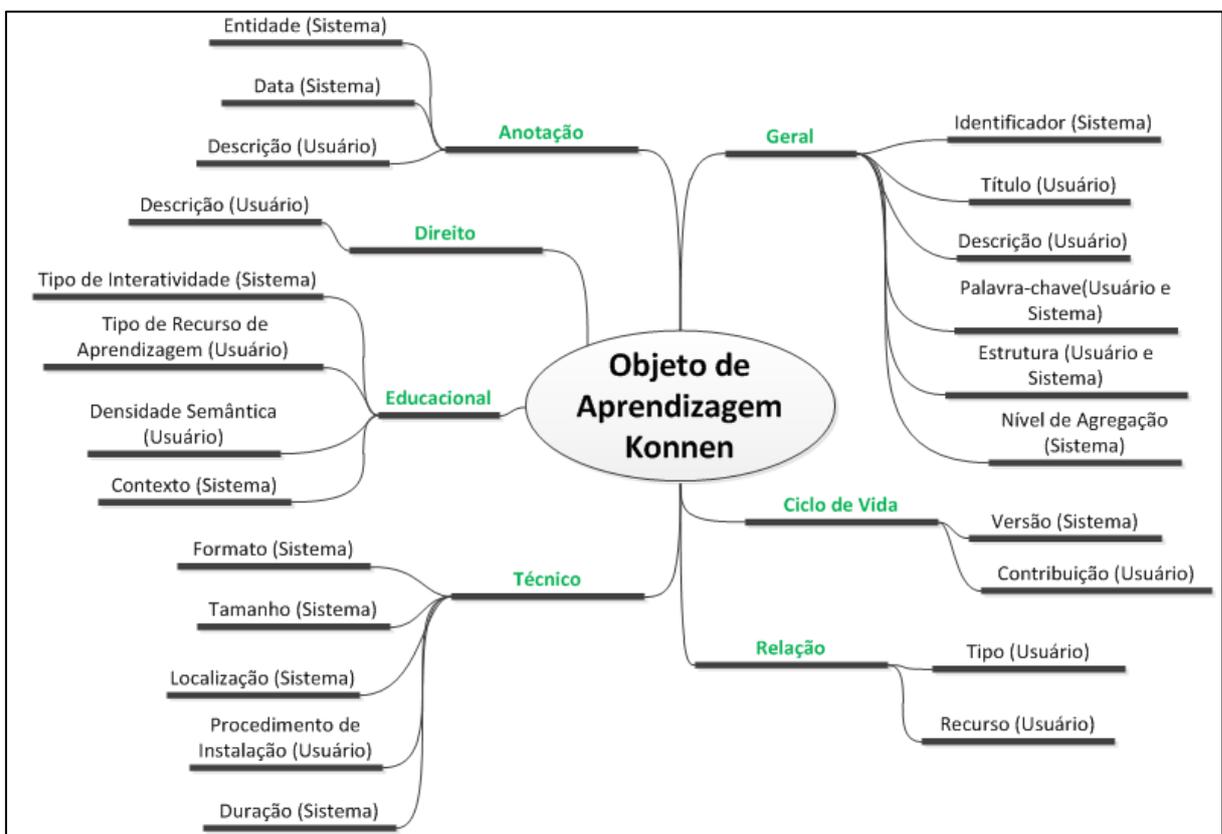


Figura 34 - Estrutura do Metadado de Objeto de Aprendizagem baseado no LOM.

Como pode ser observado na Figura 26, a estrutura do Metadado é composta por sete categorias (Geral, Ciclo de Vida, Técnico, Educacional, Direito, Relação e Anotação) e dentro de cada categoria existe os elementos que as compõem. Para o preenchimento dos elementos que compõem as categorias do metadado de objeto de aprendizagem, priorizou-se que a maior parte das tarefas fosse realizada automaticamente pelo sistema, de forma que não sobrecarregasse o usuário no preenchimento dos dados.

Para cada categoria foi criada uma entidade no banco de dados. A imagem a seguir

(Figura 35) ilustra o modelo relacional do metadado de objeto de aprendizagem.

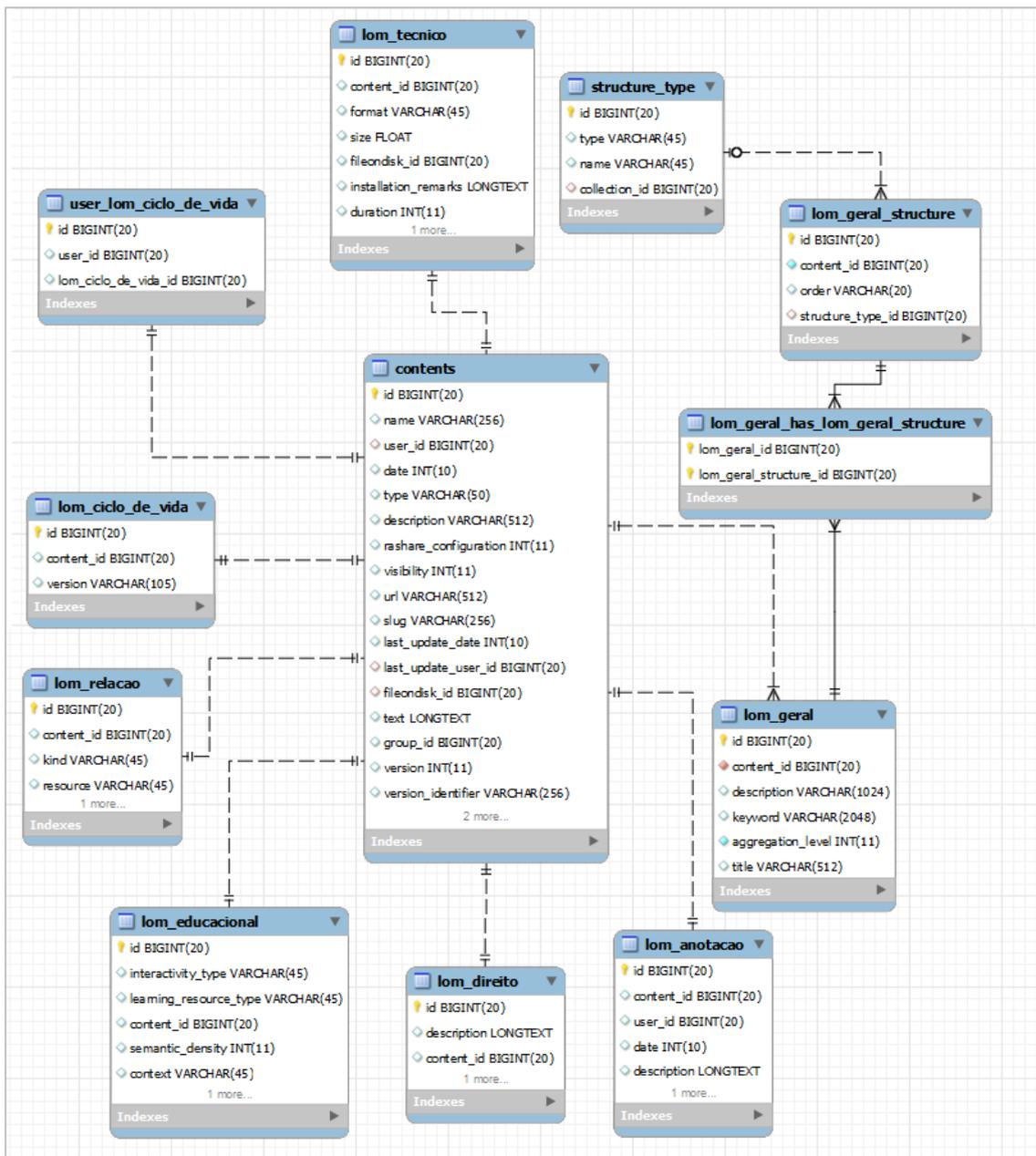


Figura 35 - Modelo Relacional do Metadado de Objeto de Aprendizagem

De acordo com a Figura 35, o modelo relacional desenvolvido para armazenar os Metadados dos Objetos de Aprendizagem é composto por 12 entidades de relacionamento, tendo como principal tabela a entidade “contents”. Esse modelo é extensível, pois permite o armazenamento das informações de vários tipos de conteúdos (Texto, Imagem, Vídeo, *Link*, Arquivos) e tem como uma das principais finalidades a organização das mesmas.

Para melhor entendimento da implementação da estrutura dos metadados do objeto de aprendizagem, as próximas seções explicarão separadamente cada categoria:

4.3.1. Geral

A categoria Geral tem como principal objetivo agrupar as informações básicas que descrevem o Objeto de Aprendizagem. A imagem a seguir apresenta os dois elementos que compõem essa categoria.

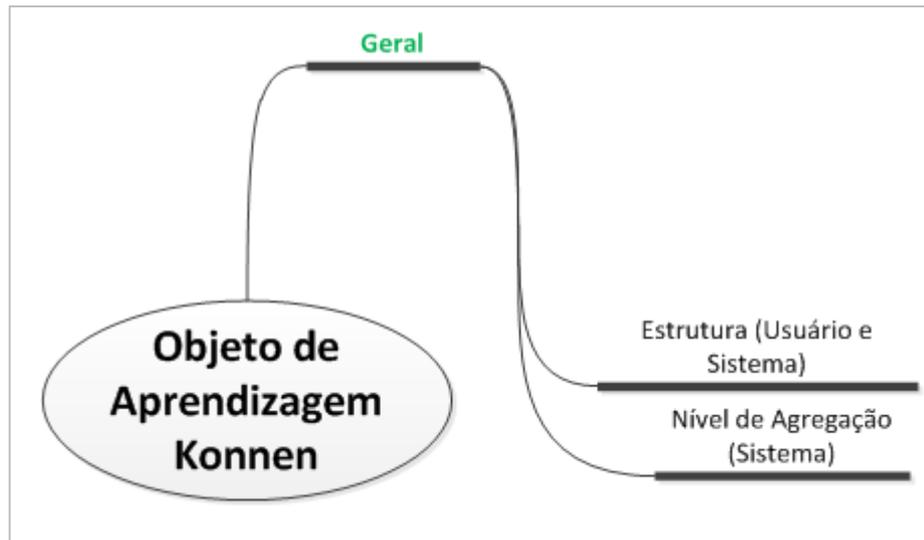


Figura 36 - Metadado de Objeto de Aprendizagem (Geral - A)

A Figura 36 evidencia os elementos Estrutura e Nível de Agregação de um Objeto de Aprendizagem. O Nível de Agregação vai depender da Estrutura, ou seja, se a Estrutura for Atômica, o Nível de Agregação será 1, se for uma Coleção, o Nível será 2, Em Rede ou Hierárquica, tem-se o Nível 3 ou se a Estrutura for Linear, o nível será 4. O trecho de código a seguir mostra a *action* responsável por armazenar uma estrutura.

```

1 public function action_structureSave(){
2     $contents_id = $_POST['contents_id'];
3     $orders = $_POST['orders'];
4     $titulo = $_POST['titulo'];
5     $type = $_POST['type'];
6
7     Manager_Content::save_lom_geral_structure($contents_id, $orders, $titulo, $type);
8     $this->request->redirect('content/listCompleto');
9 }

```

Figura 37 - Action StructureSave do Controller Content

Conforme apresentado na `action_structureSave()` (Figura 37), na linha 2 a

variável instanciada (`$contents_id`) armazena a lista dos identificadores dos conteúdos guardados na posição “`contentes_id`” do *array* `$_POST`, a mesma ação é feita na posição “`$orders`” que, de acordo com o tipo de estrutura escolhida (em rede, hierárquica ou linear), armazenará um valor (*emrede*, *hierarquia*, *linear*).

Para estruturas do tipo Atômica e Coleção, é realizada outra verificação. Por exemplo, no momento em que um usuário posta um conteúdo em seu perfil, pode-se inferir que sua estrutura é do tipo atômica, pois não existe ligação com nenhuma coleção ou outro conteúdo. Já no momento que um usuário cria uma coleção e adiciona vários conteúdos na mesma e estes conteúdos não possuem uma ligação a não ser o fato de estarem em um mesmo contexto (pasta), é inferido que a estrutura possui o tipo “Coleção”.

Por exemplo, se foi escolhida a estrutura “em rede”, os possíveis valores seriam:

- `$content_id[] = 1, $orders[] = 1`
- `$content_id[] = 2, $orders[] = 1`
- `$content_id[] = 3, $orders[] = 1`

De acordo com a estrutura anterior pode-se dizer que os conteúdos (1, 2 e 3) estão interligados na ordem 1. Caso o usuário tenha escolhido a estrutura “hierárquica”, poderiam ter os seguintes valores:

- `$content_id[] = 1, $orders[] = 1`
- `$content_id[] = 2, $orders[] = 1.1`
- `$content_id[] = 3, $orders[] = 1.2`

A hierarquia, então, é definida da seguinte forma: conteúdo 1 é pai dos conteúdos 2 e 3, ou seja, a estrutura pode ser entendida como uma topologia em árvore. Já na estrutura “linear”, os elementos de um mesmo nível (1.1, 1.2) também são considerados como uma sequência, o que contribui com o processo de definição de uma linha de estudo que melhor se adequa ao processo de ensino-aprendizagem de um dado conteúdo.

Os processos anteriores serão exemplificados de acordo com definição de estruturas feita por parte do usuário. A Figura 38 ilustra a tela em que o usuário informa como deseja criar as estruturas (Em Rede, Hierárquica e Linear).

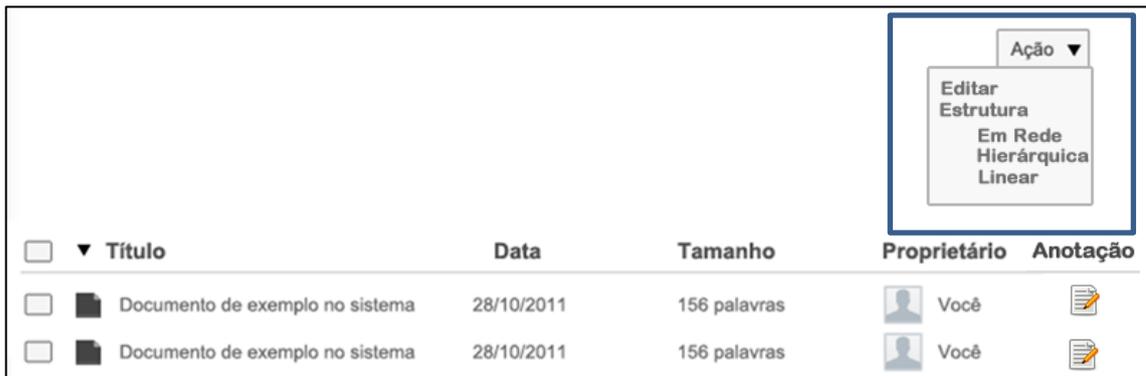


Figura 38 - Parte da Tela do Gerenciador de Conteúdo (Geral)

Inicialmente na tela do Gerenciador de Conteúdo (Figura 38) o usuário encontrará todos os conteúdos criados por ele ou compartilhados com ele. Partindo do pressuposto de que o usuário seja um professor e deseja encaminhar para sua turma uma sequência de estudos, ou seja, uma estrutura organizada de estudo, é necessário informar primeiramente no menu “Ação” o modo como os conteúdos deverão ser estruturados. Após a escolha da estrutura, o sistema fornece uma interface que permite a ordenação dos elementos. Por exemplo, se a opção de estrutura for “Em Rede”, o usuário informará os conteúdos que têm uma ligação entre si e realiza a ação de compartilhamento. O exemplo a seguir (Figura 39) demonstra como os conteúdos são ligados em rede.

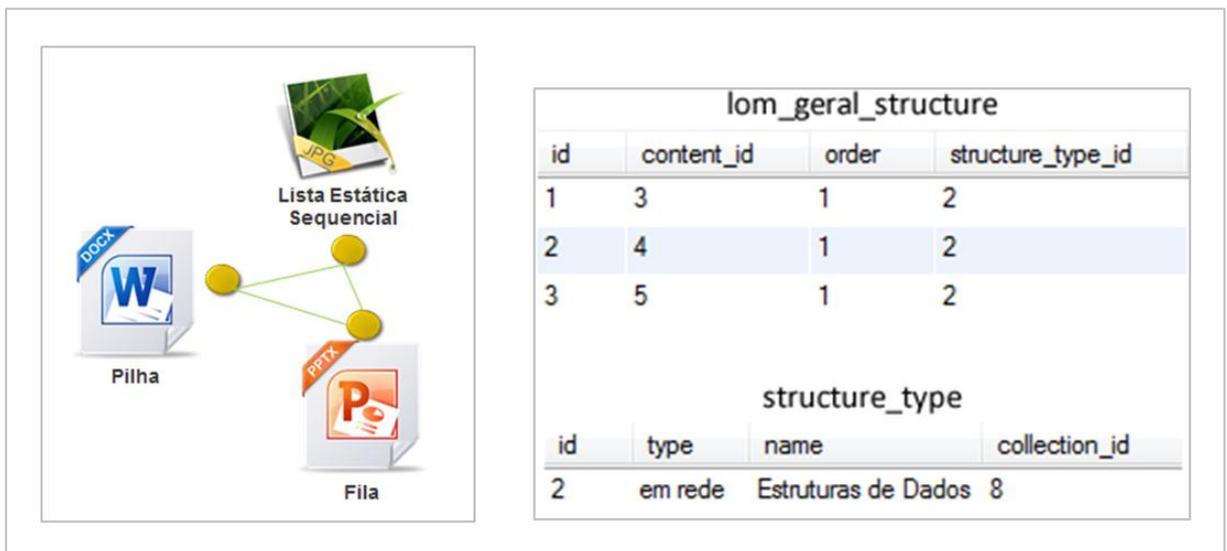


Figura 39 - Estrutura (Em Rede)

Como pode ser observado na Figura 39, os conteúdos (pilha, lista estática sequencial e

fila) fazem parte da mesma estrutura (em rede). Assim, a partir do momento que o usuário define a estrutura é criado um registro nas tabelas `structure_type` e `lom_geral_structure`, indicando que os conteúdos (`id = 3`, `id = 4` e `id = 5`) estão relacionados na mesma ordem (`order = 1`).

Se a opção for “Hierárquica”, o sistema permite que o usuário informe a hierarquia (pai e filho) dos conteúdos (não se importando com a sequência correta dos níveis) que deseja compartilhar com sua turma. Essa hierarquia é definida no campo `order`.

Por exemplo, o conteúdo com identificador 3 (`order = 1`) é pai dos conteúdos 4 (`order = 1.1`), 5 (`order = 1.2`) e 6 (`order = 1.3`), já os conteúdos 7 (`order = 1.1.1`) e 8 (`order = 1.1.2`) são filhos do conteúdo 4. A imagem a seguir ilustra a exemplificação (Figura 40).

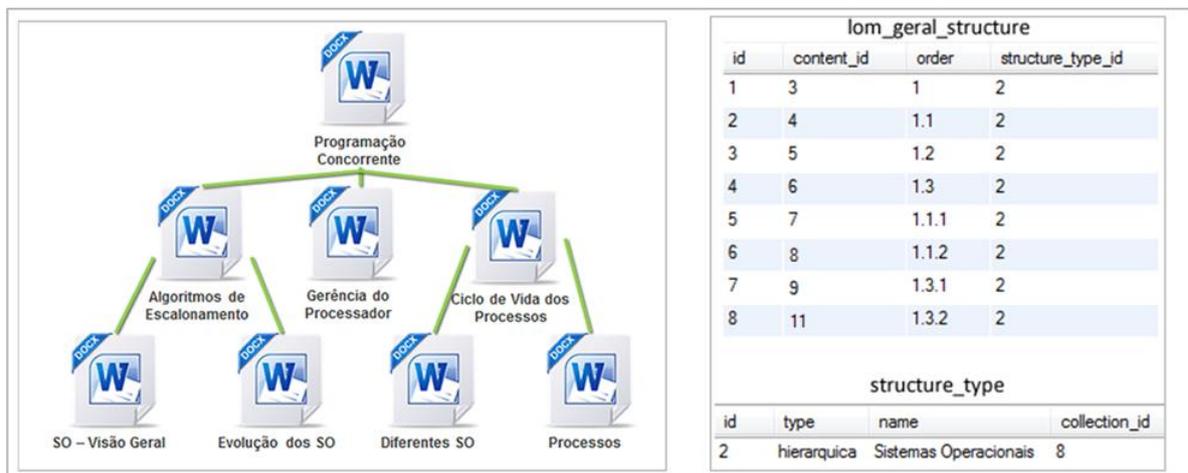


Figura 40 - Estrutura (Hierárquica)

Caso a estrutura escolhida seja “Linear”, os conteúdos serão ordenados em formato de sumário (Figura 41).

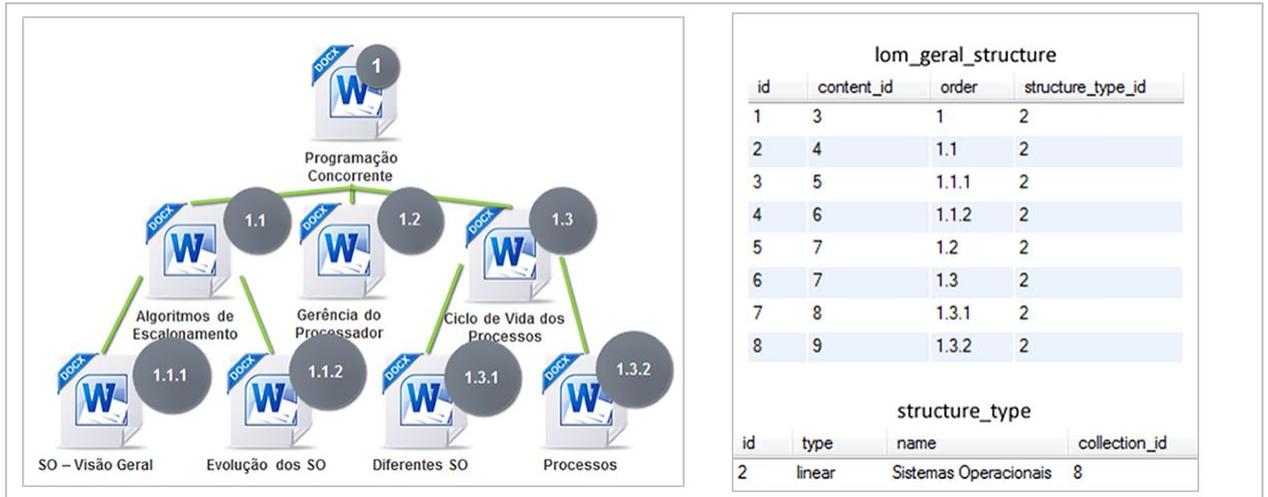


Figura 41 - Estrutura (Linear)

4.3.2. Anotação

A categoria Anotação tem como principal objetivo permitir que um usuário faça uma anotação/observação sobre um determinado conteúdo. A sua estrutura pode ser observada na imagem a seguir (Figura 42).

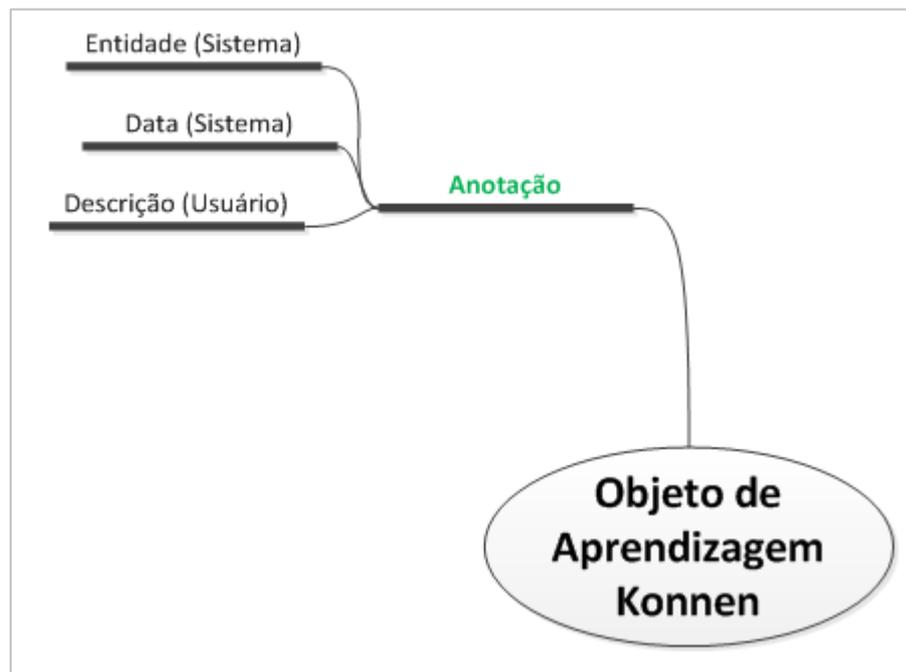


Figura 42 - Metadado de Objeto de Aprendizagem (Anotação)

Essa estrutura (Figura 42) permite que um usuário compartilhe um conteúdo que

possua uma anotação, de forma que o destinatário possa exibi-la, o que possibilita a interação e a colaboração. Assim, o destinatário fica susceptível a abstrair as informações contidas e até mesmo cooperar com a anotação. O trecho de código a seguir apresenta como é feita a anotação de um conteúdo (Figura 43).

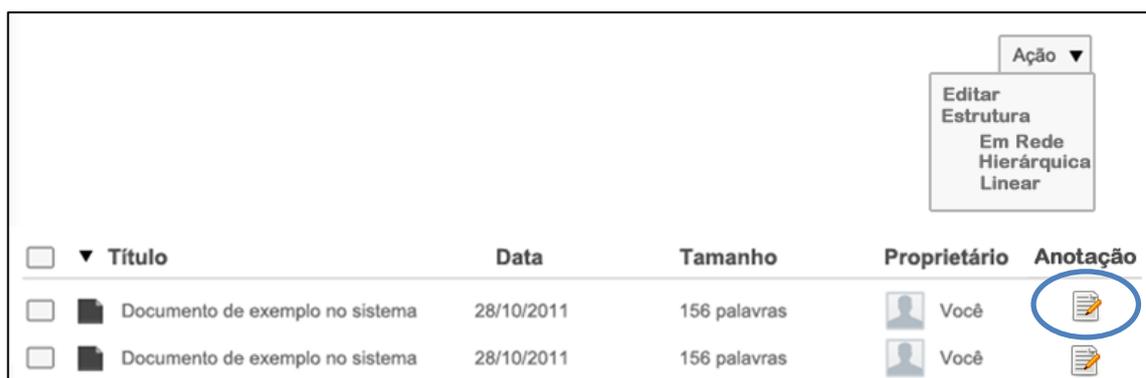
```

1 public function action_annotacao() {
2     $description = isset($_POST['description']) ? $_POST['description'] : "";
3     $content_id = isset($_POST['content_id']) ? $_POST['content_id'] : "";
4     $user = auth::instance()->get_user();
5
6     Manager_Content::save_lom_annotacao($user->id, $content_id, $description);
7     $request_url = $this->request->referrer();
8     $this->request->redirect($request_url);
9 }

```

Figura 43 - Action Anotacao do Controller Content

Conforme apresentado na Figura 43, nas Linhas 2 e 3 as variáveis instanciadas (`$description` e `$content_id`) armazenam os valores das posições “description” e “content_id” do array `$_POST`. Em seguida, é realizada uma verificação se as posições não estão vazias; caso as posições não estejam vazias, é armazenado o identificador do conteúdo e a descrição feita pelo usuário. Na linha 6 o *controller* invoca o método `save_lom_annotacao` do *manager*. A Figura a seguir apresenta a tela que informa como é feita uma anotação.



The screenshot shows a web interface for content management. At the top right, there is a dropdown menu labeled 'Ação' with options: 'Editar Estrutura', 'Em Rede', 'Hierárquica', and 'Linear'. Below this is a table with the following columns: 'Título', 'Data', 'Tamanho', 'Proprietário', and 'Anotação'. The table contains two rows of content items, both titled 'Documento de exemplo no sistema' with a date of '28/10/2011' and a size of '156 palavras'. The owner for both is 'Você'. The 'Anotação' column for the first row has a blue circle around the annotation icon (a document with a pencil).

<input type="checkbox"/>	▼ Título	Data	Tamanho	Proprietário	Anotação
<input type="checkbox"/>	 Documento de exemplo no sistema	28/10/2011	156 palavras	 Você	
<input type="checkbox"/>	 Documento de exemplo no sistema	28/10/2011	156 palavras	 Você	

Figura 44 - Parte da Tela do Gerenciador de Conteúdo (Anotação)

Como pode ser observado na Figura 44, a tela que lista os conteúdos do usuário também apresenta o ícone referente a ação de anotação. Caso ele já tenha feito uma anotação, após clicar novamente no ícone será carregada a anotação feita anteriormente, podendo alterar

o conteúdo, se necessário.

4.3.3. Ciclo de Vida

A categoria Ciclo de Vida tem como principal objetivo acompanhar as versões dos conteúdos, como também armazenar os usuários que contribuíram para a criação dos mesmos. A sua estrutura pode ser observada na Figura 45 a seguir.

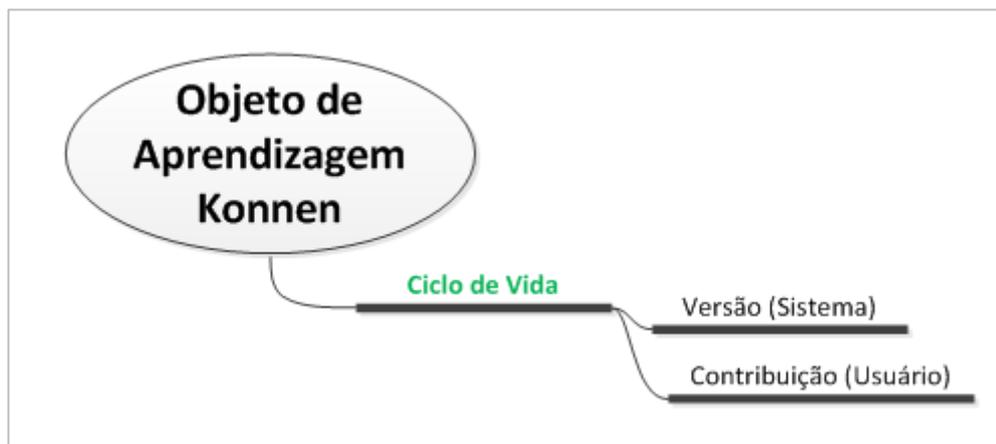


Figura 45 - Metadado de Objeto de Aprendizagem (Ciclo de Vida)

Para definir o ciclo de vida (Figura 45), o elemento versão é armazenado no momento em que um usuário cria um conteúdo, com a atribuição do valor: versão 1. O valor da versão será incrementado caso o usuário necessite editar um conteúdo. O trecho de código a seguir (Figura 45) apresenta como é feito esse processo.

```

1 public static function save($id, $user_id, $name, $type, $description,
2   $rshare_configuration, $visibility, $url, $fileondisk_id,
3   $text, $group_id = NULL) {
4   if ($id) {
5     $old_content = ORM::factory('content', $id);
6     $content->version = $old_content->version + 1;
7     $content->version_identifier = $old_content
8       ->version_identifier;
9   }
10  else {
11    $content->version = 0;
12    $content->version_identifier = md5($content
13      ->name . $content->date . $content->text
14      . $content->description);
15  }
16  $content->save();
17  return $content;
18 }

```

Figura 46 - Método Save do Manager Content

Como pode ser observado na Figura 46, das linhas 1 a 3 é realizada a declaração do método `save()` com seus respectivos parâmetros (`$id`, `$user_id`, `$name`, `$type`, `$description`, `$rshare_configuration`, `$visibility`, `$url`, `$fileondisk_id`, `$text`, `$group_id = NULL`); na linha 4 é verificado se a variável `$id` é nula, caso seja entende-se que está editando um conteúdo e que será incrementada uma nova versão (linhas 5 à 8). Caso a variável não seja nula (linha 10) entende-se que está criando um novo conteúdo e será atribuída a versão 0 para o mesmo (linhas 11 à 14).

A tela a seguir (Figura 47) apresenta parte do formulário da Tela de Editor de Conteúdo Completo responsável por adicionar os usuários que contribuíram na criação do conteúdo.

Ciclo de Vida

Contribuição:

Usuários que Contribuíram

404	douglas@ceulp.edu.br
	Diego Oliveira diegopso2@gmail.com
	Douglas Brito douglas44@gmail.com

Figura 47 - Parte da Tela do Editor de Conteúdo Completo (Ciclo de Vida)

Como pode ser analisado na Figura 47, ao digitar o nome de um usuário no campo “contribuição”, o sistema fornece uma lista contendo os usuários correspondentes à palavra que foi inserida. Caso o usuário desejado esteja na lista, o mesmo é selecionado e será inserido ao lado em: “Usuários que Contribuíram”. Por exemplo, se um professor estiver postando um conteúdo do tipo arquivo e esse conteúdo for uma apostila de Lógica de Predicados que foi desenvolvida por ele e por um determinado aluno, o aluno pode ser adicionado como um dos autores do conteúdo.

4.3.4. Técnico e Direito

A categoria Técnico tem com principal objetivo armazenar informações técnicas a respeito de um conteúdo, por exemplo, o formato, tamanho, procedimento etc. A sua estrutura pode ser observada na imagem a seguir (Figura 48).

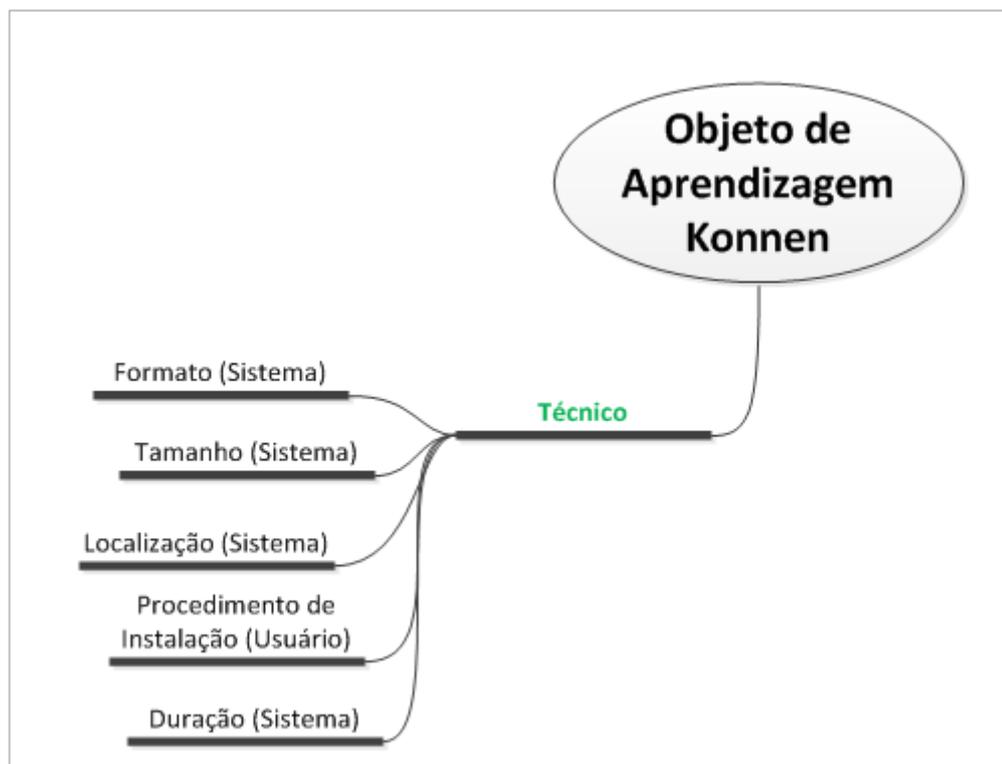


Figura 48 - Metadado de Objeto de Aprendizagem (Técnico)

No que tange às informações técnicas (Figura 48), para o armazenamento do Formato do conteúdo é necessário verificar o tipo do conteúdo a ser postado. Por exemplo, se o conteúdo for do tipo texto será armazenado o valor “text/html”; se o tipo for imagem armazenará

“imagem/extensão da imagem (png, jpg, gif, etc)”]; caso seja vídeo, o tipo mime será ”vídeo/link”; se o tipo escolhido for link armazenará “url/link”; e, por fim, se for um arquivo armazenará “file/extensão do arquivo (rar, pdf, doc, xls, etc)”. O tamanho dos conteúdos do tipo imagem e arquivo são recuperados no momento em que eles são salvos em disco pela função `filesize` do PHP. Para saber o tamanho de um conteúdo do tipo texto é verificada a quantidade de caracteres a serem postados e, a partir disso, é realizada a conversão em *bytes*. Para os conteúdos do tipo *link* e vídeo não é armazenado o tamanho do arquivo, pois eles são disponibilizados externamente. O procedimento de instalação é informado pelo usuário em um `textarea` na tela do editor de conteúdo completo. O mesmo caso acontece para o elemento descrição da categoria Direito (informado pelo usuário em um `textarea` na tela do editor de conteúdo completo).

4.3.5. Educacional

A categoria Educacional tem com principal objetivo armazenar informações pedagógicas do Objeto de Aprendizagem. A sua estrutura pode ser observada na imagem a seguir (Figura 49).

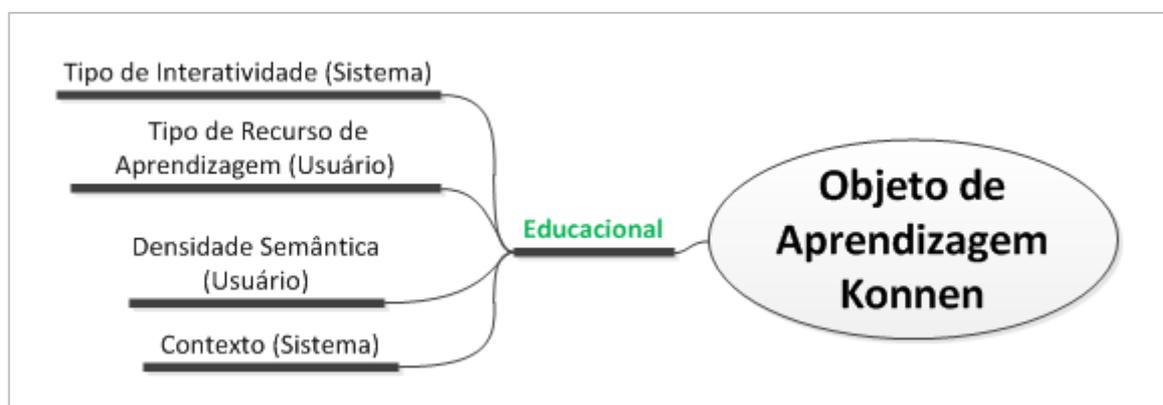


Figura 49 - Metadado de Objeto de Aprendizagem (Educacional)

Como pode ser observada na Figura 49, a categoria Educacional é composta por quatro elementos:

- Tipo de Interatividade: Devido ao fato do módulo de Atividades da Rede Social Acadêmica ainda não ter sido implementado não é possível verificar se o tipo de interatividade de um conteúdo é ativo, expositivo ou misto. Todos os conteúdos que são cadastrados a partir do módulo *Content* faz parte da categoria expositiva.;
- Tipo de Recurso de Aprendizagem: Se o conteúdo a ser cadastro faz parte de algo tipo

(Questionário, Diagrama, Experimento, Tabela, Apostila, Artigo, Tese/Dissertação, entre outros);

- Densidade Semântica: 1 para conteúdo menos denso e mais simples - 5 para conteúdo mais denso e complexo;
- Contexto: Armazena o local onde o conteúdo foi publicado.

A tela a seguir (Figura 50) mostra parte do formulário da Tela de Editor de Conteúdo Completo responsável por adicionar informações a respeito da categoria Educacional.

Educacional

Densidade Semântica:
 1 2 3 4 5

- 1 para conteúdo menos denso e mais simples - 5 para conteúdo mais denso e complexo.

Tipo de Recurso de Aprendizagem:

- Questionário
- Diagrama
- Experimento
- Tabela
- Apostila
- Artigo
- Tese/Dissertação

Figura 50- Parte da Tela do Editor de Conteúdo Completo (Educacional)

Como pode ser observado na Figura 50, na parte Educacional informará o valor da Densidade Semântica do conteúdo a ser criado e também informará o Tipo de Recurso de Aprendizagem.

4.4 Ontologia no modelo LOM+OA

Os Metadados de Objeto de Aprendizagem criados com embasamento no LOM são responsáveis por descreverem aspectos relevantes no que tange os recursos de aprendizagem. Porém, estes aspectos não são suficientes para delinear o domínio do sistema. Diante desse contexto, foi definido que a ontologia existirá para que seja possível fazer com que a representação do domínio encontre aspectos gerais de parte do universo do ensino, que contém usuários, cursos, disciplinas e turmas, e a criação de objetos de aprendizagem. A Estrutura criada para representar o domínio pode ser vista na Figura 51.

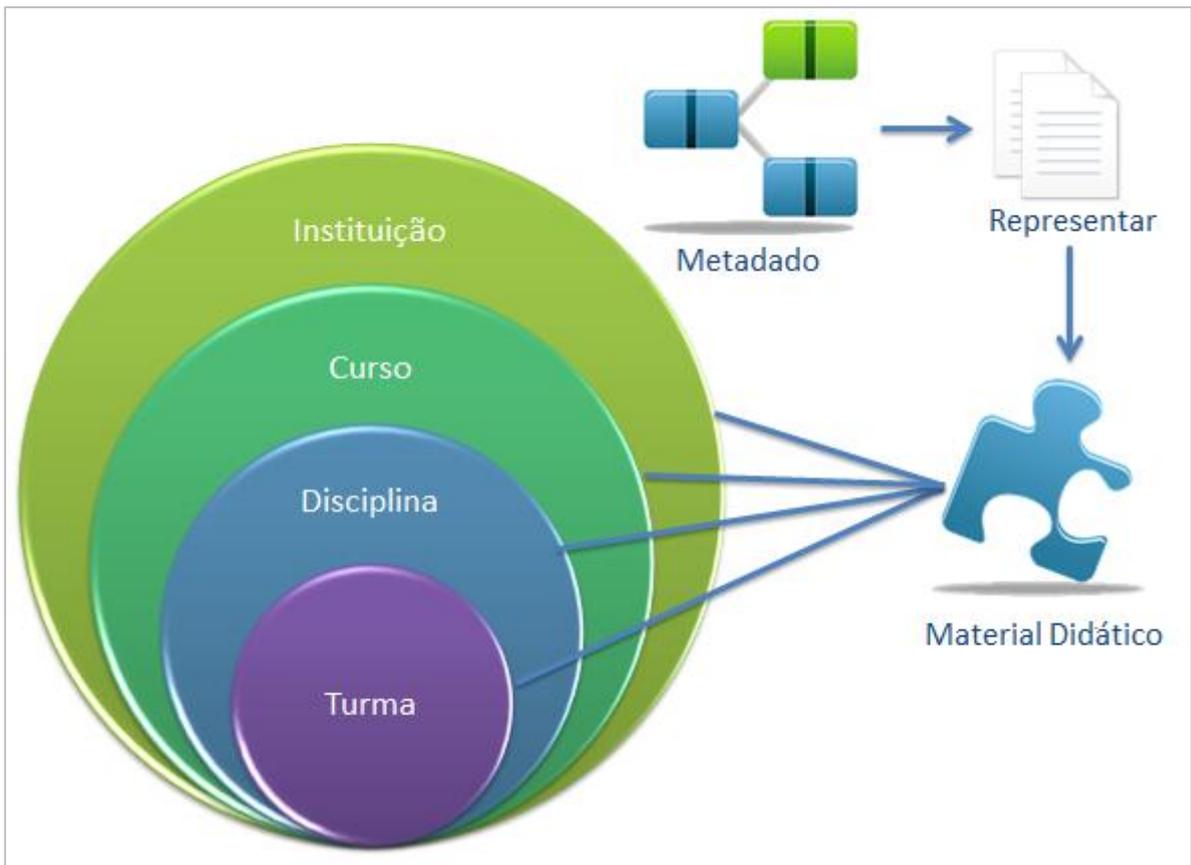


Figura 51 - Estrutura

Utilizando o conceito de Ontologia apresentado anteriormente (conceito – seção 2.2), a 4-tupla $\langle C,R,I,A \rangle$ permite que a Ontologia exista de forma abstrata no domínio. A partir dessa definição é possível identificar cada elemento (4-tupla) no domínio (Figura 51). Aplicando este conceito no contexto “material didático de uma rede social acadêmica”, tem-se o seguinte modelo:

- **C** - o conjunto de Conceitos, que são as classes (Instituição, Curso, Disciplina e Turma);
- **R** - o conjunto de Relações existente entre as classes (instituição \supset curso, curso \supset disciplina e disciplina \supset turma);
- **I** - o conjunto de instâncias, como exemplo, um Material Didático (vídeo);
- **A** - o conjunto de Axiomas, por exemplo, todo objeto relacionado a mais de uma disciplina é equivalente a um objeto interdisciplinar, logo é inferida a relação de “interdisciplinaridade”.

A seguir será apresentado como, de fato, a ontologia existe no domínio a partir de uma exemplificação de cada elemento da 4-tupla $\langle C,R,I,A \rangle$. O conjunto de conceito $\langle C \rangle$ passa a ser criado no momento em que são adicionados grupos na rede social acadêmica, Konnen. Para os grupos são atribuídos tipos (instituição, curso, disciplina e turma). A imagem a seguir (Figura 52) apresenta exemplos de grupos criados.

id	name	description	date	group_type_id	super_group_id
4	meu grupo	NULL	1326402193	0	NULL
17	Paraíso do Tocantins	NULL	1331584543	4	20
18	CEULP/ULBRA	ceulp	1331640827	0	NULL
19	CAIXA Paraíso		1331646042	4	20
20	Sistemas de Informação	Curso de Sistemas	1339101911	5	NULL
21	Lógica de Predicados	Fomalização de argumento...	1339101980	4	20
22	0309	Turma 2012/1 - Lógica	1339102066	6	NULL
23	Estruturas de Dados I	Representação, manipulaç...	1339679149	4	20

Figura 52 - SELECT * FROM na Tabela Group (1)

O select realizado na tabela Group lista todos os grupos cadastrados no sistema. Como pode ser observado, o grupo com id “20” (Sistemas de Informação) armazena o id “5” na coluna group_type_id, isso significa que todos os grupos que receberem esse identificador serão do tipo “curso”. Já os grupos que receberem o identificador “4” serão do tipo “disciplina” e os que receberem “6” do tipo turma. A relação entre esses grupos será apresentada na Figura 53.

id	name	description	date	group_type_id	super_group_id
4	meu grupo	NULL	1326402193	0	NULL
17	Paraíso do Tocantins	NULL	1331584543	4	20
18	CEULP/ULBRA	ceulp	1331640827	0	NULL
19	CAIXA Paraíso		1331646042	4	20
20	Sistemas de Informação	Curso de Sistemas	1339101911	5	NULL
21	Lógica de Predicados	Fomalização de argumento...	1339101980	4	20
22	0309	Turma 2012/1 - Lógica	1339102066	6	NULL
23	Estruturas de Dados I	Representação, manipulaç...	1339679149	4	20

Figura 53 - SELECT * FROM na Tabela Group (2)

A partir do momento que uma disciplina é criada, deve-se vinculá-la a um curso. A relação <R> que se forma pode ser visualizada na coluna `super_group_id` da Figura 53. Essa coluna referencia a própria tabela `Group`, logo as disciplinas receberão o `id` do curso a qual elas pertencem. Por exemplo, as disciplinas Lógica de Predicados (`id = 21`) e Estruturas de Dados I (`id = 23`) pertencem ao curso de Sistemas de Informação (`id = 20`). O elemento instância da 4-tupla <I> passa a existir no momento em que um usuário cria um conteúdo. Isso pode ser evidenciado na Figura 54.

id	name	user_id	date	type	description
158	LOM	4	1339164383	text	Na Figura 8 são apresentadas as categorias ...
161	Version 1.3 Public Draft	4	1339165204	image	Resolves the drift between the IMS Meta-dat...
162	LOM - Konnen	4	1339165577	image	Metadado de Objeto de Aprendizagem do Ko...
163	Ontologia utilizando Protegé	4	1339166081	video	Tutorial utilizando o Protegé
165	Ontologia	4	1339166297	link	Tudo sobre ontologia
167	Ontologia	4	1339167695	arquivo	Usando ontologia
168	Teste do LOM	4	1339170166	arquivo	Teste de implementação

Figura 54 - SELECT * FROM na Tabela Content

Cada linha da tabela apresentada na Figura 54 representa uma instância de um conteúdo.

Por fim, tem-se o elemento <A> da estrutura de ontologia, que existirá por meio de inferências estabelecidas no domínio. A seção a seguir explicará o processo dos axiomas inferidos no sistema.

4.4.1. Axioma

Os axiomas definidos nesta seção foram estabelecidos em reuniões com a especialista do domínio. A partir de pesquisas realizadas na internet sobre tal temática (com o auxílio da especialista), percebeu-se que há constantes revisões conceituais dos termos “interdisciplinaridade”, “multidisciplinaridade” e “transversalidade” usados nesse trabalho. Assim, optou-se, por sugestão da especialista, em adotar uma característica de cada conceito como a verdade que permearia o axioma. Isso pode ser revisto em versões futuras dos axiomas da ontologia.

Desta forma, tem-se que, no momento em que um material é vinculado a mais de uma

disciplina, é inferida a relação de “interdisciplinaridade”, pois na ontologia há um axioma que define essa relação. O diagrama a seguir exemplifica esse axioma.

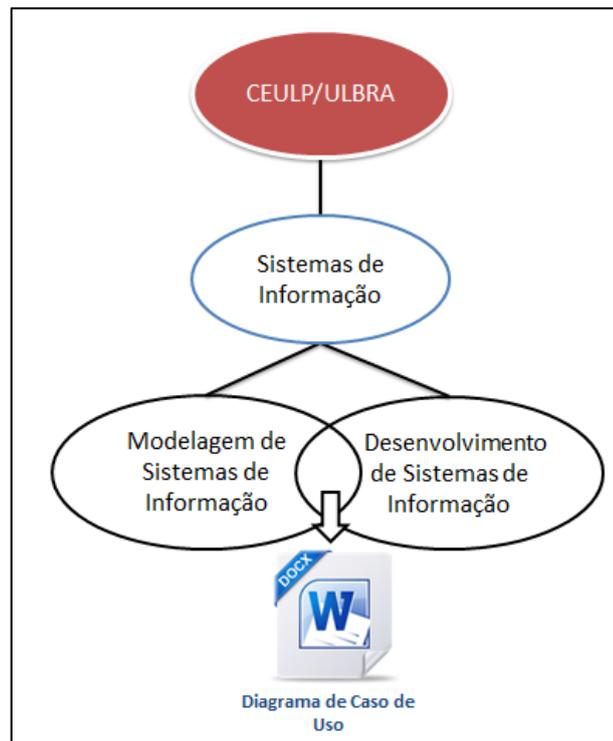


Figura 55 - Diagrama que define a relação Interdisciplinaridade

No que se refere à relação de interdisciplinaridade, a Figura 55 apresenta o seguinte cenário:

- Conceito: instituição = ceulp/ulbra, curso = sistemas de informação, disciplina = modelagem de sistemas de informação, disciplina = desenvolvimento de sistemas de informação;
- Relação: ceulp/ulbra \supset sistemas de informação, sistemas de informação \supset modelagem de sistemas de informação e sistemas de informação \supset desenvolvimento de sistemas de informação;
- Instância: Conteúdo do tipo arquivo = Diagrama de Caso de Uso;
- Axioma: Pode-se inferir que todo objeto (Diagrama de Caso de Uso) relacionado a mais de uma disciplina é equivalente a um objeto interdisciplinar, assim tem-se a interdisciplinaridade, por exemplo, quando há uma intersecção entre duas disciplinas (modelagem de sistemas de informação e desenvolvimento de sistemas de informação) de um mesmo curso (sistemas de informação) .

A seguir é apresentada a *View* que retorna esse axioma.

```

1  SELECT a.*
2  FROM (SELECT gd.gd_id,
3           gd.gd_name,
4           gd.super_group_id,
5           gc.gc_id,
6           gc.gc_name,
7           vcr.id          AS content_id,
8           vcr.content_name AS content_name
9           FROM (SELECT g.id AS gd_id,
10              g.name AS gd_name,
11              g.super_group_id
12             FROM groups g
13              INNER JOIN group_types gt
14                 ON g.group_type_id = gt.id
15                 AND gt.name = 'discipline') gd
16          INNER JOIN (SELECT g.id AS gc_id,
17              g.name AS gc_name
18             FROM groups g
19              INNER JOIN group_types gt
20                 ON g.group_type_id = gt.id
21                 AND gt.name = 'course') gc
22             ON gd.super_group_id = gc.gc_id
23          INNER JOIN view_content_recipients vcr
24             ON gd.gd_id = vcr.recipient_id
25             AND vcr.recipient_type = 'group') a
26  INNER JOIN (SELECT Count(gd.gd_id) AS count,
27              vcr.content_name,
28              vcr.id          AS content_id,
29              gc.gc_name
30             FROM (SELECT g.id AS gd_id,
31              g.name AS gd_name,
32              g.super_group_id
33             FROM groups g
34              INNER JOIN group_types gt
35                 ON g.group_type_id = gt.id
36                 AND gt.name = 'discipline') gd
37          INNER JOIN (SELECT g.id AS gc_id,
38              g.name AS gc_name
39             FROM groups g
40              INNER JOIN group_types gt
41                 ON g.group_type_id = gt.id
42                 AND gt.name = 'course')
43             gc
44             ON gd.super_group_id = gc.gc_id
45          INNER JOIN view_content_recipients vcr
46             ON gd.gd_id = vcr.recipient_id
47             AND vcr.recipient_type = 'group'
48          GROUP BY gc.gc_id,
49                 vcr.id) b
50  ON a.content_id = b.content_id
51  AND count > 1

```

Figura 56 - View Axioma (Interdisciplinaridade)

A consulta apresentada na Figura 56 é dividida em duas etapas: das Linhas 9 a 25 são

selecionadas todas as informações de conteúdos que foram compartilhados em disciplinas e os cursos dessas disciplinas (atribuído “A” para o mapeamento dessa consulta - Linha 25). Na segunda etapa (Linhas 26 a 47) é contada a quantidade de disciplinas nas quais um conteúdo foi compartilhado – dividido por curso (atribuído “B” para o mapeamento dessa consulta – Linha 49). Das linhas 48 a 51 é feita uma junção de “A e B” quando o *id* do conteúdo de “A” for igual ao *id* do conteúdo de “B” e logo depois é verificado na função *count* se a quantidade de disciplinas de um mesmo curso que está usando o mesmo conteúdo é maior que um.

Em uma outra situação, tem-se que, no momento em que um material é vinculado a mais de uma disciplina de cursos distintos, é inferida a relação de “multidisciplinaridade”, que se refere à ideia de duas ou mais disciplinas compartilharem um mesmo conteúdo, sem necessariamente integrar resultados ou seguir metodologias semelhantes.. Na figura 57 é apresentada uma exemplificação dessa relação: todo objeto relacionado a mais de um curso é equivalente a um objeto multidisciplinar.

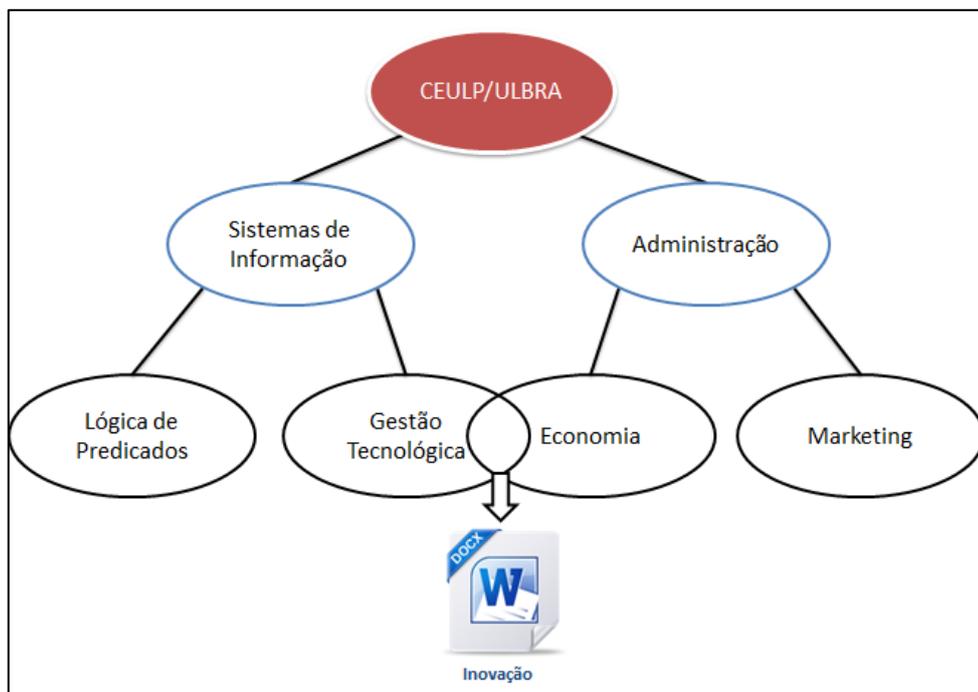


Figura 57 - Diagrama que define a relação Multidisciplinaridade

No tange à relação de multidisciplinaridade (Figura 57), é importante existir o seguinte cenário:

- Conceito: instituição = ceulp/ulbra, curso = sistemas de informação, curso = administração, disciplina = lógica de predicados, disciplina = gestão tecnológica, disciplina = economia, disciplina = marketing;

- Relação: ceulp/ulbra \supset sistemas de informação, ceulp/ulbra \supset administração, sistemas de informação \supset lógica de predicados, sistemas de informação \supset gestão tecnológica, administração \supset economia, administração \supset marketing ;
- Instancia: Conteúdo do tipo arquivo = Inovação;
- Axioma: Pode-se inferir que todo objeto (Inovação) relacionado a mais de uma disciplina é equivalente a um objeto multidisciplinar quando há uma intersecção entre duas disciplinas (gestão tecnológica e economia) de cursos distintos (sistemas de informação e administração).

A seguir é apresentada a *View* que retorna esse axioma.

```

1  SELECT a.*
2  FROM (SELECT gd.gd_id,
3         gd.gd_name,
4         gd.super_group_id,
5         gc.gc_id,
6         gc.gc_name,
7         vcr.id AS content_id,
8         vcr.content_name AS content_name
9  FROM (SELECT g.id AS gd_id,
10         g.name AS gd_name,
11         g.super_group_id
12        FROM groups g
13         INNER JOIN group_types gt
14          ON g.group_type_id = gt.id
15          AND gt.name = 'discipline') gd
16        INNER JOIN (SELECT g.id AS gc_id,
17         g.name AS gc_name
18        FROM groups g
19         INNER JOIN group_types gt
20          ON g.group_type_id = gt.id
21          AND gt.name = 'course') gc
22         ON gd.super_group_id = gc.gc_id
23        INNER JOIN view_content_recipients vcr
24         ON gd.gd_id = vcr.recipient_id
25         AND vcr.recipient_type = 'group') a
26        INNER JOIN (SELECT Count(DISTINCT gc.gc_id) AS count,
27         vcr.content_name,
28         vcr.id AS content_id,
29         gc.gc_name
30        FROM (SELECT g.id AS gd_id,
31         g.name AS gd_name,
32         g.super_group_id
33        FROM groups g
34         INNER JOIN group_types gt
35          ON g.group_type_id = gt.id
36          AND gt.name = 'discipline') gd
37        INNER JOIN (SELECT g.id AS gc_id,
38         g.name AS gc_name
39        FROM groups g
40         INNER JOIN group_types gt
41          ON g.group_type_id = gt.id
42          AND gt.name = 'course')
43        gc
44         ON gd.super_group_id = gc.gc_id
45        INNER JOIN view_content_recipients vcr
46         ON gd.gd_id = vcr.recipient_id
47         AND vcr.recipient_type = 'group'
48        GROUP BY vcr.content_id) b
49  ON a.content_id = b.content_id
50  AND count > 1

```

Figura 58 - View Axioma (Multidisciplinaridade)

A consulta apresentada na Figura 58 também é dividida em duas etapas: a primeira etapa faz o mesmo mapeamento (A) realizado na *View* axioma (Interdisciplinaridade) (Figura 56) “A”; já a etapa atribuído o nome “B” para o mapeamento conta a quantidade de disciplinas de cursos distintos que utilizam um mesmo conteúdo (Linhas 26 a 47). Nas linhas

48 e 49 são agrupados os conteúdos e contados quando os cursos quem contêm as disciplinas que utilizam um mesmo conteúdo forem distintos. Após isso é verificado se essa quantidade retornada é maior que um (Linha 50).

Em outro cenário, verifica-se que no momento em que um material é vinculado a mais de uma disciplina de um mesmo curso, totalizando em no mínimo 70% das disciplinas (porcentagem usada como teste e sugerida pela especialista do domínio), é inferida a relação de “transversalidade”, pois na ontologia há o seguinte axioma que define essa relação.

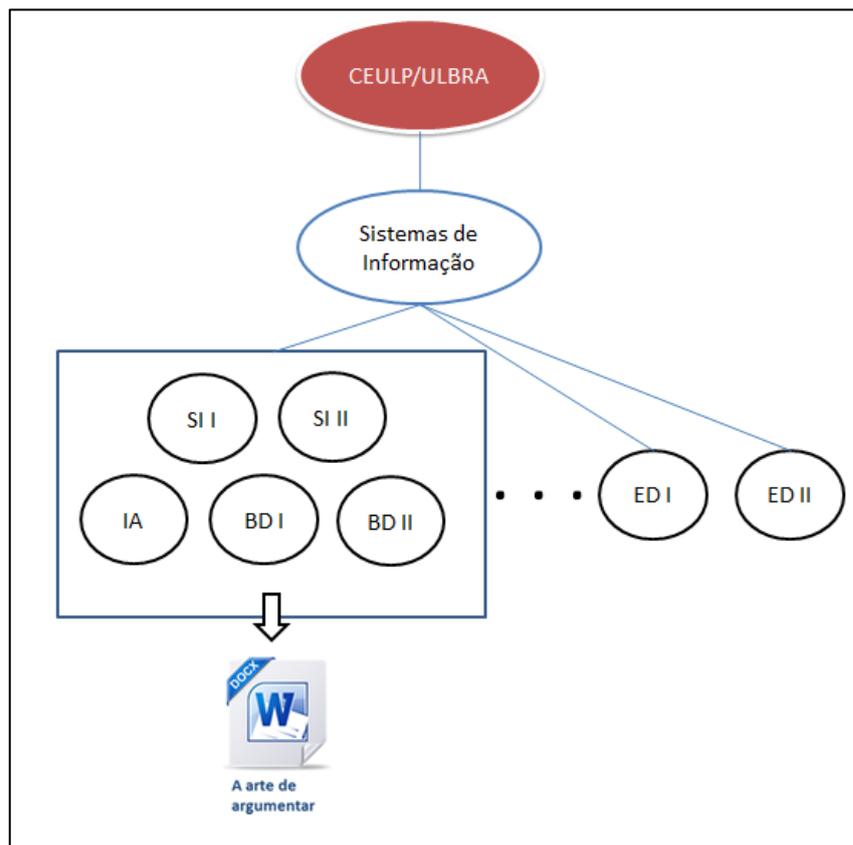


Figura 59 - Diagrama que define a relação Transversalidade

Para demonstrar esse tipo de relação (Figura 59), deve-se existir o seguinte cenário:

- Conceito: instituição = ceulp/ulbra, curso = sistemas de informação, disciplina = gerência de projetos, disciplina = SI II, disciplina = IA, disciplina = banco de dados;
- Relação: ceulp/ulbra \supset sistemas de informação, sistemas de informação \supset gerência de projetos, sistemas de informação \supset SI II, sistemas de informação \supset IA, sistemas de informação \supset banco de dados;
- Instância: Conteúdo do tipo arquivo = A arte de Argumentar ;
- Axioma: Pode-se inferir que todo objeto (A arte de Argumentar) relacionado a mais de

uma disciplina de um mesmo curso, em 70% das disciplinas, é equivalente a um objeto transversal quando há intersecção entre mais de 70% das disciplinas (SI I, SI II, IA, banco de dados) de um mesmo curso (sistemas de informação).

A seguir é apresentada parte da *View* que retorna esse axioma.

```

52         INNER JOIN (SELECT Count(DISTINCT gd.gd_id) AS count,
53         gc.gc_name, gc.gc_id
54     FROM (SELECT g.id AS gd_id,
55           g.name AS gd_name,
56           g.super_group_id
57     FROM   groups g
58           INNER JOIN group_types gt
59               ON g.group_type_id = gt.id
60               AND gt.name = 'discipline') gd
61     INNER JOIN (SELECT g.id AS gc_id,
62           g.name AS gc_name
63     FROM   groups g
64           INNER JOIN group_types gt
65               ON g.group_type_id = gt.id
66               AND gt.name = 'course')
67           gc
68           ON gd.super_group_id = gc.gc_id
69     GROUP BY gc.gc_id) c
70     ON b.gc_id = c.gc_id
71
72 WHERE b.count / c.count > 0.7

```

Figura 60 - View Axioma (Transversalidade)

Como observado no código SQL apresentado na Figura 60, essa *View* é dividida em três etapas, sendo a primeira e a segunda (mapeamento A e B) iguais a da *View* axioma (Interdisciplinaridade) (Figura 56). Já na terceira etapa é agrupada todas as disciplinas de um curso (Mapeamento C - Linhas 52 a 70) enquanto a quantidade de disciplinas nas quais um conteúdo foi compartilhado dividido por C (disciplinas de um curso) seja maior que 70%.

Através das relações que envolvam desde relacionamentos simples, estabelecidos por propriedades (instituição \supset curso, curso \supset disciplina e disciplina \supset turma), como relações complexas, construídas a partir da definição de axiomas (Interdisciplinaridade, Multidisciplinaridade e Transversalidade), será possível por meio da ontologia, obter uma base de informações que garante a facilidades e flexibilidade no que diz respeito ao alcance dos valores e consultas envolvendo múltiplos elementos.

5. CONSIDERAÇÕES FINAIS

A utilização de conteúdos educacionais digitais torna-se cada vez mais necessária para estimular o estudo autônomo dos alunos. Além disso, a forma como os conteúdos são disponibilizados tem um importante papel nesse contexto. Atualmente, o acréscimo contínuo de tecnologias que utilizam um contexto interativo e colaborativo na *web* faz com que sistemas de gerenciamento de conteúdos, que são ferramentas que permitem a gestão e disseminação da informação, sejam bem difundidos, dadas as características de flexibilidade e praticidade que apresentam.

Um dos pontos para o qual este trabalho foi direcionado consistiu na definição dos metadados definidores dos objetos de aprendizagem, de forma a alicerçar uma ontologia de materiais didáticos de uma rede social acadêmica. Com a utilização de objetos de aprendizagem a partir de uma ontologia, foi possível ter um entendimento amplo das propriedades e características das classes e relacionamentos existentes em um contexto, podendo realizar inferências sobre aspectos relevantes que auxiliarão no processo de ensino-aprendizagem.

Desta forma, foi realizado um estudo sobre os elementos que compõem o LOM. A partir deste estudo foi possível compreender como funciona a sua estrutura e criar metadados de objetos de aprendizagem relativos ao contexto da rede social acadêmica, Konnen. Também foram estudados os conceitos de Ontologia e sua utilização. A Ontologia permitiu que os conteúdos fossem contextualizados, tornando possível através da utilização da 4-trupla a geração de inferências, que para o presente trabalho, foram centradas na relação de interdisciplinaridade, multidisciplinaridade e transversalidade de um conteúdo.

A partir do desenvolvimento do trabalho foi possível notar a importância de um especialista de domínio para definição da Ontologia e seus axiomas, pois esse módulo apresenta um alto nível de complexidade. Sobre essa temática, compreendeu-se, também, que os axiomas definidos servem como subsídios para o entendimento da estruturação desse conceito, mas ainda é necessário uma reflexão maior sobre os elementos que compõem o domínio para se ter conhecimento suficiente para a definição de outros axiomas. Para tanto, é necessário que haja uma intersecção de profissionais da área de educação e computação, pois

o conhecimento que pode ser abstraído do domínio requer um entendimento aprofundado do processo de ensino-aprendizagem.

Uma das maiores dificuldades encontradas no trabalho foi a reimplantação do Gerenciador de Conteúdo, pois para realizar a mudança de tecnologia foi necessário ter o entendimento da linguagem PHP, para depois compreender o funcionamento do *framework* Kohana. Esse processo de entendimento, reestruturação, implementação, modelagem e testes consumiram bastante tempo na execução do projeto.

Este trabalho abre a possibilidade para diferentes formas de continuação. Dentre elas podem ser citadas:

- Definição e implementação de novos axiomas para o domínio. Criação de *Web Services* para o módulo *Content* tornar-se independente da linguagem;
- Criar técnicas de *Text Mining* para que a avaliação sobre a densidade semântica do objeto seja realizada automaticamente pelo sistema;
- Realização de pesquisas para o desenvolvimento de interfaces adaptativas, que considerem tanto os dados que compõem os objetos de aprendizagem e o domínio definido pela ontologia, como as preferências do usuário;
- Utilização da ontologia dos Objetos de Aprendizagem de forma a definir sequências de aprendizagem.

REFERÊNCIAS BIBLIOGRÁFICAS

BRITO, Douglas M. **Gerenciador de Conteúdos para um Rede Social Acadêmica**. 2011, 59 p. Relatório de Estágio – Sistemas de Informação, Centro Universitário Luterano de Palmas, CEULP/ULBRA, Palmas - TO.

DAVIES, John; STUDER, Rudi; WARREN, Paul. **Semantic Web Technologies: trends and research in ontology-based systems**. 1ª ed. England: John Wiley & Sons Ltd, 2006. 312 p.

FILATRO, Andrea. **Design Instrucional Contextualizado: educação e tecnologia**. 1ª ed. São Paulo: Editora Senac, 2003. 216 p.

FERRAS, Reinaldo. **Acessibilidade na Web: o caminho das pedras para construir sítios acessíveis**. Disponível em: <<http://www.w3c.br/palestras/2009/Apres-acessibilidade-na-web.pdf>> Acessado dia 25 de Setembro de 2011.

IEEE **Learning Technology Standards Committee (LTSC)** Draft Standard for Learning Object Metadata Version 6.1. 2001

IEEE , SA. **Learning Technology Standards Committee**. IEEE 1484.12.1-2002 Draft Standard for Learning Object Metadata, 2002.

FUJIL, Noemi P. N. **Uma Proposta de Objetos de Aprendizagem Reutilizáveis Adaptativos para o Ensino de Estatística**. 2006. 122 p. Dissertação (Mestrado em Ensino de Ciências e Matemática) - UNICSUL, São Paulo.

GAGNE, R., **The conditions of learning**, New York: Holt, Rhinehart and Winston, 1985.

GARCIA, Simone. **Objetos de Aprendizagem: investindo na mediação digital do conhecimento**. In: ENCONTRO DO CÍRCULO DE ESTUDOS LINGUÍSTICOS DO SUL, 7, 2006, Pelotas. **Anais...** Pelotas: CELSIL, 2006.

GOLDER, Scott A.; HUBERMAN, Bernardo A. The Structure of Collaborative Tagging System. **Journal of Information Science**, Thousand Oaks, v. 32 , n. 2, p. 198-208, 2006, Disponível em: <<http://www.hpl.hp.com/research/scl/papers/tags/tags.pdf>>. Acesso em: 13 nov. 2011.

GRUBER, Thomas R. Towards Principles for the Design of Ontologies Used for Knowledge Sharing. **International Journal Human-Computer Studies**. 1993, v. 43, p. 907-928, Padova, Itália. Disponível em: <<http://tomgruber.org/writing/onto-design.pdf>>. Acesso em: 21 abr. 2011.

JACOBSEN, Peter. Reusable Learning Objects- What does the future hold?. **e-learning Magazine**, p. 1. 2002.

KIM, sunha. **The Conceptualization, Utilization, Benefits and Adoption of Learning Objects**, 2009. 151 p. Dissertação de Doutorado (Psicologia) - Virginia Polytechnic Institute, Blacksburg, Virginia.

L'Allier J.J. **Frame of Reference: NETg's Map to the Products, Their Structure and Core Beliefs**. NetG Whitepaper, 1997.

LOM. **Learning Object Metadata**. Disponível em <<http://ltsc.ieee.org/wg12/>>. Acessado em 03 de Setembro de 2011.

MENDES, Rozi Mara; SOUZA, Vanessa Inávio; CAREGNATO, Sonia Elisa. A Propriedade Intelectual na Elaboração de Obejtos de Aprendizagem. IN: CIFORM-ENCONTRO NACIONAL DE CIÊNCIA DA INFORMAÇÃO. 2004, Bahia. **Anais...** Bahia: UFBA, 2004.

PRIETO-DÍAZ, Rubén. **Reuse as a New Paradigm for Software Development**. IN: INTERNATIONAL WORKSHOP ON SYSTEMATIC REUSE. 1996, Liverpool. **Anais...** Liverpool: Marjan Sarshar, 1996.

SILVEIRA, Ismar; ARAÚJO, Carlos F; AMARAL, Luiz H. OLIVEIRA, Ivan C. Granularity and Reusability of Learning Objects. In: _____. **Learning Objects and Instructional Design**. 1. ed. California: Santa Rosa, 2007. p. 139-170.

SANTANA, V. F; MELO-SOLARTE, D.S.; NERIS, V.P.A.; MIRANDA, L.C. and BARANUSKAS, M.C.C. Redes Sociais Online: Desafios e Possibilidades para o Contexto Brasileiro. In: CONGRESSO DA SOCIEDADE BRASILEIRA DE COMPUTAÇÃO, 29, 2009, Bento Gonçalves (RS). **Anais...** Bento Gonçalves: SBC, 2009.

SILVA, Edeilson M. **SWEETS**: um Sistema de Recomendação de Especialistas aplicado a Redes Sociais. 2009. 129 p. Dissertação (Mestrado em Ciência da Computação) - Ciência da Computação, Universidade Federal de Pernambuco, Recife - PE.

SOUZA, Jackson; BRITO, Parcilene; SOUSA, Cristina; SILVA, Edeilson; FAGUNDES, Fabiano; OLIVEIRA, Fernando; MARIOTI, Madianita; **Aprendizagem Organizacional Através de uma Rede de Gestão de Conhecimento**, 2012, Palmas. Projeto de Pesquisa... COPEX, CEULP/ULBRA.

STOJANOVIC, Ljiljana. **Methods and Tools for Ontology Evolution**. 2004. 249 p. Tese (Doutorado em Economia) - Faculdade de Economia Fidericiana, Universidade de Karlsruhe, Karlsruhe - Alemanha.

TERRA, José Cláudio. **Portais Corporativos e Gestão de Conteúdo**. Disponível em < <http://www.terraforum.com.br/biblioteca/Documents/libdoc00000020v002Portais%20Corporativos%20e%20Gestao%20de%20Conteudo%20.pdf> > acessado em 14 de Março de 2012.

VILAR, Bruno S. C. M. **Um Framework Para o Desenvolvimento de Sistemas Adaptativos a partir de Ontologias**. 2006. 147 p. Trabalho de Conclusão de Curso (Sistemas de Informação) - Centro Universitário Luterano de Palmas, Palmas, TO.

WANG, Xiaodan; FANG, Fang; FAN, Lei. Ontology-Based Description of Learning Object. In: ICWL, 2008, Berlin. **Anais...** Berlin: Springer-Verlag, 2008. p. 468-476.

WILEY, David. Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In:____. **The Instructional Use of Learning Objects**. Estados Unidos: Agency for Instructional Technology, 2002, cap. 1.1, p. 1-35..

WILEY, David. A. **Learning Object Design and Sequencing Theory**. 2000. 142 p. Dissertação (Mestrado em Filosofia) - Brigham Young University, Provo, Utah.

YONG-HE, Wu; ZHENG-HONG, Wu. A Survey of Application of Chinese e-Learning Technology Standards in Distance Education Colleges. IN: 17TH INTERNATIONAL CONFERENCE ON COMPUTERS IN EDUCATION. 2009, Hong Kong. **Anais...** Hong Kong: Asia-Pacific Society for Computers in Education, 2009.

YOUTUBE, **Aula de Teclado 3 Acordes facilitados por Cezar (Tutorial)**, 2008. Vídeo online (2:42 min), digital, widescreen, color, som. Disponível em < <http://www.youtube.com/watch?v=sQBZnL3sUZ4> > Acessado dia 28 de Outubro de 2011.

ZANETTE, Elisa Netto; NICOLEIT, Evânio Ramos; GIACOMAZZO Graziela Fátima. A produção do material didático no contexto cooperativo e colaborativo da disciplina de Cálculo Diferencial e Integral I, na modalidade de educação a distância, na graduação. In: VII CICLO DE PALESTRAS SOBRE NOVAS TECNOLOGIAS NA EDUCAÇÃO, 9, 2006, Porto Alegre. **Anais...** Porto Alegre: Revista Renote, 2006.

ANEXOS

Análise de Requisitos

Em uma das reuniões com pesquisadores do KONNEN realizou-se a coleta dos requisitos. Assim, foi possível identificar, especificar e descrever todos os requisitos do sistema. Nessa seção serão apresentados os requisitos coletados sobre o domínio em questão:

Tabela 12 - Requisito Gerenciar Conteúdo.

US1 - Gerenciar Conteúdo
R.1.1 Pesquisar conteúdo;
R.1.2 Excluir conteúdo;
R.1.3 Compartilhar conteúdo;
R.1.4 Fazer download;
R.1.5 Imprimir conteúdo;
R.1.6 Usar Editor de Texto

O requisito apresentado na Tabela 12 engloba os requisitos que poderão existir dentro de qualquer tipo de conteúdo.

Para cada conteúdo listado será disponibilizado um botões que realizam ações sobre esse conteúdo (editar, excluir, detalhes, compartilhar, entre outros.), poderá imprimir um conteúdo do tipo texto, fazer download de conteúdos do tipo arquivo e imagem.

US2 – Ver/Comentar Conteúdo

Após o usuário selecionar um conteúdo da “Dashboard/Timeline”, ele é exibido na íntegra em outra página. Nesta página é exibido inicialmente a foto do usuário criador do conteúdo e o título. Abaixo têm-se as seguintes informações: o nome do usuário criador, seguido da data e hora em que o conteúdo foi criado (se for um conteúdo postado em uma comunidade, após a data deve ser adicionada a expressão: “Via ‘nome da comunidade’”); em seguida está todo o restante da postagem, e ao final são apresentados os comentários que esse conteúdo recebeu bem como uma área para inserir um novo comentário. Dos comentários são exibidos: data que foi feito, foto do usuário que comentou, nome do usuário (com link para o perfil desta pessoa), e o comentário

US3 – Criar Conteúdo

O usuário terá duas opções para criar um conteúdo, criar um conteúdo rápido pela página inicial ou um conteúdo mais elaborado pela página de gerenciamento de conteúdo, logo em seguida deverá escolher o tipo de conteúdo que será criado, e automaticamente o sistema o redirecionará para a página de criação. A seguir serão listados os requisitos funcionais e não funcionais

Tabela 13 - Requisitos Funcionais- Criar Conteúdo.

#	Descrição
US3-RF1	O sistema deve permitir que o usuário escolha o tipo de conteúdo que ele deseja criar.
US3-RF2	O sistema deve permitir que o usuário crie um conteúdo.
US3-RF3	O sistema deve permitir que o usuário cancele a criação de um conteúdo.

Tabela 14 - Requisitos Não Funcionais - Criar Conteúdo.

#	Descrição
US3-RNF1	O sistema não deve permitir o cadastro de um conteúdo caso os dados digitados pelo usuário estejam incompletos ou forem inválidos.
US3-RNF2	O sistema deve fazer a autenticação utilizando HTTP Seguro (HTTPS)
US3-RNF3	O sistema deve fazer log das tentativas de logon do usuário

Regras de navegação:

- Usuário clica no tipo de conteúdo que deseja criar: sistema apresenta a tela de edição de conteúdo
- Usuário preenche os campos referentes ao conteúdo que será criado
- Usuário confirma a criação do conteúdo: sistema confirma a criação do conteúdo e redireciona o usuário para a tela inicial do gerenciador de conteúdos.

US4 – Inicial do Gerenciador de Conteúdos

Levando em consideração que o usuário esteja *logado* no sistema e que ele se encontra na página inicial do gerenciador de conteúdo, ele terá na barra de navegação do lado esquerdo

as opções de “Criar Conteúdo” e são apresentadas as coleções que este usuário criou ou as que estão compartilhadas com ele. No centro da tela são listados todos os conteúdos deste usuário (os criados por ele ou compartilhados com ele), apresentando para cada conteúdo seu nome, criador e a data que foi criado. O usuário poderá ordenar estes conteúdos da forma que desejar. O usuário poderá utilizar as configurações (editar, excluir, compartilhar, entre outras) que existem no sistema clicando nas configurações de cada conteúdo, estas são encontradas ao lado de cada um. Quando é selecionado um conteúdo são apresentadas suas informações. De acordo com o tipo de conteúdo, por exemplo, se o usuário selecionar um conteúdo do tipo imagem, serão apresentadas informações referentes ao criador, formato da imagem, dimensões, compartilhado com quem, data da criação e uma miniatura da imagem.

Tabela 15 - Requisitos Funcionais - Inicial Gerenciador de Conteúdos

#	Descrição
US4-RF1	O sistema deve permitir que o usuário escolha o tipo de conteúdo que ele deseja cadastrar.
US4-RF2	O sistema deve permitir que o usuário ordene os conteúdos da forma que desejar.
US4-RF3	O sistema deve permitir que o usuário realize ações (editar, excluir, compartilhar) em cada conteúdo que é listado.
US4-RF4	O sistema deve exibir as informações de um conteúdo que o usuário selecionou. Se o conteúdo for do tipo imagem, deverá apresentar uma miniatura da imagem ao lado das informações, isso também servirá para conteúdos do tipo vídeo e áudio que mostrará o player.

Regras de navegação;

- Usuário clica em criar conteúdo: o sistema exibe as opções de conteúdos que usuário pode criar.
- O usuário clica em ordenar conteúdo: o sistema exibe as opções de ordenação dos conteúdos listados.
- O usuário clica no título do conteúdo: o sistema exibe as informações do conteúdo.
- O usuário clica nas coleções criadas: o sistema lista os conteúdos existentes dentro da coleção.

Tabela 16 - Requisito Gerenciar Galeria.

US5 – Gerenciar Galeria
R.5.1 Cadastrar galeria; R.5.2 Editar galeria.

Esse requisito apresentado na Tabela 16 tem objetivo de realizar o cadastro de uma galeria de imagens e gerenciá-la de acordo com as necessidades do usuário. O cadastro de uma galeria será feito por meio de: inserção de um *link* que contenha uma imagem, *upload* ou pela seleção de imagens já cadastradas anteriormente no sistema. Será apresentada uma imagem intuitiva na capa dessa galeria, indicando o tipo de imagem que existe dentro dessa galeria, assim como seu título, descrição e *tags*.

Tabela 17 - Requisito Gerenciar Coleções.

US6 – Gerenciar Coleções
R.6.1 Criar coleções; R.6.2 Editar coleções. R.6.3 Cadastrar conteúdos dentro das coleções; R.6.4 Inserir conteúdos existentes dentro das coleções; R.6.5 Inserir coleções existentes dentro de outra; R.6.6 Excluir Coleções.

O requisito gerenciar coleções (Tabela 17) tem o objetivo de agrupar um conjunto de conteúdos dentro de uma pasta, ou seja, um usuário deseja criar uma coleção “Trabalhos” e para melhor organizar seus conteúdos pretende colocar todos os conteúdos que fazem parte desse trabalho dentro da coleção criada. Quando o criador excluir uma coleção que contenha uma coleção ou conteúdo, por exemplo, eles automaticamente são jogados para um diretório acima.

Tabela 18 - Requisito Gerenciar Permissões.

US7 – Gerenciar Permissões
R.7.1 Definir permissão do conteúdo; R.7.2 Editar permissão do conteúdo; R.7.3 Definir permissão coleção; R.7.4 Editar permissão coleção.

Na tabela 18, é apresentado o requisito que tem como objetivo definir os tipos de permissões para um conteúdo ou uma coleção. Dentre as permissões que existem sobre um determinado conteúdo ou coleção, a “visibilidade” é uma delas, pois ao cadastrar um conteúdo o usuário poderá disponibilizá-lo de forma que fique:

- Público na Web (visível para qualquer pessoa na internet);
- Público para qualquer pessoa com o *link* (visível para pessoa que tiver o *link*);
- Particular (somente pessoas que tiverem a permissão poderão acessar).

Esses conteúdos ou coleções também poderão ter as seguintes permissões de compartilhamento:

- Os conteúdos podem ter permissão para gravação e leitura;
- As coleções podem ter permissão para gravação e leitura;
- Os conteúdos que estão em coleções assumem, por padrão, as permissões da coleção, mas as permissões podem ser alteradas (para um conteúdo específico).

Quando um conteúdo ou coleção é criado, é atribuído um criador a eles, que é o usuário responsável por esse conteúdo ou coleção. O a função de “criador” de um conteúdo só poderá ser alterada pelo próprio criador. Ao compartilhar uma coleção com outros usuários, e a permissão da coleção for de gravação e leitura, esses usuários com os quais a coleção foi compartilhada, poderão criar outras coleções dentro desta e, por padrão, terão as mesmas permissões da coleção pai, logo os usuários terão permissão para criar uma outra coleção dentro da que foi compartilhada com eles, podendo alterar a permissão dessas coleções de acordo com suas necessidades.

Tabela 19 - Requisito Gerenciar Recompilhamento.

US8 – Gerenciar Recompilhamento
R.8.1 Realizar compartilhamento com cópia;
R.8.2 Realizar compartilhamento sem cópia;
R.8.3 Não compartilhar.

O requisito apresentado na Tabela 19 permite que conteúdos que foram compartilhados possam ser recompartilhados. Para esse requisito o usuário que compartilhou um conteúdo poderá realizar as seguintes ações:

- Um conteúdo ou coleção que foi compartilhado não poderá ser recompartilhado;
- Pode ser recompartilhado com cópia (neste caso, o usuário faz uma cópia do conteúdo);

a cópia é de sua propriedade, então ele pode alterá-la). O usuário poderá alterar as *tags* (colocar novas), mas não poderá remover as existentes;

- Poderá ser recompartilhado sem cópia (neste caso, o usuário não faz uma cópia do conteúdo, mas repassa o *link* do original), onde o usuário só poderá adicionar ou cadastrar as suas *tags* (mantendo as *tags* do proprietário).

Banco de Dados Relacional

O modelo relacional desenvolvido para o sistema em questão é composto por 34 entidades de relacionamento, tendo como principal tabela a entidade “Conteúdo”. Esse modelo é extensível, pois permite criar vários tipos de conteúdos (Imagem, Vídeo, Áudio, Texto, Arquivo e Galerias) com a finalidade de organizar as informações que serão inseridas pelos usuários. A seguir é apresentada a estrutura (Figura 29) que representa esse modelo:

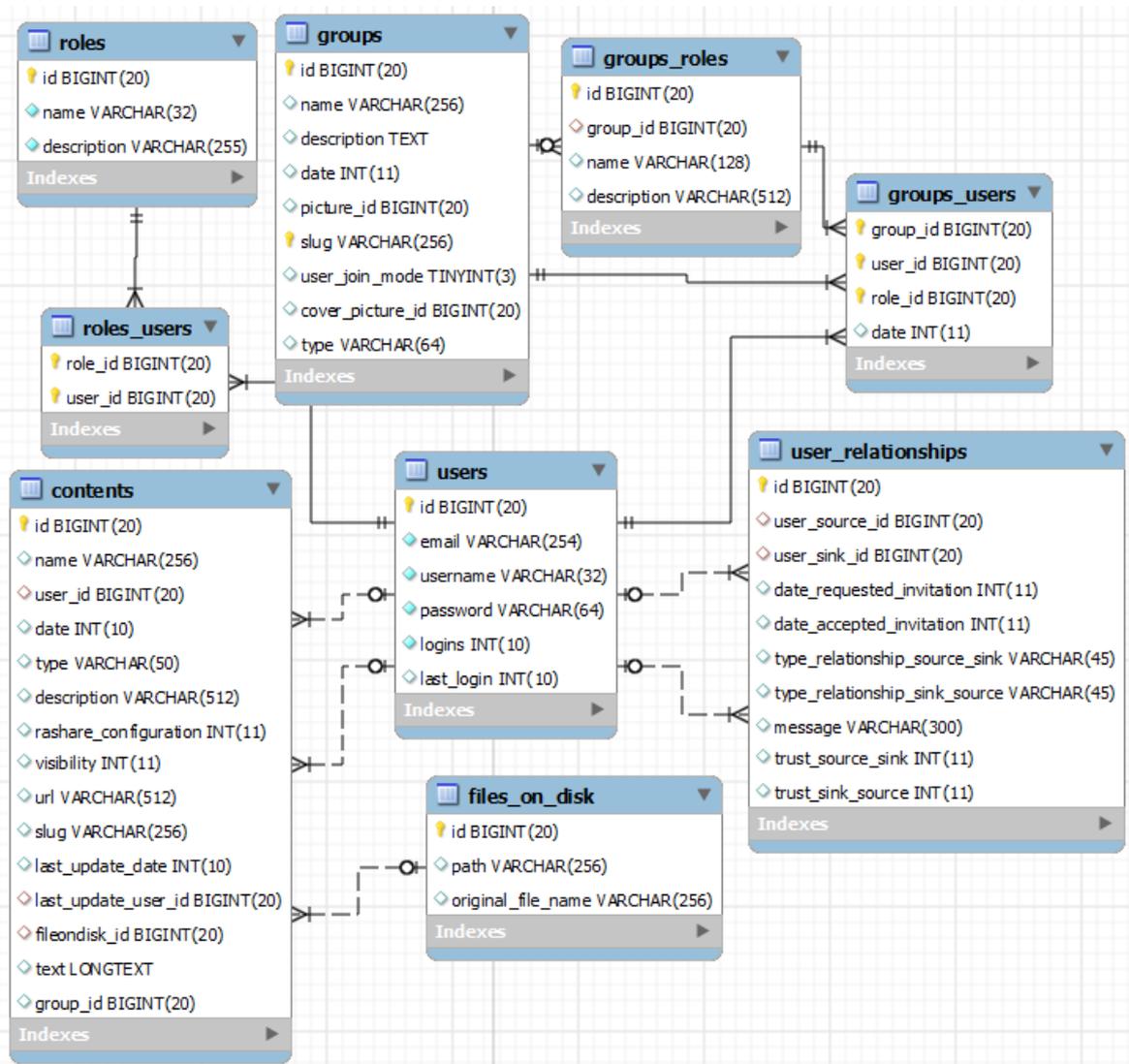


Figura 61 - Parte do Modelo Relacional (Banco de Dados).

Dicionário de Dados

O Dicionário de dados representa a documentação do banco de dados do sistema de Gerenciamento de Conteúdo, onde serão apresentadas tabelas contendo cada entidade e sua descrição, bem como os atributos dessas entidades, o seu tipo e a descrição de cada um.

- Tabela: Content
- Descrição: Armazena dados do conteúdo cadastrado pelo usuário do sistema.

Tabela 20 - Dicionário de Dados da Entidade Content

Campo	Tipo	Descrição
id	uniqueidentifier	Chave Primária
name	varchar(225)	Título do conteúdo

user_id	int	Chave(Referencia Usuario.Id)
date	datetime	Data de Cadastro do Conteúdo
type	int	Indica o tipo de conteúdo cadastrado
description	varchar(max)	Descreve o conteúdo cadastrado
rashare_configuration	int	Indica o tipo de compartilhamento que será permitido
visibility	int	Guarda o tipo de visibilidade que o conteúdo terá.
url	varchar(500)	Link que indica algum conteúdo existente na web.
slug	Varchar(225)	Armazena o título do conteúdo sem caracteres e espaço.
last_update_date	datetime	Armazena a data da última atualização realizada no conteúdo.
last_update_user_id	int	Chave(Referencia Usuario.Id)
Fileondisk_id	int	Chave(Referencia ArquivoEmDisco.Id)
text	longtext	Armazena o conteúdo do tipo texto
group_id	int	Chave(Referencia Group.Id)

- Tabela: file_on_disk
- Descrição: Entidade responsável por armazenar conteúdo em disco.

Tabela 21 - Dicionário de Dados da Entidade file_on_disk

Campo	Tipo	Descrição
id	uniqueidentifier	Chave Primária
path	varchar(225)	Armazena o caminho mais o nome gerado do arquivo
original_file_name	varchar(225)	Descreve o nome original do arquivo

- Tabela: User
- Descrição: Armazena dados do usuário cadastrado no sistema.

Tabela 22 - Dicionário de Dados da Entidade User

Campo	Tipo	Descrição
Id	bigint	PK
Name	varchar(64)	
Password	varchar(128)	
PasswordFormat	int	Representa o formato da senha: 0 - texto plano (clear); 1 - cifrado (encrypted); 2 - hashed. Não está sendo utilizado no momento
PasswordSalt	varchar(128)	
CreateDate	datetime	
LastLoginDate	datetime	
LastPasswordChangeDate	datetime	
IsOnline	bit	
IsLockedOut	bit	
LastLockoutDate	datetime	
FailedPasswordAttemptCount	int	
FailedPasswordAttemptWindowStart	datetime	

IsApproved	bit	
PasswordQuestion	varchar(256)	
PasswordAnswer	varchar(256)	
FailedPasswordAnswerAttemptCount	int	
FailedPasswordAnswerAttemptWindowStart	datetime	
Comment	varchar(1024)	

- Tabela: user_relationships
- Descrição: Armazena os relacionamentos entre os contatos da rede.

Tabela 23 - Dicionário de Dados da Entidade User_relationships

Campo	Tipo	Descrição
id	BIGINT(20)	primary key
user_source_id	BIGINT(20)	FK(Referencia User.OID)
user_sink_id	BIGINT(20)	FK(Referencia User.OID)
date_request_invitation	INT	Data de realização do convite para relacionamento
date_accepted_invitation	INT	Data em que o convite de relacionamento foi aceito
message	VARCHAR(300)	A mensagem representa um texto que o usuário solicitante (user_source) envia ao usuário solicitado para auxiliar na sua identificação.
type_relationship_source_sink	VARCHAR(40)	representa o tipo de

		relacionamento do usuário source (origem) para sink (destino)
type_relationship_sink_source	VARCHAR(40)	representa o tipo de relacionamento do usuário sink para o usuário source.