

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
DEPARTAMENTO DE INFORMÁTICA E ESTATÍSTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA  
COMPUTAÇÃO**

João de Amorim Junior

**MODELO DE SELEÇÃO DINÂMICA DE OBJETOS DE  
APRENDIZAGEM BASEADO EM AGENTES**

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina para a obtenção do Grau de Mestre em Ciência da Computação  
Orientador: Prof. Dr. Ricardo Azambuja Silveira

Florianópolis  
2015

Ficha de identificação da obra elaborada pelo autor,  
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Amorim Junior, João de  
MODELO DE SELEÇÃO DINÂMICA DE OBJETOS DE APRENDIZAGEM  
BASEADO EM AGENTES / João de Amorim Junior ; orientador,  
Ricardo Azambuja Silveira - Florianópolis, SC, 2015.  
189 p.

Dissertação (mestrado) - Universidade Federal de Santa  
Catarina, Centro Tecnológico. Programa de Pós-Graduação em  
Ciência da Computação.

Inclui referências

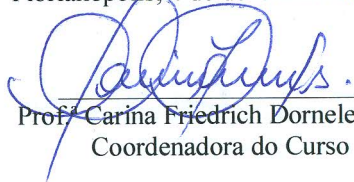
1. Ciência da Computação. 2. Inteligência computacional  
aplicada à educação. 3. Seleção dinâmica de objetos de  
aprendizagem. 4. Experiência de aprendizagem adaptativa e  
reutilizável. 5. Objetos inteligentes de aprendizagem. I.  
Silveira, Ricardo Azambuja. II. Universidade Federal de  
Santa Catarina. Programa de Pós-Graduação em Ciência da  
Computação. III. Título.

João de Amorim Junior

**MODELO DE SELEÇÃO DINÂMICA DE OBJETOS DE  
APRENDIZAGEM BASEADO EM AGENTES**


Esta dissertação foi julgada adequada para obtenção do título de mestre e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Florianópolis, 4 de dezembro de 2015.

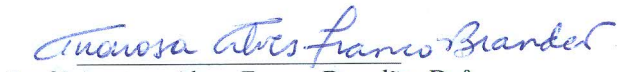


Prof.<sup>a</sup> Carina Friedrich Dorneles, Dr.<sup>a</sup>  
Coordenadora do Curso


**Banca Examinadora:**




Prof. Ricardo Azambuja Silveira, Dr.  
Orientador  
Universidade Federal de Santa Catarina



Prof.<sup>a</sup> Anarosa Alves Franco Brandão, Dr.<sup>a</sup>  
Universidade de São Paulo



Prof.<sup>a</sup> Sílvia Modesto Nassar, Dr.<sup>a</sup>  
Universidade Federal de Santa Catarina



Prof. Elder Rizzon Santos, Dr.  
Universidade Federal de Santa Catarina



Este trabalho é dedicado aos meus  
queridos pais e irmã.



## AGRADECIMENTOS

Agradeço inicialmente a Deus pela oportunidade de vivenciar mais essa importante etapa da vida, e a meus pais, João e Iloir, pela dedicação e carinho que sempre demonstraram para com os filhos. Meus agradecimentos também a minha irmã Katiucia que serviu de inspiração e incentivo para cursar o mestrado. Da mesma forma, minha gratidão a todos os meus parentes que torceram para que eu alcançasse este objetivo, em especial a minha avó Rosa, e aos saudosos avós Areneu, João e Luzia.

Agradeço também ao meu orientador, Prof. Ricardo Azambuja Silveira, pela receptividade e aceite em me orientar mesmo não me conhecendo, e por ter dado todo o apoio necessário durante o desenvolvimento desta pesquisa. Em momentos onde a direção a seguir não estava clara para mim, o Prof. Ricardo Silveira apontou o caminho e contribuiu muito para o resultado alcançado neste trabalho.

Agradeço ainda aos ilustres professores membros da banca, Prof.<sup>a</sup> Anarosa Brandão, Prof.<sup>a</sup> Silvia Nassar e Prof. Elder Santos, por aceitarem participar da avaliação do meu trabalho, bem como pelas contribuições feitas para o aperfeiçoamento desta pesquisa.

Estendo os agradecimentos a todos os professores e alunos do Grupo de pesquisa em Inteligência Artificial e Tecnologia Educacional (IATE - UFSC), que colaboraram de alguma forma para o desenvolvimento deste trabalho, em especial ao colega Thiago Ângelo Gelaim pelas conversas elucidativas sobre a abordagem Multiagente.

Meus sinceros agradecimentos também à chefe da Seção de Suporte ao Ensino a Distância do TJSC, Neusa do E. S. Costa Buss, pelo apoio e compreensão em relação à flexibilização de meus horários de trabalho em dias de aula, bem como nas marcações de férias nos fins de semestre.

Aproveito para agradecer igualmente a todas as pessoas que me auxiliaram e ensinaram durante minha vida profissional e acadêmica, sejam colegas e ex-colegas de estudo, e de trabalho, ou os caríssimos professores que me ajudaram a chegar até aqui, desde a minha primeira professora no ensino fundamental até o meu orientador do PPGCC.

A todos os meus sinceros agradecimentos e votos de que tenham sempre muito sucesso em suas vidas.





## RESUMO

Este trabalho descreve um modelo que possibilita o oferecimento de experiências de aprendizagem na educação on-line, baseado no paradigma de Sistemas Multiagente (SMA), com o intuito de facilitar o uso de Objetos de Aprendizagem (OA) de forma adaptativa em sistemas de gerenciamento de aprendizagem (LMS), bem como de favorecer o reuso de materiais instrucionais que sigam padrões de referência.

O modelo proposto estende o conceito de Objetos Inteligentes de Aprendizagem (OIA), através da utilização de uma arquitetura de agentes BDI (*Belief, Desire, Intention*), sendo capaz de se comunicar com os elementos que constituem o OA, de acordo com um padrão de empacotamento e sequenciamento de objetos.

O modelo teórico proposto possibilita a seleção dinâmica de recursos instrucionais com base em informações do LMS, dos metadados do OA, e do desempenho obtido pelo estudante durante a interação com o objeto. Esses elementos do modelo de dados são usados no processo de raciocínio dos agentes, possibilitando o desenvolvimento de experiências de aprendizagem aprimoradas, e com adaptatividade dinâmica.

Além disso, a integração com um LMS favorece o reuso de recursos educacionais que são desenvolvidos segundo padrões de referência, e que podem ficar acessíveis ao processo de adaptação automaticamente, logo que forem incluídos no repositório de objetos, sem a necessidade de reconfigurar a estrutura do curso.

Para instanciar o modelo teórico, foi desenvolvido um protótipo compatível com objetos de aprendizagem segundo o padrão de integração SCORM e cujos metadados sigam o padrão IEEE-LOM. Este protótipo integra o ambiente de agentes (denominado ILOMAS) ao LMS Moodle, permitindo o acesso aos OIA de forma integrada a cursos reais de instituições que se utilizem da plataforma Moodle.

O protótipo foi desenvolvido com base em um subconjunto do modelo teórico, com o intuito de permitir a validação do sistema proposto, do ponto de vista computacional, através de simulações. Conforme verificado na avaliação deste protótipo, obteve-se adaptatividade e reuso. Por fim, apontou-se possibilidades de evolução do modelo, principalmente com a integração à busca semântica de objetos de aprendizagem com base em ontologias.

**Palavras-chave:** Seleção dinâmica de objetos de aprendizagem, Experiência de aprendizagem adaptativa e reutilizável, Objetos inteligentes de aprendizagem.



## ABSTRACT

This work describes a model that enables the offering of learning experiences in online education, based on the paradigm of Multi-Agent Systems (MAS), in order to facilitate the use of Learning Objects (LO) adaptively in learning management systems (LMS) as well as to encourage the reuse of instructional materials that follow standards.

The proposed model extends the concept of intelligent learning objects (ILO), by using a BDI agents' architecture (Belief, Desire, Intention), being able to communicate with the LO's elements, according to a packaging and sequencing standard.

The proposed theoretical model enables dynamic selection of instructional resources based on information from the LMS, the learning object metadata, and the performance achieved by the student during the interaction with the LO. These elements of the data model are used in the agent's reasoning process, enabling the development of improved learning experiences, and dynamic adaptivity.

Furthermore, the integration with an LMS promotes reuse of educational resources that are developed following reference standards, enabling their access in the adaptation process automatically, as soon as they are included in the LO repository, without having to reconfigure the course structure.

To instantiate the theoretical model, it was developed a prototype compliant with SCORM standard LO and whose metadata follow the IEEE-LOM standard. This prototype integrates the agent environment (called ILOMAS) to LMS Moodle, allowing access to the ILO seamlessly to real courses of institutions that use Moodle platform.

The prototype was developed based on a subset of the theoretical model, in order to enable the computational validation of the proposed system, through simulations. As verified on this prototype's evaluation, adaptivity and reuse were obtained. Finally, evolution opportunities to the model was pointed, especially with the integration of semantic search for learning objects based on ontologies.

**Keywords:** Dynamic selection of learning objects, Adaptive and reusable learning experience, Intelligent learning objects.



## LISTA DE FIGURAS

Figura 1 – Relacionamento entre OA, Repositórios e LMS.....	33
Figura 2 – Modelo conceitual da especificação <i>IMS Content Packaging</i> .....	39
Figura 3 – Especificações da arquitetura SCORM.....	43
Figura 4 – Componentes do conteúdo SCORM.....	44
Figura 5 – O link de comunicação entre um SCO e um LMS.....	45
Figura 6 – O diagrama de estados do SCORM RTE API .....	48
Figura 7 – Uso do modelo de dados com a API SCORM.....	50
Figura 8 – Exemplo da sintaxe de regras de sequenciamento .....	51
Figura 9 – Visão geral da estrutura do <i>Common Cartridge</i> .....	52
Figura 10 – Ecossistema de aprendizagem via TinCan.....	54
Figura 11 – Estrutura e processo de um agente BDI.....	66
Figura 12 – Pilha da linguagem da Web Semântica.....	72
Figura 13 – Elementos da Arquitetura do Modelo Teórico.....	89
Figura 14 – Fases da Metodologia Prometheus .....	92
Figura 15 – Diagrama da Visão Geral da Análise.....	96
Figura 16 – Diagrama de Objetivos do Sistema.....	97
Figura 17 – Diagrama de Visão Geral dos Papéis.....	98
Figura 18 – Diagrama de Acoplamento de Dados .....	99
Figura 19 – Diagrama de Relacionamento entre Papéis e Agentes.....	100
Figura 20 – Diagrama de Visão Geral do Sistema.....	100
Figura 21 – Diagrama de Visão Geral do Agente - LMSAgent.....	101
Figura 22 – Diagrama de Visão Geral do Agente - ILOAgent.....	101
Figura 23 – Diagrama de Capability – <i>IdentifyLearningExperienceCapability</i> .....	102
Figura 24 – Diagrama de Capability – <i>SearchLearningObjectCapability</i> .....	102
Figura 25 – Classes da ontologia do modelo conceitual do OIA .....	108
Figura 26 – Propriedades da ontologia do modelo conceitual do OIA .....	109
Figura 27 – Arquitetura do sistema.....	119
Figura 28 – Arquitetura de Componentes Ativos.....	120
Figura 29 – Elementos do Padrão de Metadados IEEE-LOM .....	129
Figura 30 – Modelo de Dados SCORM.....	131
Figura 31 – ILOMAS - Fluxo de Execução.....	135
Figura 32 – Início da sessão de aprendizagem e busca pelo OA.....	137
Figura 33 – Interface ILOAgent e SCORM RTE .....	138
Figura 34 – Fluxo do processo de raciocínio do ILOAgent .....	139
Figura 35 – Tela de Configuração do <i>Plug-in</i> ILOIR .....	142
Figura 36 – Estrutura do curso com SCORM padrão .....	145
Figura 37 – Estrutura do curso com o recurso ILOIR.....	146
Figura 38 – Diagnóstico e sinalização de novo OA ao Aluno 3 (tempo t1).....	152
Figura 39 – Oferta do novo OA (pré-requisito do OA prévio) ao Aluno 3 (tempo t2).....	153



## LISTA DE QUADROS

Quadro 1 – Categorização das funções definidas na API SCORM.....	47
Quadro 2 – Elementos do Modelo de Dados SCORM.....	48
Quadro 3 – Comparação entre os trabalhos relacionados e esta proposta.....	79
Quadro 4 – Planos para tratamento de falhas técnicas .....	104
Quadro 5 – Planos para tratamento de falhas educacionais .....	105
Quadro 6 – Resultados da configuração dos cursos no LMS.....	147
Quadro 7 – Resultados da interação simulada dos estudantes com o recurso ILOIR quanto ao comportamento dinâmico .....	149
Quadro 8 – Resultados da interação dos estudantes com o recurso ILOIR quanto à oferta de objetos de aprendizagem extras, conforme o desempenho.....	150





## LISTA DE ABREVIATURAS E SIGLAS

AICC – Aviation Industry CBT (Computer-Based Training) Committee  
ACM – Association for Computing Machinery  
ADL – Advanced Distributed Learning  
AOSE – Agent Oriented Software Engineering  
API – Application Programming Interface  
AVA – Ambiente Virtual de Aprendizagem  
BB – Belief Base  
BDI – Beliefs, Desires and Intentions (Crenças, Desejos e Intenções)  
CAPES – Coordenação de Aperfeiçoamento de Pessoal de Nível Superior  
CNJ – Conselho Nacional de Justiça  
DI – Designer Instrucional  
DL – Description Language  
EaD – Ensino a Distância (ou Educação a Distância)  
EPIA – Encontro Português de Inteligência Artificial  
FIPA – Foundation For Intelligent, Physical Agents  
HTML – HiperText Markup Language  
HTTP – HiperText Transfer Protocol  
IA – Inteligência Artificial  
IATE – Grupo de pesquisa em Inteligência Artificial e Tecnologia Educacional  
IDE – Integrated Development Environment  
IEC – International Electrotechnical Commission  
IEEE – Institute of Electrical and Electronics Engineers  
IJELLO – Interdisciplinary Journal of E-learning and Learning Objects  
ILO – Intelligent Learning Object  
ILOIR – Intelligent Learning Object’s Interface Resource  
ILOMAS – Intelligent Learning Object Multi-Agent System  
IMS – Instructional Management Systems  
IMS CC – IMS Common Cartridge  
IMS CP – IMS Content Packaging  
IMS GLC – IMS Global Learning Consortium  
IMS LTI – IMS Learning Tools Interoperability  
INE – Departamento de Informática e Estatística  
ISO – International Organization for Standardization  
IWT – Intelligent Web Teacher  
JSON – JavaScript Object Notation  
KB – Knowledge Base  
LMS – Learning Management System  
LO – Learning Object

LOM – Learning Object Metadata  
LTSC – Learning Technology Standards Committee  
LRS – Learning Record Store  
MASLE – Multiagent System Based Learning Environments  
MEC – Ministério da Educação  
MOOC – Massive Open On-line Course  
MOODLE – Modular Object-Oriented Dynamic Learning Environment  
OA – Objeto de Aprendizagem  
OBAA – Objetos de Aprendizagem Baseados em Agentes  
OIA – Objeto Inteligente de Aprendizagem  
OO – Orientação a Objetos  
OWL – Web Ontology Language  
PAAMS – Practical Applications of Agents and Multi-Agent Systems  
PDT – Prometheus Design Tool  
PHP – Hypertext Processor  
PRS – Procedural Reasoning System  
RDF – Resource Description Framework  
SBIE – Simpósio Brasileiro de Informática na Educação  
SCA – Service Component Architecture  
SCO – Sharable Content Objects  
SCORM – Sharable Content Object Reference Model  
SCORM CAM – Content Aggregation Model  
SCORM RTE – SCORM Run-Time Environment  
SCORM SN – SCORM Sequencing and Navigation  
SMA – Sistema Multiagente  
SOA – Service-Oriented Architecture  
SPARQL – SPARQL Protocol and RDF Query Language  
STI – Sistemas Tutores Inteligentes  
TIC – Tecnologias de Informação e Comunicação  
TJSC – Tribunal de Justiça de Santa Catarina  
TLA – Training and Learning Architecture  
UFRGS – Universidade Federal do Rio Grande do Sul  
UFRJ – Universidade Federal do Rio de Janeiro  
UFSC – Universidade Federal de Santa Catarina  
xAPI – Experience API  
XML – eXtensible Markup Language  
W3C – World Wide Web Consortium

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>21</b>
1.2	MOTIVAÇÕES E JUSTIFICATIVAS .....	23
1.2	PROBLEMA DE PESQUISA .....	24
1.3	OBJETIVOS .....	26
<b>1.3.1</b>	<b>Objetivo Geral.....</b>	<b>27</b>
<b>1.3.2</b>	<b>Objetivos Específicos .....</b>	<b>27</b>
1.4	METODOLOGIA.....	27
1.5	CONTRIBUIÇÕES ESPERADAS.....	28
1.6	ESTRUTURA DO TEXTO .....	29
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>31</b>
2.1	OBJETOS DE APRENDIZAGEM.....	31
2.2	PADRÕES DE OBJETOS DE APRENDIZAGEM .....	33
<b>2.2.1</b>	<b>Padrões de Metadados.....</b>	<b>34</b>
2.2.1.1	IEEE-LOM.....	34
2.2.1.2	Dublin Core.....	36
2.2.1.1	OBAA .....	37
<b>2.2.2</b>	<b>Padrões de Empacotamento.....</b>	<b>37</b>
<b>2.2.3</b>	<b>Padrões de Interface e Comunicação .....</b>	<b>39</b>
2.2.3.1	AICC .....	39
2.2.3.2	IMS LTI .....	40
<b>2.2.4</b>	<b>Padrões de Integração .....</b>	<b>41</b>
2.2.4.1	SCORM .....	41
2.2.4.1.1	CAM ( <i>Content Aggregation Model</i> ) .....	43
2.2.4.1.2	RTE ( <i>Run-Time Environment</i> ) .....	45
2.2.4.1.3	SN ( <i>Sequencing and Navigation</i> ).....	50
2.2.4.2	IMS <i>Common Cartridge</i> .....	51
2.2.4.3	ADL Training and Learning Architecture .....	53
2.3	REPOSITÓRIOS DE OBJETOS DE APRENDIZAGEM.....	55
2.4	SISTEMAS DE GERENCIAMENTO DE APRENDIZAGEM .....	56
2.5	SISTEMAS DE APOIO AO ENSINO BASEADOS EM IA .....	58
<b>2.5.1</b>	<b>Sistemas Tutores Inteligentes.....</b>	<b>59</b>
<b>2.5.2</b>	<b>Sistemas Multiagente.....</b>	<b>60</b>
2.5.2.1	Modelo BDI .....	64
2.5.2.2	Arquitetura PRS .....	65
2.5.2.3	Desenvolvimento de Sistemas Multiagente .....	67
<b>2.5.3</b>	<b>Representação do Conhecimento.....</b>	<b>68</b>
2.5.3.1	Ontologias .....	69
2.5.3.2	Web Semântica .....	71
2.6	SISTEMA TUTOR INTELIGENTE BASEADO EM AGENTES COMPATÍVEL COM O PADRÃO SCORM.....	73
<b>3</b>	<b>TRABALHOS RELACIONADOS .....</b>	<b>75</b>
3.1	SISTEMAS DE APOIO AO ENSINO ADAPTATIVOS.....	76

3.2	COMPARAÇÃO ENTRE OS TRABALHOS IDENTIFICADOS ..	78
3.2.1	Análise das Questões Elencadas .....	81
3.2.1	Características da Proposta .....	84
<b>4</b>	<b>MODELO PROPOSTO: ILOMAS .....</b>	<b>85</b>
4.1	MODELO TEÓRICO .....	87
4.2	MODELAGEM DO SMA .....	91
4.3	REPRESENTAÇÃO DAS CRENÇAS .....	106
4.4	ESTRATÉGIAS ADOTADAS NO MODELO TEÓRICO .....	110
<b>5</b>	<b>RESULTADOS .....</b>	<b>115</b>
5.1	TECNOLOGIAS UTILIZADAS NA IMPLEMENTAÇÃO .....	115
5.2	ARQUITETURA DO SISTEMA .....	118
5.3	PROCESSO DE DELIBERAÇÃO DO OIA .....	127
5.4	ILOIR: MOODLE PLUG-IN .....	141
5.5	AVALIAÇÃO DO PROTÓTIPO .....	144
5.6	LIMITAÇÕES .....	153
5.7	DIFICULDADES ENCONTRADAS .....	155
5.8	CONTRIBUIÇÕES OBTIDAS .....	156
5.9	DISCUSSÕES SOBRE A INTEGRAÇÃO ENTRE O MODELO PROPOSTO E ONTOLOGIAS .....	157
<b>6</b>	<b>CONSIDERAÇÕES FINAIS.....</b>	<b>161</b>
6.1	CONCLUSÃO.....	161
6.2	TRABALHOS FUTUROS .....	164
	<b>REFERÊNCIAS .....</b>	<b>169</b>
	<b>APÊNDICE A - Publicações.....</b>	<b>189</b>

## 1 INTRODUÇÃO

O processo de ensino-aprendizagem realizado de forma on-line vem contribuindo para a universalização do acesso à educação no Brasil e no mundo (DETONI *et al.*, 2014), sobretudo com a recente evolução da Internet nos últimos anos (PEREIRA *et al.*, 2014). Além disso, destaca-se que a ampla difusão dos dispositivos móveis na população brasileira e mundial contribuiu para disseminação da educação on-line, propiciando novas possibilidades educacionais a partir das tecnologias digitais em rede (SCHONS *et al.*, 2008; WEBER e SANTOS, 2013).

De acordo com Pereira *et al.* (2005), a educação on-line (ou *e-learning*) tem papel importante no sistema educacional brasileiro, pelo fato de permitir o acesso a cursos de nível superior aos cidadãos que tem menos possibilidade de frequentar um curso presencial convencional, e também por favorecer a capacitação contínua de profissionais dos diversos ramos do mercado de trabalho.

Segundo Bottentuit Junior e Coutinho (2012), embora muitas vezes utilizado como sinônimo de Educação a Distância (EaD), o conceito de educação on-line está relacionado a uma realidade mais abrangente, propiciando a disseminação do conhecimento com base nos recursos disponíveis na Internet, seja na modalidade presencial, a distância, ou ainda, mista (*blended learning*).

A educação on-line permite que o professor se aproxime dos alunos, ao ultrapassar as fronteiras físicas da sala de aula, e enriqueça o processo de ensino-aprendizagem com novas práticas pedagógicas e outras fontes de informação, tais como: som, vídeo, animação, etc. (BOTTENTUIT JUNIOR e COUTINHO, 2012).

Filatro (2010, p. 52) afirma que:

o conceito principal de educação on-line é de uma educação distribuída, que valoriza o processo, e não uma educação feita a distância, que valoriza os extremos de produção e consumo.

Diversas instituições de ensino utilizam o *e-learning* como uma das suas vertentes de atuação (PEREIRA *et al.*, 2005; FILATRO, 2010, BUSS e HÜLSE, 2015), e formas relativamente novas de educação e treinamento on-line são cada vez mais presentes, como por exemplo, os

cursos massivos on-line (MOOCs) (FASSBINDER *et al.*, 2014) e os ambientes 3D (ALLISON *et al.*, 2012).

Como consequência, percebe-se um aumento da demanda por cursos e materiais de educação on-line de qualidade. O uso de diferentes tecnologias de informação e comunicação (TIC) nesta modalidade educacional resulta em melhores experiências de aprendizagem. A área da Informática aplicada à Educação (ou Informática na Educação) se preocupa com questões desta natureza.

Quando se fala em produção de conteúdo educacional, um termo comumente utilizado é o *design* instrucional (ou desenho instrucional). Segundo Filatro (2010, p. 64-65), *design* instrucional deve ser compreendido como:

a ação intencional 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.

O processo do desenho instrucional é multidisciplinar, envolvendo diversos atores, tais como: gestores, professores, conteudistas, especialistas em tecnologia, programadores, etc. Outro elemento importante desta engrenagem é o *Designer* Instrucional (DI), um profissional que atua diretamente em todas as etapas que envolvem a oferta de um curso na educação on-line, interagindo com os todos os outros atores (FILATRO, 2010).

Uma destas etapas é a produção do conteúdo educacional propriamente dito, o que deve ser realizado com base nas necessidades de aprendizagem, nos objetivos instrucionais, nas características dos alunos e em restrições existentes. Os materiais e recursos produzidos devem ser instrumentos que favoreçam a aprendizagem (FILATRO, 2010).

Os softwares educacionais (ou aplicativos educacionais, sistemas de apoio ao ensino, etc.) são importantes ferramentas utilizadas com intuito de aprimorar as práticas de ensino-aprendizagem. Cano (in SANCHO, 1998, p. 169) define software educacional como:

um conjunto de recursos informáticos projetados com a intenção de serem usados em contextos de ensino e aprendizagem. Tais programas abrangem finalidades muito diversas que podem ir da

aquisição de conceitos até o desenvolvimento de habilidades ou resolução de problemas.

## 1.1 MOTIVAÇÕES E JUSTIFICATIVAS

Existem diversas características que permitem verificar a qualidade de um software, tais como: fatores ergonômicos, tempo de resposta, tratamento de erros, confiabilidade, eficiência, interoperabilidade, facilidade de uso, portabilidade, adaptatividade, reusabilidade, entre outros (RAMOS e MENDONÇA, 1991; ROCHA e CAMPOS, 1993; BATISTA *et al.*, 2004; ISO/IEC 25010, 2011).

Além dos fatores supracitados, os softwares educacionais precisam levar em consideração questões específicas referentes à aprendizagem, pois a solução computacional deve ser adequada a uma perspectiva educacional (PEREIRA *et al.*, 2005; GUENAGA *et al.*, 2012).

Dentre as características de qualidade citadas, pode-se destacar a reusabilidade e a adaptatividade, por serem importantes aspectos que facilitam a produção de conteúdos educacionais por parte dos professores e designers instrucionais (DI), bem como auxiliam o processo de ensino-aprendizagem, favorecendo o entendimento por parte dos estudantes.

Segundo Tangarife (2007), um sistema que apresente adaptatividade é capaz de se modificar conforme sua percepção do usuário, sem que o indivíduo solicite alterações explicitamente. Do ponto de vista de sistemas de apoio ao ensino, a adaptatividade permite que o ambiente se molde às diferentes necessidades e aos estilos de aprendizagem dos alunos. Um software educacional adaptativo potencializa a compreensão dos conteúdos por parte do estudante, de forma personalizável, levando em consideração seu nível de conhecimento e suas preferências (BREMIGARTNER *et al.*, 2014; MAHKAMEH e BAHREININEJAD, 2011).

A reusabilidade, por sua vez, é a característica que possibilita o reaproveitamento (integral ou parcial) de funções desenvolvidas em um programa em outras aplicações (ROCHA e CAMPOS, 1993). Da perspectiva educacional, é um atributo que pode evitar a necessidade de se desenvolver novos recursos quando já existam outros materiais semelhantes previamente produzidos, e que estejam relacionados à mesma intenção instrucional (SILVA *et al.*, 2012).

Na literatura relacionada à Informática na Educação, e à Inteligência Computacional aplicada à Educação, são citadas algumas ferramentas computacionais que enriquecem o processo de ensino-aprendizagem, tais como: Objetos de Aprendizagem (OA), Ambientes

Virtuais de Aprendizagem (AVA), Sistemas de Gerenciamento de Aprendizagem (LMS, em inglês) e Sistemas Tutores Inteligentes (STI).

Com relação aos critérios de qualidade destacados anteriormente, pode-se examinar as referidas ferramentas com base no grau de adaptatividade e de reusabilidade que apresentam. Embora ambientes virtuais baseados em objetos de aprendizagem favoreçam o reuso de artefatos educacionais e os LMS contribuam para a sua disseminação em larga escala, estes apresentam um grau limitado de adaptação ao aluno, como ocorre nos Sistemas Tutores Inteligentes, havendo a necessidade de reconfiguração do sistema caso sejam necessárias mudanças de comportamento de acordo com o estudante.

O tipo de adaptação oferecido por estas ferramentas, de modo geral, precisa ser pré-programado pelo designer instrucional (DI) ou professor dentro do recurso ou ambiente, resultando em sistemas que em sua maioria não são dinamicamente adaptáveis (KOMLENOV *et al.*, 2010; SANTOS e JORGE 2013).

De acordo com Pozzebon *et al.* (2004, p. 1), a utilização de diferentes abordagens da Inteligência Computacional (ou Inteligência Artificial) na educação on-line “pode ser uma forma de diversificar ferramentas de auxílio ao ensino atendendo às necessidades pedagógicas e tecnológicas em questão”.

Além do que já foi exposto, esta pesquisa é motivada pela tentativa em se avançar nas questões que limitam e dificultam a ampla utilização de objetos de aprendizagem, principalmente devido à falta de ferramentas que facilitem o reuso e a adaptação dos objetos existentes (PARRISH, 2004; NASH, 2005; CHURCHILL, 2007; CHIKH, 2014). Propõe-se, portanto, uma abordagem que facilite a utilização de objetos de aprendizagem, de maneira dinâmica.

A motivação desta pesquisa decorre também de trabalhos anteriores desenvolvidos pelo grupo de pesquisa em Inteligência Artificial e Tecnologia Educacional (IATE) – INE – UFSC, no que se refere ao desenvolvimento de sistemas educacionais com comportamento inteligente. Portanto, a principal contribuição desta pesquisa se dá no campo da Informática na Educação e da Inteligência Computacional aplicada à Educação.

## 1.2 PROBLEMA DE PESQUISA

Conforme o cenário apresentado e com o intuito de produzir conteúdo *e-learning* o mais adaptável e reutilizável possível, depara-se com a seguinte questão: Como desenvolver sistemas de ensino on-line



adaptativos e reutilizáveis de maneira dinâmica, com base no desempenho do aluno?

Acredita-se que isto pode ser alcançado através da integração de duas das ferramentas computacionais de apoio ao ensino mencionadas, a saber, Objetos de Aprendizagem e Sistemas de Gerenciamento de Aprendizagem, em um modelo baseado na abordagem de agentes da Inteligência Computacional (os Sistemas Multiagente), para permitir a adaptação dinâmica da sessão de ensino-aprendizagem, de acordo com o desempenho do estudante durante esta experiência.

Desta forma, a utilização de elementos dos modelos de dados de padrões de objetos de aprendizagem (metadados, empacotamento, sequenciamento, etc.) possibilita que seja projetado um modelo de agentes cognitivos, que se utilizam destas informações para decidir quais objetos devem ser ofertados ao estudante, durante a interação deste aluno com o OA (seleção dinâmica dos objetos).

Considera-se que o monitoramento em tempo real (“*on the fly*”) da interação do aluno com o objeto de aprendizagem por um agente cognitivo enriquece a possibilidade adaptativa do sistema, devido à capacidade de se tomar decisões no decurso da sessão de aprendizagem.

Assim, o potencial de adaptatividade do sistema ao nível cognitivo do estudante é ampliado, e dessa forma, pode-se proporcionar experiências de aprendizagem mais adequadas às características do aluno. Com isso, é possível se obter melhores resultados educacionais, no que diz respeito ao desempenho da aprendizagem dos alunos.

Ademais, com a integração de um ambiente de agentes a um sistema de gerenciamento de aprendizagem (LMS), a reusabilidade de recursos instrucionais desenvolvidos previamente (segundo padrões de referência) é favorecida. Com isso, não seria necessário modificar a estrutura do curso no LMS para poder disponibilizar um novo objeto ao processo de adaptação do conteúdo para o estudante.

Deste modo, objetos de aprendizagem que sejam adicionados a repositórios de recursos instrucionais estarão automaticamente disponíveis para serem oferecidos pelos agentes a um estudante, em uma sessão de aprendizagem (caso este repositório seja acessível pelo agente).

O modelo a ser proposto deverá tomar como referência os modelos apresentados anteriormente por Gomes (2005) e por Silva (2007). Ambos propuseram a integração da abordagem de agentes e objetos de

aprendizagem resultando em experiências de ensino-aprendizagem reutilizáveis e aderentes a padrões de referência de recursos instrucionais.

Pretende-se avançar em algumas limitações das propostas mencionadas, tais como:

- A proposição de uma arquitetura de agentes cognitivos (BDI), diferindo dos agentes reativos que foram projetados à época.
- A expansão do modelo de dados do sistema, os quais serão utilizados pelos agentes cognitivos no processo de deliberação, levando-se em consideração os elementos de um padrão de integração de objetos de aprendizagem (empacotamento e sequenciamento), em especial o padrão SCORM (*Sharable Content Object Reference Model*).
- A integração do modelo de agentes com um Sistema de Gerenciamento de Aprendizagem (LMS) largamente difundido, para permitir a utilização da proposta em cursos on-line.

### 1.3 OBJETIVOS

Esta dissertação apresenta uma proposta para integrar os três tipos diferentes de paradigmas mencionados para a concepção de ambientes inteligentes de aprendizagem, utilizando os objetos de aprendizagem como as unidades instrucionais básicas. Intenciona-se, com este trabalho, o desenvolvimento de sistemas, que de forma dinâmica, sejam adaptativos às características do estudante, e que favoreçam o reuso de materiais

educacionais por parte dos professores ou *designers* instrucionais em um sistema de gerenciamento de aprendizagem (LMS).

### 1.3.1 Objetivo Geral

Propor um modelo de seleção dinâmica de Objetos de Aprendizagem (OA), de forma adaptativa, baseado na abordagem multiagente, e que favoreça o reuso de OA.

### 1.3.2 Objetivos Específicos

- Desenvolver um modelo teórico que estenda a abordagem de objetos inteligentes de aprendizagem.
- Propor um modelo conceitual de dados que serão utilizados pelos agentes cognitivos.
- Especificar o modelo da sociedade dos agentes através de metodologias de desenvolvimento orientado a agentes (AOSE).
- Projetar a integração do sistema multiagente elaborado com o padrão IEEE-LOM, o padrão SCORM e o LMS Moodle.
- Avaliar o modelo proposto com base no protótipo desenvolvido, através da simulação de cursos.

## 1.4 METODOLOGIA

Esta é uma pesquisa de base tecnológica, com a elaboração de um modelo computacional e o desenvolvimento de um protótipo para verificar a viabilidade deste modelo, do ponto de vista técnico. O método de trabalho é composto das seguintes etapas:

- Elaboração do modelo para o desenvolvimento dos objetos inteligentes de aprendizagem (OIA) capazes de serem integrados em um LMS, através de agentes com capacidade de raciocínio, com base em dados do ambiente virtual, do objeto de aprendizagem (OA) e da interação do aluno com o OA.
- Modelagem do SMA segundo a metodologia Prometheus (2015), e das crenças dos agentes (modelo conceitual) representando dados do LMS Moodle, do padrão de metadados de objetos de

aprendizagem IEEE-LOM, e do modelo de dados do padrão SCORM.

- Implementação do SMA, com base no framework Jadex (2015), para o desenvolvimento de agentes segundo a arquitetura BDI e compatível com tecnologias *web*.
- Criação de um protótipo para agregar novos Objetos Inteligentes de Aprendizagem ao LMS Moodle, sem a necessidade de prévia inclusão na estrutura do curso (teste de validação).

O Capítulo 4 (“Modelo Proposto: ILOMAS”) traz detalhes do modelo teórico proposto nesta dissertação.

## 1.5 CONTRIBUIÇÕES ESPERADAS

A principal contribuição desta pesquisa é na área da Informática na Educação, em especial no favorecimento ao reuso de recursos instrucionais com base em padrões e da adaptação de conteúdo conforme o desempenho apresentado pelo estudante. Desta forma, o modelo proposto é mais uma opção, dentre as abordagens existentes na literatura, com o intuito de ampliar a qualidade de recursos educacionais e de reduzir o esforço de desenvolvimento deste tipo de material através do reuso (DALMON *et al.*, 2012).

Almeja-se contribuir com um dos grandes desafios do ensino on-line, conforme citado por Abbad (2007), que é garantir uma estruturação suficiente dos eventos instrucionais, porém respeitando as diferenças individuais dos alunos. A autora afirma que “o ideal, em muitos casos, seria poder oferecer atividades personalizadas, de modo a otimizar os resultados de aprendizagem” (ABBAD, 2007, p. 360).

Sendo assim, como resultado deste trabalho, espera-se que a estratégia proposta seja utilizada no desenvolvimento de ferramentas inteligentes de ensino-aprendizagem, enriquecendo a experiência educacional on-line ao possibilitar a oferta de conteúdo mais aderente ao perfil do estudante.

Deseja-se que a ferramenta desenvolvida com base no modelo proposto seja utilizada em cursos de capacitação profissional, cursos massivos (MOOCs) ou outros cursos abertos, para acrescentar-lhes a capacidade de sequenciamento dinâmico de conteúdo segundo o desempenho do aluno.

Com relação às contribuições esperadas para a área da IA, mais especificamente para a abordagem de agentes, vislumbra-se a ampliação

no uso da tecnologia de agentes, através do desenvolvimento de sistemas multiagente com base na metodologia Prometheus e na ferramenta Jadex.

Esta contribuição à área de agentes é importante, especialmente com relação às abordagens para desenvolvimento, pois este campo ainda carece de uma disseminação mais ampla, principalmente fora do âmbito acadêmico.

Além disso, espera-se o favorecimento ao desenvolvimento de sistemas multiagente segundo o Jadex, pelo fato desta plataforma ser relativamente nova, principalmente esta versão mais recente baseada em tecnologia Java pura (POKAHR *et al.*, 2014), a qual abandona a ideia de lógica proposicional para implementar crenças e contexto de planos.

Esta abordagem apresenta vantagens de implementação para os desenvolvedores que possuem experiência em OO. Ressalta-se também a integração da plataforma Jadex com tecnologias *web* e dispositivos móveis, característica esta que favorece a utilização da abordagem de agentes em sistemas ubíquos e de aplicação prática.

Espera-se ainda que com base nesta pesquisa outros trabalhos venham a ser desenvolvidos, com o intuito de expandir o modelo, agregar novas características aos agentes e incorporar novas funcionalidades ao sistema como um todo, continuando o processo de evolução desta solução educacional que vem sendo desenvolvida pelo grupo de pesquisa há alguns anos.

## 1.6 ESTRUTURA DO TEXTO

Esta dissertação está organizada da seguinte maneira: o capítulo 2 descreve a fundamentação teórica; no capítulo 3, os trabalhos correlatos são mencionados; o capítulo 4 descreve detalhadamente o modelo proposto; o capítulo 5 apresenta os resultados obtidos; por fim, o capítulo 6 menciona as considerações finais da pesquisa.



## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta a fundamentação teórica desta pesquisa, ou seja, os conceitos teóricos que serviram de base para o desenvolvimento do modelo proposto. Dentre estes conceitos, destacam-se: os Objetos de Aprendizagem (OA), com os padrões relacionados; os Sistemas de Gerenciamento de Aprendizagem (em inglês, LMS); e os Sistemas Multiagente, com os modelos e ferramentas de desenvolvimento associados.

Os objetos de aprendizagem e os LMS são utilizados no contexto deste trabalho como as principais tecnologias educacionais envolvidas no processo de adaptação do conteúdo instrucional. Estes objetos seguem alguns padrões de referência, como o padrão de metadados IEEE-LOM e o modelo de integração SCORM.

O modelo de seleção dinâmica de objetos de aprendizagem proposto nesta dissertação é baseado na abordagem multiagente. Estes agentes seguem a teoria BDI. Foram utilizadas ferramentas para o desenvolvimento do modelo de agentes, como a metodologia Prometheus e a plataforma Jadex.

No decorrer desta dissertação, serão apresentados também questões relacionadas à forma de representação de conhecimento dos agentes do modelo teórico, bem como possibilidades de evolução ao modelo proposto, no que se refere à busca semântica por objetos de aprendizagem em repositórios.

### 2.1 OBJETOS DE APRENDIZAGEM

Os objetos de aprendizagem são recursos digitais que auxiliam os professores e designers instrucionais no contexto educacional. No modelo proposto nesta dissertação, os objetos de aprendizagem são as unidades instrucionais mais básicas onde a adaptação ocorre, e onde o reuso é favorecido.

Objetos de Aprendizagem (OA) são considerados artefatos digitais que apoiam o processo de ensino-aprendizagem, com o objetivo de facilitar o aprendizado de conceitos por parte dos alunos, com forte ênfase na interoperabilidade e reusabilidade (SANTOS *et al.*, 2014; SILVA *et al.*, 2012).

O Comitê de Padrões de Tecnologias Educacionais (em inglês, LTSC) do IEEE, formado com o intuito de definir e promover padrões tecnológico-instrucionais (WILEY, 2000, p. 4-5), define objetos de

aprendizagem (OA) como “qualquer entidade, digital ou não-digital, que pode ser usada para ensino, aprendizagem ou formação”.

Os padrões definidos pelo referido comitê visam assegurar a interoperabilidade das tecnologias instrucionais desenvolvidas por diferentes universidades e organizações, principalmente dos seus objetos de aprendizagem, visando facilitar a adoção mundial da abordagem de OA. Além disso, projetos e organizações surgiram, como o IMS e a ADL, contribuindo com o desenvolvimento de padrões técnicos que apoiam à ampla disseminação dos objetos de aprendizagem (WILEY, 2000).

Segundo Wiley (2000), Objetos de Aprendizagem são elementos de uma tecnologia instrucional baseada no paradigma de Orientação a Objetos (OO) da Ciência da Computação, permitindo que os designers instrucionais (DI) criem componentes menores (relativo ao tamanho do curso como um todo) que podem ser reutilizados diversas vezes, e em múltiplos contextos de aprendizagem.

De acordo com McGreal (2004), os OA apresentam certas características de destaque, tais como: interoperabilidade, durabilidade, reusabilidade, adaptabilidade, entre outras. Os objetos de aprendizagem podem ser compostos por textos, imagens, vídeos, áudios, animações, software, sendo descritos por um padrão de metadados (ex.: LTSC/IEEE-LOM, 2002), podendo ainda ser desenvolvidos segundo um padrão de empacotamento e sequenciamento (ex.: SCORM). Um ou mais OA podem ser combinados para a criação de objetos maiores, como aulas ou cursos (NASH, 2005).

Gluz e Vicari (2011, p. 204) afirmam que

a tecnologia de Objetos de Aprendizagem (OA) fundamenta-se na hipótese de que é possível criar componentes de material pedagógico e organizá-los de forma a possibilitar sua reutilização, promovendo uma economia de tempo e de custo na produção deste tipo de material.

Souza e Alves (2012, p. 221) mencionam que embora seja possível criar OA sem informações de metadados e reutilizá-los desta forma, que

isso demanda tempo e modificações que podem ser evitadas, porque as ligações entre o ambiente e o objeto digital de aprendizagem precisam ser

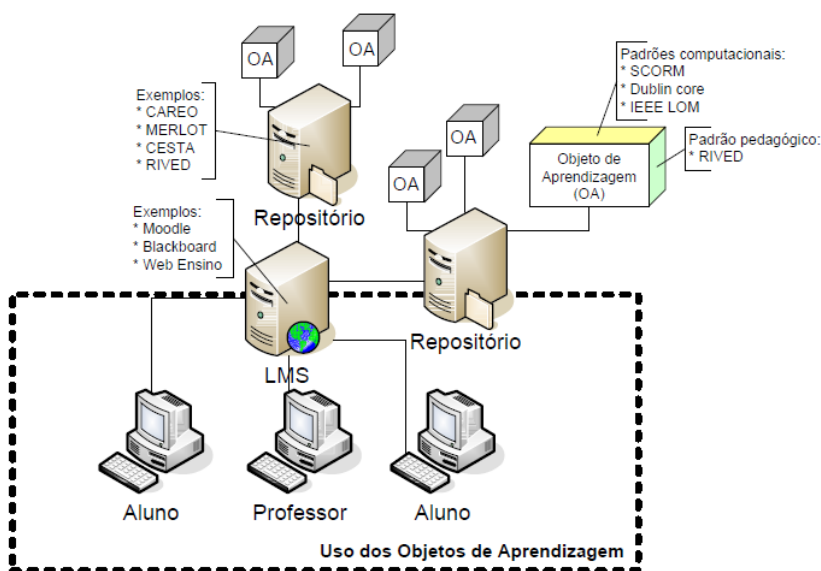


alteradas, para que a comunicação seja bem-sucedida.

Já se os objetos de aprendizagem forem construídos utilizando um modelo de referência, estes poderão ser reutilizados nas plataformas e ambientes confeccionados nesse modelo.

Em geral, os objetos de aprendizagem não são utilizados isoladamente, e sim integrados a outras ferramentas tecnológicas como os repositórios de OA e os ambientes virtuais de aprendizagem ou sistemas de gerenciamento de aprendizagem (LMS). A Figura 1 ilustra a integração destas entidades, cujos conceitos serão abordados com maiores detalhes no decorrer deste capítulo.

Figura 1 – Relacionamento entre OA, Repositórios e LMS



Fonte: Silva (2007)

## 2.2 PADRÕES DE OBJETOS DE APRENDIZAGEM

A utilização de padrões no desenvolvimento de objetos de aprendizagem favorece a recuperação, reusabilidade e interoperabilidade destes materiais educacionais. A adoção de padrões facilita a produção de

cursos on-line através de objetos de aprendizagem, pelo fato de permitir a localização de objetos pré-existentes em repositórios, e de possibilitar que sejam compostos novos objetos com base em OA diferentes.

Além disso, a utilização de objetos de aprendizagem compatíveis com padrões de recursos educacionais largamente difundidos potencializa o emprego destes materiais em diferentes tipos de ambientes virtuais de aprendizagem, favorecendo a interoperabilidade.

De acordo com Dias *et al.* (2009), os padrões de objetos de aprendizagem são divididos de acordo com suas funcionalidades em: (1) padrões de metadados; (2) padrões de empacotamento (conteúdo); (3) padrões de interface e comunicação (sequenciamento); e (4) padrões de integração.

### **2.2.1 Padrões de Metadados**

Metadados são estruturas de informação que descrevem recursos (físico ou digital), permitindo a identificação, catalogação e gerenciamento destes, e facilitando sua localização e recuperação, inclusive por aplicações computacionais. Metadados são muitas vezes denominados como dados sobre dados (CAMPOS, 2013; VICARI *et al.*, 2010).

Da perspectiva das tecnologias educacionais, metadados são elementos que descrevem dados sobre os objetos de aprendizagem, possibilitando sua busca e localização em repositórios, bem como facilitando sua utilização em sistemas de gerenciamento de aprendizagem (LMS).

Conforme mencionado na especificação IEEE-LOM (2002), uma instância de metadado descrevendo um objeto de aprendizagem pode ser utilizada por uma ferramenta de tecnologia educacional para gerenciar, localizar, obter, usar, avaliar e/ou intercambiar este OA, inclusive através de um processo automatizado.

#### **2.2.1.1 IEEE-LOM**

O *LTSC/IEEE Learning Object Metadata* (LOM), um dos padrões de metadados de objetos de aprendizagem mais aceitos mundialmente (VICARI *et al.*, 2010), torna possível a obtenção de informações sobre como usar o objeto e em que condições (CARVALHO *et al.*, 2014). O padrão LOM especifica um esquema de dados conceituais que define a

estrutura de uma instância de metadado para um objeto de aprendizagem (IEEE-LOM, 2002).

As características do OA definida pela instância do metadado no padrão IEEE-LOM podem ser agrupadas nas seguintes categorias:

- Geral (*general*) – Descreve o objeto como um todo. Define elementos tais como título (*title*), descrição (*description*), palavras-chave (*keywords*), entre outros.
- Ciclo de vida (*life cycle*) – Elementos relacionados ao histórico e o estado atual do objeto, como por exemplo a versão (*version*) em que se encontra.
- Meta-metadados (*meta-metadata*) – Define informações sobre a própria instância do metadado, e não sobre o OA em si.
- Técnico (*technical*) – Agrupa os requisitos e características técnicas do objeto de aprendizagem. Tamanho (*size*), formato (*format*) e duração (*duration*) são exemplos de elementos definidos por esta categoria.
- Educacional (*educational*) – Comporta as características educacionais e pedagógicas do OA, definindo elementos como: o tipo e o nível de interatividade (*interactivity type* e *interactivity level*), o tipo do recurso de aprendizagem (*learning resource type*), o contexto de aprendizagem (*context*) e o grau de dificuldade (*difficulty*).
- Direitos (*rights*) – Elementos relacionados à propriedade intelectual e permissões de utilização.
- Relações (*relation*) – Define relacionamentos entre o objeto de aprendizagem e outros OA, como por exemplo, se o objeto é parte de outro, ou se referencia outro OA (de acordo com os valores dos elementos *kind* e *resource:identifier*).
- Anotações (*annotation*) – Permite a definição de comentários sobre o uso educacional do objeto de aprendizagem.
- Classificação (*classification*) – Descreve o OA em relação a um sistema de classificação taxonômica particular, o que permite a definição de pré-requisitos ou hierarquia de assuntos, por exemplo. Destacam-se os elementos: propósito (*purpose*),

caminho taxonômico (*taxon path*), fonte (*source*) e unidade taxonômica (*taxon*).

Cada elemento descrito permite a definição de valores correspondentes, podendo ser textos livres, ou valores pré-definidos. Como exemplo de elementos com valores previamente definidos, podem ser destacados: (1) tipo de interatividade (“ativo”, “expositivo”, “misto”); (2) tipo de recurso (“exercício”, “diagrama”, “figura”, “gráfico”, “tabela”, etc.); e (3) nível de dificuldade (“fácil”, “médio”, “difícil”, etc.).

Segundo citado por Pereira *et al.* (2003, p. 10), ao especificar um esquema comum de dados conceituais, o padrão de metadados IEEE-LOM também assegura “que *bindings* (representações desses metadados através de linguagens conhecidas, como XML, por exemplo) entre metadados de LOs terão melhor grau de interoperabilidade semântica” (LO é a sigla em inglês para Objetos de Aprendizagem).

Embora seja apontado como um modelo completo, contudo, cabe ressaltar que o IEEE-LOM também é considerado de difícil preenchimento devido a sua extensão (mais de 40 elementos) (VICARI *et al.*, 2010).

### 2.2.1.2 *Dublin Core*

Assim como o IEEE-LOM, outro padrão de metadados mundialmente difundido é o *Dublin Core*, o qual é mantido pelo *Dublin Core Metadata Initiative* (DUBLIN CORE, 2015), e que surgiu com o objetivo inicial de descrever conteúdo baseado na *web* (CAMPOS, 2013). O *Dublin Core* é considerado mais conciso, apresentando apenas quinze elementos, incluindo os mais utilizados pelos professores e designers instrucionais (VICARI *et al.*, 2010).

Os elementos definidos pelo padrão *Dublin Core* são (em inglês): (1) *Title*, (2) *Creator*, (3) *Subject*, (4) *Description*, (5) *Publisher*, (6) *Contributor*, (7) *Date*, (8) *Type*, (9) *Format*, (10) *Identifier*, (11) *Source*, (12) *Language*, (13) *Relation*, (14) *Coverage*, (15) *Rights*.

Pelo fato dos seus elementos serem abrangentes e genéricos, o *Dublin Core* pode ser usado para descrever qualquer tipo de recurso digital, e não somente objetos de aprendizagem. O padrão inclui noções de melhores práticas em *Web Semântica*, apresentando evolução em suas versões no decorrer do tempo, incluindo a atribuição de domínios e faixas de valores (classes) em linguagem natural (DUBLIN CORE, 2015).

Campos (2013) ressalta que existem divergências de entendimento entre a abrangência do padrão, de um lado os que defendem um conjunto

mínimo e simplificado de elementos, e de outro lado aqueles que comungam de uma perspectiva estruturada mais refinada e extensa.

### 2.2.1.3 OBAA

O padrão de metadados OBAA foi desenvolvido através do projeto homônimo envolvendo pesquisadores das universidades UFRGS e UNISINOS, com o objetivo de prover interoperabilidade entre diferentes plataformas de conteúdo digital no contexto educacional brasileiro.

Segundo Vicari *et al.* (2010, p. 2),

a proposta de um novo padrão para a descrição de objetos de aprendizagem interoperáveis para ambientes educativos surgiu a partir da realidade de uso cada vez mais constante da *Web* e da perspectiva que isso se estenda à TV-Digital e aos dispositivos móveis.

O padrão OBAA foi concebido como uma extensão do padrão de metadados educacionais IEEE-LOM, adicionando-se especificações relacionadas à catalogação de arquivos multimídia e TV-Digital, além de elementos para interoperabilidade na *Web Semântica* (VICARI *et al.*, 2010).

## 2.2.2 Padrões de Empacotamento

Padrões de empacotamento (conteúdo), por sua vez, são desenvolvidos com o intuito de agrupar em um mesmo arquivo (pacote), e de forma padronizada, todos os recursos digitais (arquivos) que compõem um objeto de aprendizagem.

Segundo Dutra e Tarouco (2006 *apud* DIAS *et al.*, 2009, p. 5),

o empacotamento de um conteúdo consiste no processo de agregação de múltiplos artefatos digitais pertencentes a um objeto, de uma determinada unidade de aprendizagem, em um único local, gerando um arquivo compactado, denominado de pacote. O padrão de empacotamento compacta em um único arquivo o

conteúdo de objetos complexos, ou seja, de um OA formado por um conjunto de dois ou mais arquivos.

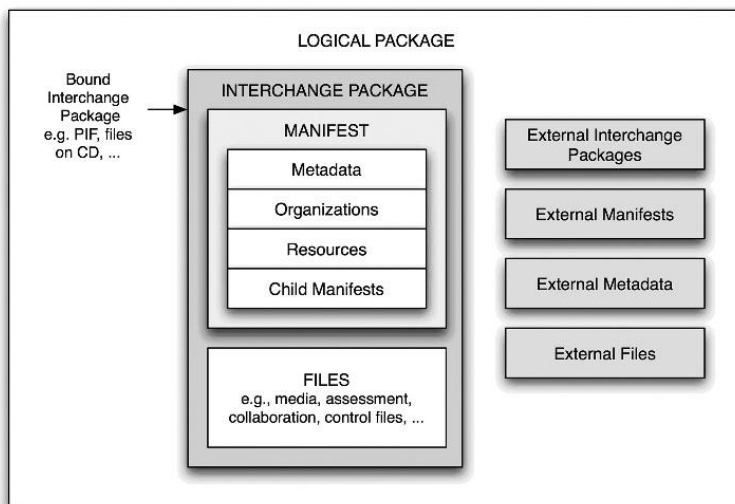
Dentre os padrões de empacotamento de objetos de aprendizagem, destaca-se o *IMS Content Packaging* (IMS CP), o qual foi desenvolvido pelo *IMS Global Learning Consortium*. A especificação IMS CP define estruturas de dados para compartilhamento de conteúdo educacional entre sistemas de gerenciamento de aprendizagem (LMS) e/ou repositórios digitais, possibilitando a importação, exportação, agregação e desagregação de pacotes de conteúdo (IMS CP, 2015).

Através desta especificação, é possível obter a descrição das mídias (recursos estáticos) que compõem o pacote, juntamente com detalhes da estruturação dos conteúdos, e a sinalização de qual página *web* (HTML) deve ser exibida primeiro (IMS CP, 2015).

A Figura 2 ilustra o modelo conceitual dos elementos que compõem um pacote (geralmente um arquivo ZIP) segundo a especificação IMS CP. O pacote é formado pelos recursos digitais que fazem parte do objeto de aprendizagem em si, e também por um arquivo da tecnologia XML (*eXtensible Markup Language*), denominado *manifest.xml*, que descreve o que o pacote contém e como estes conteúdos

são organizados (WILSON e CURRIER, 2002; DIAS *et al.*, 2009; IMS CP, 2015).

Figura 2 – Modelo conceitual da especificação *IMS Content Packaging*



Fonte: *IMS Global Learning Consortium* (2015)

### 2.2.3 Padrões de Interface e Comunicação

Padrões de interface e comunicação (sequenciamento) permitem que um objeto de aprendizagem possa ser acessado através de um ambiente virtual de aprendizagem (como o Moodle), coordenando a execução deste OA no LMS conforme o aluno navega pelo conteúdo do objeto (DIAS *et al.*, 2009).

Na sequência serão descritos alguns dos principais padrões de interface e comunicação de objetos de aprendizagem mencionados em pesquisas da área da informática na educação.

#### 2.2.3.1 AICC

O AICC – *Aviation Industry CBT (Computer-Based Training) Committee* foi uma associação existente entre 1988 e 2014, e que definiu

várias especificações para tecnologias educacionais (AICC, 2014). Estas especificações são adotadas pelas principais ferramentas largamente utilizadas pela indústria de *e-learning*, por exemplo: Adobe Captivate (CAPTIVATE, 2015), Adobe Connect (CONNECT, 2015), Skillsoft (SKILLSOFT, 2015) e Articulate (ARTICULATE, 2015).

Dentre as especificações AICC, destaca-se: (1) o protocolo para comunicação HTTP entre o LMS e o objeto de aprendizagem (conteúdo), possibilitando o monitoramento do progresso do estudante, e (2) a forma como o conteúdo (OA) deve ser disponibilizado no LMS (AICC, 2014).

#### 2.2.3.2 IMS LTI

O IMS *Learning Tools Interoperability* (LTI) é uma especificação desenvolvida pela IMS *Global Learning Consortium* que permite a integração de conteúdo remoto e aplicações educacionais fornecidas por serviços externos (*Tool Providers*, acessíveis via protocolo HTTP) com plataformas como os LMS ou os repositórios de objetos de aprendizagem (*Tools Consumers*) (IMS LTI, 2015; JURADO e REDONDO, 2014).

O IMS LTI permite o acesso transparente e seguro a outras aplicações educacionais a partir do ambiente virtual de aprendizagem, através do navegador *web* (*browser*) do estudante, beneficiando: (1) os desenvolvedores de outras aplicações educacionais (por não ser necessário conhecer a estrutura de cada LMS); (2) os administradores do ambiente virtual (por não precisar ser feita a instalação do software externo); e (3) os professores (pelo fato de poderem selecionar e conectar



facilmente ao LMS aplicações e recursos externos que melhor atendem as necessidades de seus alunos) (VICKERS, 2012).

## 2.2.4 Padrões de Integração

Segundo Dutra e Tarouco (2006, DIAS *et al.*, 2009, p. 7), um padrão de integração

unifica em um modelo de referência diferentes tipos de padrões, tais como padrões de metadados, empacotamento, interface e comunicação, desenvolvidos por outras organizações.

Os principais padrões de integração existentes na indústria de tecnologias educacionais, e amplamente citados na literatura, serão descritos na sequência.

### 2.2.4.1 SCORM

O modelo de referência para objetos de conteúdo compartilhável SCORM (*Sharable Content Object Reference Model*) (ADL, 2004) é um padrão de agregação, sequenciamento e execução de Objetos de Aprendizagem em cursos on-line (RODRIGUES *et al.*, 2009), definindo a estrutura de empacotamento de recursos instrucionais e os protocolos (API) de comunicação com os principais LMS.

O padrão SCORM é amplamente utilizado na educação on-line em diversos países, desde o seu surgimento em 2001. O referido modelo foi desenvolvido pela ADL, uma iniciativa do Departamento de Defesa dos Estados Unidos, com a finalidade de promover a unificação entre diferentes padrões técnicos de recursos educacionais, a saber:

- Metadados – com uma extensão e adaptação do padrão IEEE-LOM.
- Empacotamento – estendendo e adaptando o padrão IMS CP.
- Interface e Comunicação – através de uma derivação do AICC.

Os conteúdos produzidos segundo o padrão SCORM são empacotados em um arquivo compactado (ZIP). Pacotes SCORM podem ser disponibilizados e acessados pelos estudantes em qualquer LMS que forneça compatibilidade com este modelo. Os principais LMS comerciais

e de código aberto (como o Moodle) implementam a especificação SCORM. Assim, estes LMS conseguem identificar: (1) como o conteúdo do pacote SCORM está organizado; (2) a sequência em que os itens (denominados SCO) que compõem o objeto devem ser apresentados; e (3) as ações do aluno referente à sua interação com o conteúdo do pacote SCORM (DIAS *et al.*, 2009).

Dutra (2008) e Silva (2007), mencionam os requisitos arquiteturais de alto nível que norteiam o desenvolvimento do modelo SCORM:

- Acessibilidade – o conteúdo pode ser localizado e acessado de diferentes locais.
- Interoperabilidade – o recurso pode ser acessado de diferentes plataformas educacionais, sistemas operacionais e navegadores *web*.
- Durabilidade – não deve ser necessário reconfigurar o conteúdo instrucional devido a mudanças e atualizações em sistemas e plataformas educacionais.
- Reusabilidade – os conteúdos instrucionais devem poder facilmente ser incorporados e agrupados em diferentes contextos de aprendizagem.

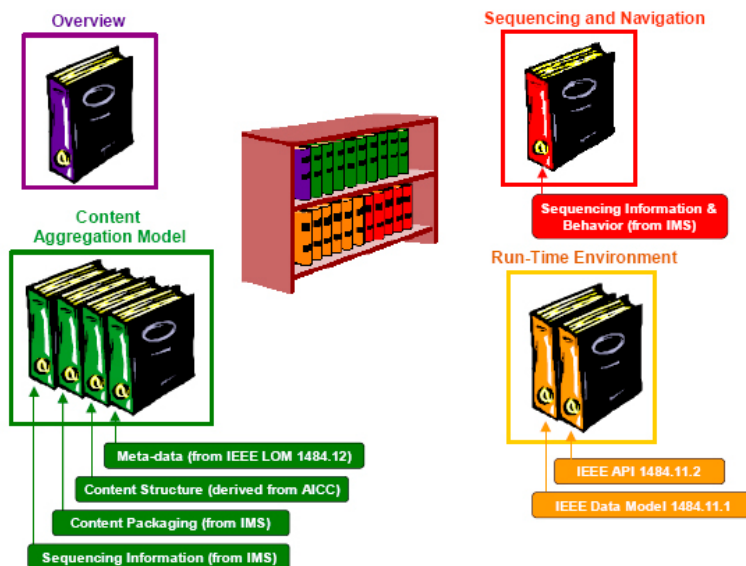
Souza e Alves (2012) citam alguns dos benefícios de se adotar o padrão SCORM: (1) independência de plataforma de ambiente virtual de aprendizagem (AVA) ou sistema de gerenciamento de aprendizagem (LMS); facilidade de migração do conteúdo entre diferentes plataformas; (2) possibilidade de comprar (ou receber gratuitamente) cursos de terceiros, e ainda combinar unidades de ensino produzidas pela própria instituição ou externamente; (3) economia de tempo e custos na integração do conteúdo ao LMS; e (4) capacidade de se acompanhar o desempenho dos alunos, através dos relatórios implementados no LMS que possuem como base os dados obtidos do SCORM.

A versão mais recente do modelo SCORM (SCORM 2004 *4<sup>th</sup> Edition*) é composta por um conjunto de quatro especificações, as quais foram publicadas nos seguintes livros: (1) Visão Geral (*Overview*); (2) CAM (*Content Aggregation Model*); (3) RTE (*Run-Time Environment*); e (4) SN (*Sequencing and Navigation*).

O documento *Overview* descreve as motivações, os requisitos de alto nível e outros detalhes sobre o modelo SCORM. As especificações

restantes serão detalhadas na sequência deste capítulo. A Figura 3 ilustra as especificações que compõem a arquitetura SCORM.

Figura 3 – Especificações da arquitetura SCORM



Fonte: SCORM 2004 2<sup>nd</sup> Edition Overview (ADL, 2009b)

#### 2.2.4.1.1 CAM (Content Aggregation Model)

O modelo de agregação de conteúdo (em inglês, *Content Aggregation Model*) define a estrutura de recursos que compõem o pacote SCORM (arquivo ZIP). Um objeto de aprendizagem segundo o modelo SCORM é composto por recursos físicos e declarações de metadados que definem a estrutura completa do objeto. Estes metadados são descritos através da linguagem XML, em arquivos denominados “manifesto”, mais precisamente *manifest.xml* (SOUZA e ALVES, 2012).

Os componentes físicos elementares de um objeto SCORM são denominados *assets* (textos, imagens, sons, arquivos HTML, etc.). Estes

tipos de recursos não podem se comunicar com o LMS, porém são os componentes que apresentam a maior capacidade de reutilização.

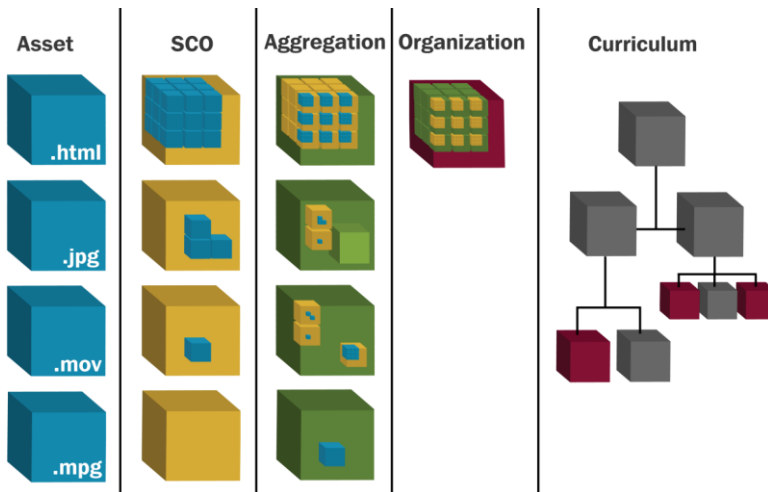
A composição de *assets* possibilita a criação das unidades lógicas mais básicas que podem ser disponibilizadas em um LMS, denominados SCO (*Sharable Content Objects*). Os SCO são os componentes que utilizam a SCORM API para se comunicar com o sistema de gerenciamento de aprendizagem (LMS). Cada SCO possui o próprio arquivo de manifesto.

Existe a possibilidade de se agrupar os SCO, para que sejam criadas coleções de atividades educacionais relacionadas (*aggregations*). Este agrupamento pode conter outras *aggregations*. Esta agregação é declarada através de um arquivo manifesto.

Por fim, os componentes definidos no modelo de agregação de conteúdo SCORM podem ser ordenados em uma estrutura de árvore, de tal forma que sejam atribuídos comportamentos de sequenciamento aos SCO. Esta configuração é realizada através dos elementos do tipo *organization*, definidos no respectivo arquivo *manifest.xml* (ADL, 2011).

A estrutura de possíveis componentes de um objeto de aprendizagem segundo o modelo SCORM é demonstrada na Figura 4.

Figura 4 – Componentes do conteúdo SCORM



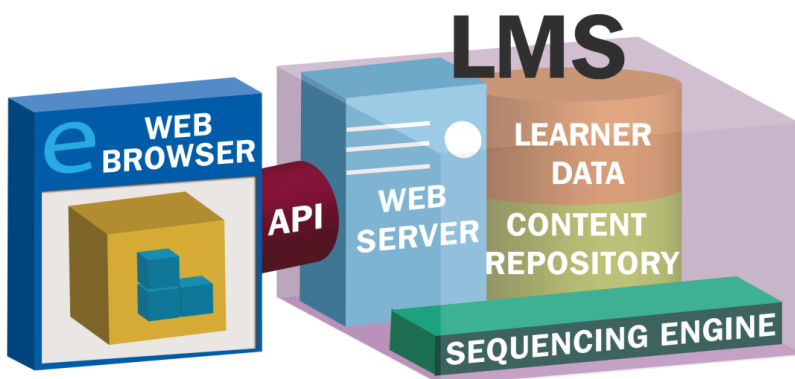
### 2.2.4.1.2 RTE (*Run-Time Environment*)

O ambiente de execução SCORM permite a troca de informações entre o conteúdo e o sistema de gerenciamento de aprendizagem (LMS). De acordo com Silva (2007, p. 15),

a proposta do Ambiente de Execução (*Run-Time Environment*) é formalizar os mecanismos para promoção da interoperabilidade entre o LMS e os objetos de aprendizagem no padrão SCORM.

A Figura 5 ilustra a ligação de comunicação entre o conteúdo do objeto SCORM (SCO) e o LMS.

Figura 5 – O link de comunicação entre um SCO e um LMS



Fonte: SCORM 2004 4<sup>th</sup> Edition (ADL, 2011b)

Segundo Dutra (2008), a especificação SCORM RTE define três elementos que possibilitam a comunicação SCO-LMS:

- Mecanismo de execução (*Launch*) – define uma forma comum para os LMS iniciarem e navegarem sobre os objetos de aprendizagem. O mecanismo estabelece procedimentos e responsabilidades para a comunicação dos SCO com o LMS, através do navegador *web* do aluno. A padronização da comunicação é garantida pela definição de uma API comum, independentemente da plataforma educacional utilizada (LMS).
- API (*Application Programming Interface*) – os sistemas de gerenciamento de aprendizagem que desejarem ser compatíveis com o modelo SCORM devem implementar um adaptador da API SCORM (implementação na linguagem JavaScript). As funções definidas na referida API permitem que o LMS seja informado sobre o estado de execução do recurso de aprendizagem, e que sejam trocados dados entre o LMS e os SCO, dados estes referentes à interação do aluno com o objeto (ex.: nome do

estudante, tempo da sessão de aprendizagem, em que parte do objeto o aluno interrompeu a última sessão, etc.).

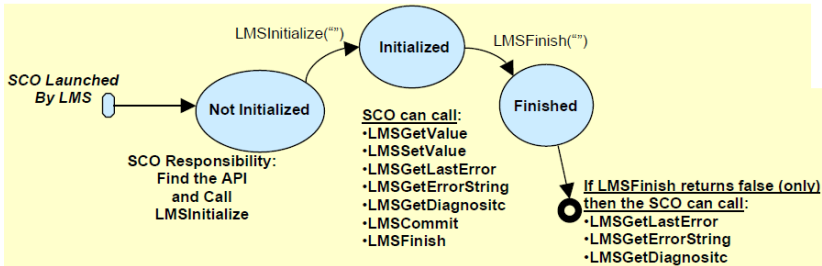
O Quadro 1 categoriza os métodos especificados na API SCORM.

Quadro 1 – Categorização das funções definidas na API SCORM

<b>Categoria</b>	<b>Descrição da Categoria</b>	<b>Função</b>	<b>Descrição da Função</b>
Funções de Sessão	Funções para inicialização e término de sessões de comunicação entre SCO e LMS.	<i>Initialize</i>	Usada para iniciar uma sessão de comunicação entre o SCO e o LMS.
		<i>Terminate</i>	Usada pelo SCO para finalizar uma sessão de comunicação com o LMS.
Funções de Transferência de Dados	Funções usadas para a troca de valores referentes a elementos do modelo de dados entre um SCO e um LMS, através de uma instância da API SCORM (adaptador).	<i>GetValue</i>	Permite que o SCO obtenha uma informação do LMS (referente a algum elemento do modelo de dados).
		<i>SetValue</i>	Permite que o SCO envie uma informação ao LMS (um valor para algum dos elementos do modelo de dados)
		<i>Commit</i>	Permite que o SCO solicite ao LMS a persistência das informações previamente enviadas na sessão de aprendizagem.
Funções de Apoio	Funções auxiliares para tratamento de erros.	<i>GetLastError</i>	Obtém o código do último erro encontrado (caso exista).
		<i>GetErrorString</i>	Fornece uma mensagem textual sobre o código de erro informado.
		<i>GetDiagnostic</i>	Fornece informações adicionais sobre o estado de erro atual.

A Figura 6 exibe o diagrama de transição de estados referente a comunicação entre os SCO e o LMS, através das funções da API.

Figura 6 – O diagrama de estados do SCORM RTE API



Fonte: SCORM 1.2 (ADL, 2001)

- Modelo de Dados (*Data Model*) – define um conjunto padrão de elementos do modelo de dados que são usados para monitorar a interação do estudante com um SCO, tais como: situação de completude, a medida de progresso ou o resultado de um questionário (*quiz*). O LMS deve manter os dados entre sessões de aprendizagem para cada aluno, e o SCO deve utilizar apenas estes elementos pré-definidos para garantir o reuso entre sistemas. O Quadro 2 lista os elementos do modelo de dados SCORM.

Quadro 2 – Elementos do Modelo de Dados SCORM

Elemento do Modelo de Dados	Notação do elemento ( <i>binding</i> )	Descrição
<i>Comments From Learner</i>	cmi.comments_from_learner	Contém comentários e textos informados pelo aluno.
<i>Comments From LMS</i>	cmi.comments_from_lms	Contém comentários e textos a serem exibidos ao aluno.
<i>Completion Status</i>	cmi.completion_status	Indica se o aluno completou o SCO.
<i>Completion Threshold</i>	cmi.completion_threshold	Indica o valor sobre o qual a medida de progresso do aluno será comparada (para definir se o SCO deve ou não ser considerado completo).
<i>Credit</i>	cmi.credit	Indica se o aluno receberá créditos pelo seu desempenho no SCO.



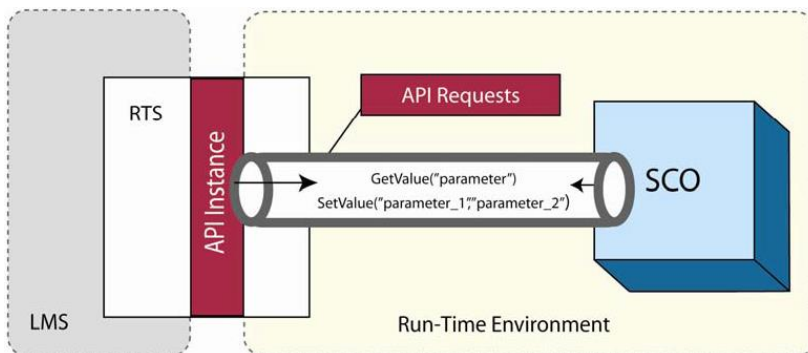
<b>Elemento do Modelo de Dados</b>	<b>Notação do elemento (<i>binding</i>)</b>	<b>Descrição</b>
<i>Entry</i>	cmi.entry	Contém informações que comprovam o acesso prévio do aluno ao SCO.
<i>Exit</i>	cmi.exit	Indica como e por que o aluno deixou o SCO.
<i>Interactions</i>	cmi.interactions.n	Define informações referentes às interações (ex.: questionários) com o intuito de medição ou avaliação.
<i>Interaction Weighting</i>	cmi.interactions.n.weighting	Peso da interação em relação a outras interações.
<i>Interaction Result</i>	cmi.interactions.n.result	Resultado da resposta do aluno na interação.
<i>Interaction Latency</i>	cmi.interactions.n.latency	Tempo decorrido entre o início da interação e a resposta do aluno.
<i>Launch Data</i>	cmi.launch_data	Fornecer dados específicos para a inicialização de um SCO.
<i>Learner Id</i>	cmi.learner_id	Identifica o código do aluno que está executando o SCO.
<i>Learner Name</i>	cmi.learner_name	Representa o nome do aluno que está executando o SCO.
<i>Learner Preference</i>	cmi.learner_preference	Indica as preferências do aluno quanto à utilização dos SCO.
<i>Location</i>	cmi.location	Representa a localização do aluno em um SCO.
<i>Maximum Time Allowed</i>	cmi.max_time_allowed	Indica o total de tempo acumulado que é permitido ao aluno durante o uso do SCO em cada tentativa.
<i>Mode</i>	cmi.mode	Identifica o modo como o SCO pode ser apresentado ao aluno.
<i>Objectives</i>	cmi.objectives	Especifica os objetivos de aprendizagem ou desempenho associados com um SCO.
<i>Progress Measure</i>	cmi.progress_measure	Identifica a medida de progresso atual do aluno durante a navegação pelo SCO.
<i>Scaled Passing Score</i>	cmi.scaled_passing_score	Identifica a escala de pontuação para aprovação em um SCO.
<i>Score</i>	cmi.score	Identifica a pontuação do aluno no SCO.
<i>Session Time</i>	cmi.session_time	Identifica a quantidade de tempo despendida pelo aluno na sessão atual do SCO.
<i>Success Status</i>	cmi.success_status	Identifica se o aluno completou com sucesso o SCO.

Elemento do Modelo de Dados	Notação do elemento ( <i>binding</i> )	Descrição
<i>Suspend Data</i>	cmi.suspend_data	Fornecer informação do resultado do acesso anterior do aluno ao SCO.
<i>Time Limit Action</i>	cmi.time_limit_action	Indica o que o SCO deve fazer quando o tempo máximo permitido é excedido.
<i>Total Time</i>	cmi.total_time	Identifica o tempo total acumulado de todas as sessões anteriores na tentativa atual do aluno no SCO.

Fonte: SCORM 2004 4<sup>th</sup> Edition (ADL, 2009a)

A Figura 7 exemplifica o uso dos elementos do modelo de dados com a API SCORM na comunicação entre um SCO e um LMS, durante a sessão de aprendizagem de um aluno.

Figura 7 – Uso do modelo de dados com a API SCORM



Fonte: SCORM 4<sup>th</sup> Edition (ADL, 2009a)

#### 2.2.4.1.3 SN (Sequencing and Navigation)

A especificação SN define maneiras de se encadear a exibição dos recursos que compõem o objeto SCORM, através de regras de sequenciamento. Cabe ressaltar que não existia na versão anterior (SCORM 1.2), sendo adicionada na versão 2004 (SILVA, 2007; DUTRA, 2008).

O livro SCORM SN (ADL, 2009) descreve como os recursos SCORM podem ser disponibilizados aos alunos, com base em um

conjunto de eventos de navegação disparados pelo estudante ou pelo sistema. Os fluxos e ramificações de exibição do conteúdo são configurados (em tempo de produção) pelos desenvolvedores de conteúdo (Designers Instrucionais) através de um conjunto pré-definido de atividades e regras, com as respectivas condições de satisfação. O mecanismo de comunicação entre o LMS e o conteúdo SCORM facilita a utilização das regras de sequenciamento e navegação, possibilitando a oferta de conteúdo aos alunos com base em suas escolhas ou desempenho (em tempo de execução).

A Figura 8 exibe um exemplo da sintaxe referente a regras de sequenciamento, definidas em um arquivo de manifesto de um conteúdo SCORM.

Figura 8 – Exemplo da sintaxe de regras de sequenciamento

```
<imsss:sequencingRules>
  <imsss:preConditionRule>
    <imsss:ruleConditions conditionCombination="all|any">
      <imsss:ruleCondition condition="someCondition" />
      <imsss:ruleCondition condition="someOtherCondition" />
    </imsss:ruleConditions>
    <imsss:ruleAction action=
      "skip|disabled|hiddenFromChoice|stopForwardTraversal" />
  </imsss:preConditionRule>
  <imsss:postConditionRule>
    <imsss:ruleConditions conditionCombination="all|any">
      <imsss:ruleCondition condition="someCondition" />
    </imsss:ruleConditions>
    <imsss:ruleAction action=
      "exitParent|exitAll|retry|retryAll|continue|previous" />
  </imsss:postConditionRule>
  <imsss:exitConditionRule>
    <imsss:ruleConditions conditionCombination="all|any">
      <imsss:ruleCondition condition="someCondition" />
    </imsss:ruleConditions>
    <imsss:ruleAction action="exit" />
  </imsss:exitConditionRule>
</imsss:sequencingRules>
```

Fonte: SCORM 4<sup>th</sup> Edition (ADL, 2011a)

#### 2.2.4.2 IMS Common Cartridge

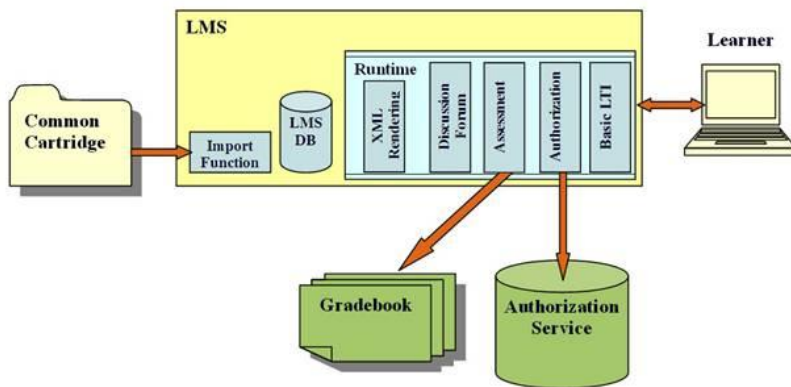
Segundo a classificação de Dias *et al.* (2009), como alternativa ao modelo SCORM, o IMS *Common Cartridge* (IMS GLC, 2013) é um conjunto de padrões abertos desenvolvidos pelo *IMS Global Learning Consortium* com o intuito de promover interoperabilidade, reusabilidade

e personalização de conteúdo digital, avaliações, fóruns colaborativos e outros tipos de aplicações educacionais baseadas na *web* (GONZALEZ-BARBONE e ANIDO-RIFON, 2010).

O IMS *Common Cartridge* especifica: (1) um formato para troca de conteúdo entre sistemas; (2) um padrão de metadados descrevendo o conteúdo empacotado como um “cartucho” (*cartridge*) (*IMS Learning Object Metadata*); (3) um padrão para testes e avaliações (*IMS Question and Test Interoperability*); (4) um padrão para execução e troca de dados com aplicações externas, através de um sistema de aprendizagem (*IMS Learning Tools Interoperability*); (5) um vocabulário para designar a utilização pretendida do conteúdo empacotado no *cartridge*; (6) um esquema de preenchimento de fóruns on-line para colaboração entre os estudantes; e (7) um esquema para preenchimento de links *web* (*IMS GLC*, 2013).

A Figura 9 apresenta a visão geral da estrutura do *Common Cartridge*.

Figura 9 – Visão geral da estrutura do *Common Cartridge*



Fonte: IMS CC (IMS GLC, 2013)

Segundo Vieira *et al.* (2012), embora o padrão *Common Cartridge* ainda não possua maturidade e difusão semelhante a outros modelos, como o SCORM, esta tecnologia mostra-se promissora, principalmente

“por favorecer a aprendizagem colaborativa, o reuso e a interoperabilidade entre diferentes LMS de uma forma mais ampla”.

#### 2.2.4.3 ADL *Training and Learning Architecture*

Embora o modelo SCORM seja amplamente difundido e compatível com as principais ferramentas de autoria de conteúdo instrucional, bem como com os principais LMS, o SCORM é menos abrangente e flexível do ponto de vista do processo educacional como um todo. Além disso, o modelo SCORM não possibilita a integração com ferramentas tecnológicas colaborativas mais recentes, como as redes sociais, e com conteúdos externos (VIEIRA *et al.*, 2012).

Com o intuito de avançar nestas questões, a ADL projetou uma nova arquitetura para tecnologias educacionais, idealizada como a evolução do modelo SCORM. A *Training and Learning Architecture* (TLA) engloba um conjunto de especificações de tecnologias destinadas à criação de um ambiente rico para a formação e aprendizagem on-line, baseada em serviços *web* e softwares de código aberto (ADL, 2012).

Os componentes da arquitetura TLA estão alinhados a quatro conceitos potenciais, em uma estrutura modular: (1) monitoramento da experiência de aprendizagem; (2) perfis do aluno (preferências e histórico de cursos); (3) oferta e gerenciamento de conteúdo; e (4) redes de competência (mapeamento de conteúdo/cursos para objetivos/competências) (ADL, 2012).

A fase inicial do projeto TLA tem como foco o monitoramento da experiência de aprendizagem, isto é, a interação do aluno com o conteúdo educacional, e inclui o desenvolvimento de quatro itens: (1) uma nova API de *run-time*; (2) um novo modelo de dados; (3) uma nova sintaxe para o modelo de dados; e (4) um novo método de comunicação.

A especificação Experience API (xAPI), também denominada API TinCan, é uma das partes deste framework, promovendo uma forma interoperável de se descrever e monitorar as experiências em diversos componentes do ecossistema de aprendizagem (HRUSKA *et al.*, 2015).

A API TinCan permite a qualquer ator do processo de ensino-aprendizagem tecnológica armazenar ou recuperar registros de aprendizagem extensíveis, ou de perfis de alunos, independentemente da plataforma utilizada (ADL, 2012).

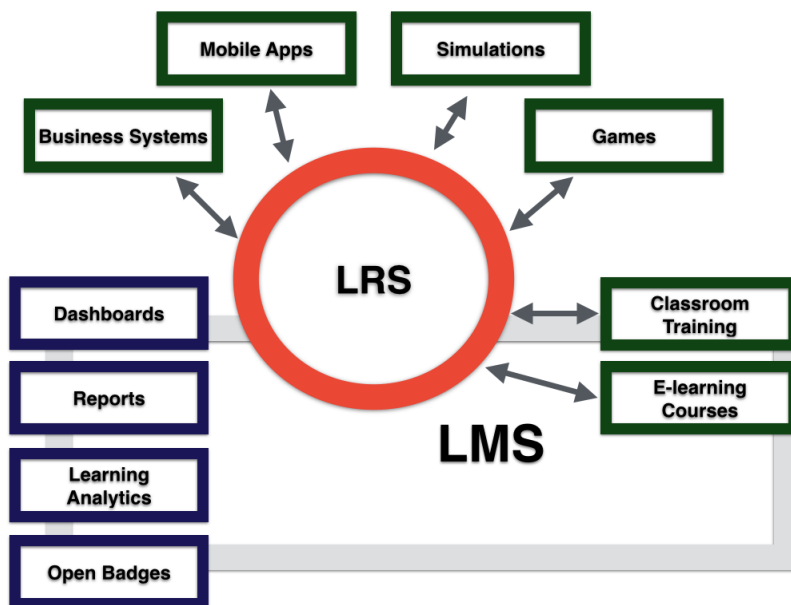
Um dos elementos principais da xAPI é o armazenamento de registros de aprendizagem (*Learning Record Store*). O LRS é um sistema que armazena dados sobre as experiências de aprendizagem, os quais são comunicados através da xAPI por diferentes fontes, denominadas

Provedores de Atividades (*Activity Providers*). O LRS trabalha em conjunto com a xAPI para coletar e retornar estas informações monitoradas, podendo ser integrado ao LMS, e também a outros sistemas externos (ADL, 2012).

Segundo De Nies *et al.* (2015), as formas mais básicas de informação obtidas através da API TinCan são sentenças que descrevem uma ação realizada por um ator (aluno), seguindo a estrutura: “<Ator> <verbo> <objeto>, com <resultado>, no <contexto>”. Um exemplo de sentença seria “*Aluno1 completed Modulo2 e-learning module*”. A especificação da xAPI define que estas experiências educacionais são declaradas no formato JSON (ADL, 2012).

A Figura 10 ilustra o ecossistema de aprendizagem composto de diferentes provedores de atividades integrados ao LMS através dos LRS e da API TinCan.

Figura 10 – Ecossistema de aprendizagem via TinCan



Fonte: TinCan API (TINCAN, 2015)

Embora a xAPI, e toda a arquitetura TLA, sejam padrões promissores, tratam-se de tecnologias ainda em desenvolvimento e que

precisam ser maturadas e difundidas em maior escala entre a comunidade educacional. A quantidade de objetos de aprendizagem desenvolvidos segundo estas abordagens ainda é muito aquém da gama de recursos instrucionais produzidos segundo o modelo SCORM, por exemplo.

Desta forma, objetos empacotados segundo o padrão SCORM continuam sendo comumente utilizados (e reutilizados) em cursos on-line de diferentes entidades (como o CNJ e Tribunais de Justiça, o governo dos Estados Unidos, etc.), o que justifica o emprego desta tecnologia como uma das bases para o modelo apresentado nesta dissertação.

### 2.3 REPOSITÓRIOS DE OBJETOS DE APRENDIZAGEM

Dentre as ferramentas tecnológicas que se integram aos objetos de aprendizagem apresentadas na seção 2.1 deste capítulo, é importante descrever brevemente os Repositórios de Objetos de Aprendizagem. Embora esta tecnologia não seja o foco desta dissertação, trata-se de uma ferramenta auxiliadora do processo de seleção dinâmica de recursos instrucionais, pois os repositórios são o local onde os objetos se encontram disponíveis para que sejam utilizados em um curso no LMS.

Repositórios de objetos de aprendizagem são plataformas de armazenamento, gerenciamento e busca de conteúdos instrucionais digitais reutilizáveis. Estes repositórios são fontes de recursos, onde os objetos podem ser catalogados, encontrados e obtidos pelos Designers Instrucionais (DI) e professores, segundo seus interesses educacionais.

A utilização de repositórios de objetos de aprendizagem surgiu como resultado de pesquisas realizadas por diversas instituições, que buscavam superar algumas dificuldades encontradas na adoção de objetos de aprendizagem, principalmente no que se refere à disseminação e ao reuso de materiais instrucionais na educação on-line (NASCIMENTO *et al.*, 2013).

Os repositórios podem possuir acesso restrito a uma determinada instituição, ou podem ser abertos e acessíveis via *web*. Dentre os repositórios abertos, destacam-se os internacionais: ARIADNE (2015), NDLR (2015), MERLOT (2015), Jorum (JORUM, 2015) e CAREO (2015). Com relação aos repositórios de OA nacionais, podem ser citados: o Banco Internacional de Objetos Educacionais (BIOE, 2015), CESTA (2015), Repositório Digital OBAA (ROBAA, 2015) e RIVED (2015).

Quanto a localização do objeto de aprendizagem (arquivo), os repositórios podem ser classificados em: (1) armazenam localmente tanto o objeto quanto seus metadados; (2) armazenam apenas os metadados dos recursos e fornecem links para outros servidores onde os objetos estão

efetivamente hospedados; e (3) armazenam objetos localmente e também fornecem links para conteúdo externo (híbridos) (MCGREAL, 2008).

## 2.4 SISTEMAS DE GERENCIAMENTO DE APRENDIZAGEM

Os Sistemas de Gerenciamento de Aprendizagem são a ferramenta tecnológica onde ocorre efetivamente a seleção dinâmica de objetos de aprendizagem, proposta nesta dissertação.

Os LMS (em inglês, *Learning Management System*) são Ambientes Virtuais de Aprendizagem (AVA) usados para disponibilização de cursos on-line (ou para disponibilização de materiais), permitindo o monitoramento do progresso do aluno e o gerenciamento dos dados educacionais (KOMLENOV *et al.*, 2010).

Embora AVA e LMS de um ponto de vista educacional mais abrangente não signifiquem o mesmo conceito (PALOMINO, 2013), por motivos de simplificação, no contexto desta dissertação estes termos serão considerados sinônimos. Esta equivalência é comumente adotada na literatura da Informática na Educação, conforme citação de Dias *et al.* (2009, p. 6): “sistemas de gerenciamento de aprendizagem, também conhecidos como ambientes virtuais de aprendizagem”.

De acordo com Palomino (2013, p. 29),

Ambientes virtuais de aprendizagem, em princípio usados basicamente em educação a distância, atualmente servem também de apoio nos cursos presenciais, como uma ferramenta de apoio ao professor para disponibilizar materiais, revisar trabalhos (tarefas), fazer o acompanhamento dos alunos na disciplina (relatórios de atividades) e também, para avaliá-los. No caso dos alunos, o ambiente facilita a entrega de trabalhos, a obtenção de material e a visualização das notas.

Os LMS são amplamente utilizados em instituições de ensino, fornecendo vários níveis de apoio para gestores, educadores e estudantes envolvidos no processo de ensino-aprendizagem. Os sistemas de gerenciamento de aprendizagem potencializam a oferta de cursos da educação on-line (nas modalidades presencial, semipresencial e a distância) melhor estruturados, alinhados à política educacional da instituição, em ambientes virtuais de aprendizagem colaborativos, com



uma quantidade mais abrangente de funcionalidades e com melhor qualidade (WEAVER *et al.*, 2008).

A adoção de um LMS em geral traz vários benefícios para a instituição de ensino, pois estes tipos de sistemas (em sua maioria): (1) fornecem vários tipos de atividades e recursos tecnológicos (ex.: fóruns colaborativos, *chats*, *web* conferências, etc.); (2) permitem a geração de relatórios educacionais; e (3) são compatíveis com diversos padrões de recursos instrucionais (tais como: SCORM, AICC, etc.) (RANKINE *et al.*, 2009; PALOMINO, 2013).

Dentre os LMS mais utilizados mundialmente, podemos destacar os comerciais: Blackboard, eCollege, ANGEL LMS e Desire2Learn. Com relação aos LMS de código aberto, destacam-se: Moodle, Amadeus, Claroline e Sakai (KUMAR *et al.*, 2011; PALOMINO, 2013).

O Moodle (acrônimo para ambiente modular de aprendizagem dinâmica orientada a objetos, em inglês: *Modular Object-Oriented Dynamic Learning Environment*) é o principal LMS de código aberto (software livre), adotado como apoio à educação on-line por inúmeras instituições de ensino em mais de 200 países, segundo o principal site da plataforma: moodle.org (2015). O Moodle foi desenvolvido na linguagem de programação PHP e utiliza como padrão o sistema gerenciador de banco de dados MySQL padrão, ambas tecnologias *open source*.

Além de ser utilizado em universidades brasileiras e no exterior, tais como UFSC, UFRJ, UFRGS, Universidade da Califórnia e Universidade de Coimbra, o Moodle também é adotado em várias instituições governamentais, podendo ser destacadas: o Conselho Nacional de Justiça (CNJ), o Tribunal Superior do Trabalho, a Academia Judicial do Tribunal de Justiça de Santa Catarina, o Ministério Público de Santa Catarina, entre outras.

O LMS Moodle foi desenvolvido com base em uma abordagem construtivista e colaborativa (DOUGIAMAS e TAYLOR, 2003), e recebe constantemente solicitações e sugestões de professores, desenvolvedores de software, administradores de sistema, *designers* instrucionais e outros membros da comunidade em geral, para o melhoramento contínuo da plataforma (PALOMINO, 2013).

Através do desenvolvimento de *plug-ins*, a ferramenta pode ser personalizada, estendida com novas funcionalidades, ou ainda, integrações com outros sistemas podem ser incorporadas. Essas modificações podem ser restritas apenas à instância Moodle de uma instituição de ensino, ou inclusive para a comunidade mundial, caso as alterações no código sejam submetidas ao processo de incorporação na versão oficial da plataforma (MOODLE, 2015).

Diante destas características mencionadas, optou-se por utilizar o LMS Moodle na implementação do modelo proposto nesta dissertação.

## 2.5 SISTEMAS DE APOIO AO ENSINO BASEADOS EM IA

Esta seção apresenta algumas pesquisas que propõem o uso da Inteligência Artificial em soluções computacionais para a educação, com o intuito de enriquecer o processo de ensino-aprendizagem.

A Inteligência Artificial (IA), ou Inteligência Computacional, é um campo da ciência que procura compreender e construir entidades inteligentes (RUSSELL e NORVIG, 2013). A ideia da racionalidade é importante na visão de Russell e Norvig (2013), pois esta é uma medida de desempenho sobre a inteligência do sistema. Um agente racional atua de forma a atingir o melhor resultado possível esperado.

Bittencourt (2006, p. 15) afirma que “o objetivo central da Inteligência Artificial (IA) é a criação de modelos para a inteligência e a construção de sistemas computacionais baseados nesses modelos”. Uma das vertentes de pesquisa em IA está relacionada ao desenvolvimento de aplicações (educacionais, comerciais ou industriais), com base em técnicas computacionais com potencial para simulação de comportamento inteligente (BITTENCOURT, 2006).

Existem diversas pesquisas publicadas que se utilizam de abordagens e técnicas propostas pelo campo da Inteligência Artificial para incluir comportamento inteligente em softwares educacionais. A incorporação de técnicas de IA em sistemas de tecnologia instrucional visa suprir deficiências encontradas em aplicações deste tipo, principalmente quanto à falta de capacidade de modelagem das características dos diferentes estudantes, e de flexibilização do comportamento do sistema (POZZEBON *et al.*, 2004).

Segundo Silveira (2001, p. 14),

a fim de prover aos sistemas computacionais de ensino, capacidade de adaptação ao contexto e de personalização do ambiente de acordo com as características do aluno, além de permitir um alto grau de interatividade entre o ambiente e os usuários e um controle de sessões de ensino em ambiente multi-usuários, as pesquisas apontam

para o uso de recursos propiciados pela Inteligência Artificial.

Palomino (2013, p. 30) defende que

ambientes virtuais de aprendizagem podem ser potencializados com técnicas de inteligência artificial, utilizando agentes inteligentes cooperativos (trabalham em background) ou pedagógicos animados (interagem com o usuário).

Silva (2007, p. 12) indica que a arquitetura de um sistema de gerenciamento de aprendizagem pode ser expandida através do emprego de técnicas de IA e da inclusão de novos componentes

a fim de tornar os LMSs mais inteligentes, capazes de interagir com o usuário (ou com um agente que o requer) para atingir os objetivos de aprendizagem desejados conforme as condições, que em alguns casos vão ao encontro um ao outro.

Mateus (2010, p. 25) aponta o uso de técnicas de IA com o objetivo de desenvolver

sistemas com capacidade de adaptação ao contexto e personalização do ambiente de acordo com as características dos alunos. Além disso, nesses sistemas gera-se um alto grau de interatividade entre o ambiente e os usuários com um controle maior das sessões de aprendizagem em ambientes multiusuários.

Na sequência deste capítulo são apresentadas algumas estratégias de computacionais baseadas em Inteligência Artificial que serviram de base para o modelo teórico proposto nesta dissertação, ou que serão incorporadas ao modelo futuramente, como evoluções do trabalho.

### **2.5.1 Sistemas Tutores Inteligentes**

Os Sistemas Tutores Inteligentes (STI) são comumente citados na literatura como uma alternativa utilizada quando se deseja incorporar IA a softwares educacionais. Os STI são ferramentas computacionais que

auxiliam os educadores no processo de ensino-aprendizagem, proporcionando experiências com maior potencial de adaptação.

Em geral, Sistemas Tutores Inteligentes (STI) são aplicações com propósito educacional de uma área específica de conhecimento (domínio), cujas arquiteturas geralmente são desenvolvidas *ad-hoc* e apresentam pouca flexibilidade para reusabilidade e pouca interoperabilidade de recursos (SANTOS e JORGE, 2013), exigindo a reconfiguração do sistema caso sejam necessárias mudanças no domínio.

Existem na literatura da Inteligência Computacional aplicada à Educação uma gama de trabalhos descrevendo sistemas tutores inteligentes dos mais diversos domínios educacionais, cada qual utilizando uma técnica de IA diferente (REIS *et al.*, 2001; DALMON *et al.*, 2011; CHEN, 2008; MATEUS, 2010; BACHARI *et al.*, 2011; SANTOS *et al.*, 2014; KURILOVAS *et al.*, 2014; CARVALHO *et al.*, 2014).

No contexto desta dissertação, optou-se por classificar os tipos de sistemas de apoio ao ensino que incorporam alguma técnica da Inteligência Artificial de acordo com o método utilizado para realizar a adaptação do conteúdo educacional.

Segundo os trabalhos identificados, em sua maioria há o emprego da abordagem de agentes inteligentes, baseados nas características oferecidas pelos sistemas multiagente (WOOLDRIDGE, 2009). Há ainda pesquisas que promovem a adaptação do conteúdo com base apenas em alguma outra técnica de IA (ex.: sistemas especialistas, redes neurais, lógica *fuzzy*, algoritmos genéticos, *swarm intelligence*, redes bayesianas, regras de produção, etc.). Por fim, destaca-se que também são adotadas estratégias híbridas, isto é, que se utilizam da abordagem de agentes juntamente com outra técnica de Inteligência computacional.

No decorrer desta dissertação serão abordados mais detalhes de alguns trabalhos sobre STI, mencionando as variadas técnicas de Inteligência Computacional empregadas, bem como comparando-os com relação às características de qualidade elencadas previamente no Capítulo 1 (ver Capítulo 3, “Trabalhos Relacionados”).

## 2.5.2 Sistemas Multiagente

O modelo de seleção dinâmica de objetos de aprendizagem apresentado nesta dissertação se baseia na tecnologia de agentes. A abordagem multiagente é uma técnica que vem sendo alvo de pesquisas da comunidade de IA há alguns anos (SILVEIRA, 2001; HÜBNER e

SICHMAN, 2003; WOOLDRIDGE, 2009; KRAVARI e BASSILIADES, 2015).

A abstração de agentes é adequada para simular comportamento de tomada de decisão, logo, muito interessante para representar perfis e características de usuários. Estes aspectos estão presentes no modelo proposto para a seleção de objetos de aprendizagem, o qual será detalhado nos capítulos posteriores.

Um Sistema Multiagente (SMA) é composto por entidades autônomas (agentes), capazes de perceber o ambiente que estão situadas, de realizar ações neste ambiente, e de se comunicar e cooperar para atingir seus objetivos. A comunicação e cooperação entre agentes individuais possibilita que estes resolvam problemas complexos que individualmente não conseguiriam fazê-lo (WOOLDRIDGE, 2009).

Wooldridge (2009) define um agente como um sistema computacional capaz de realizar ações independentes em favor de seu usuário ou “dono”. Um agente pode descobrir por sua conta o que precisa fazer para satisfazer seus objetivos de projeto, sem que precise ser explicitamente informado (ou programado) sobre o que fazer a todo o momento.

Segundo Hübner e Sichman (2003), os estudos em SMA têm como foco de atenção a forma de interação (linguagens e protocolos) entre as entidades que compõem o sistema (chamadas de agentes), o ambiente e sua organização, fatores que permitem o surgimento de comportamentos complexos e inteligentes.

Palomino (2013, p. 37) destaca que

a área de SMA estuda o comportamento de um grupo organizado de agentes autônomos que cooperam na resolução de problemas que estão além das capacidades de resolução de cada um individualmente.

Ainda segundo Palomino (2013), embora pareçam contraditórias, as propriedades autonomia e organização são fundamentais para os sistemas multiagente, fornecendo restrições aos comportamentos individuais dos agentes, com o intuito de se estabelecer um comportamento coeso do sistema (sociedade dos agentes) como um todo. É importante identificar claramente qual o(s) tipo(s) de interação(ões) que será(ão) realizada(s) entre os agentes, proporcionando-lhes a capacidade

de tomar a decisão mais adequada (melhor possível) sobre qual ação deve ser realizada por este agente.

Segundo Bittencourt (2006, p. 307),

a metáfora de inteligência utilizada em SMA é a *comunidade inteligente*, isto é, o comportamento social que é a base para a inteligência do sistema. Os membros de uma comunidade inteligente podem ser desde extremamente simples até extremamente complexos.

Brandão (2005) ressalta que o conhecimento nos sistemas multiagente está distribuído entre os agentes, os quais se organizam e cooperam entre si para alcançarem objetivos. Salazar Ospina (2014) menciona que os sistemas multiagente favorecem a aquisição e o processamento de informações distribuídas, o que se integra satisfatoriamente com a computação ubíqua e os dispositivos móveis.

Wooldridge (2009) aponta que os elementos internos (módulos) que compõem um agente são definidos através de sua arquitetura. Esta estrutura do agente estabelece a maneira como as informações são armazenadas e como se dá a interação entre os módulos internos da entidade. Em geral, os agentes possuem um conjunto pré-definido de ações que lhes permitem obter informações e/ou atuar no ambiente no qual estão inclusos. Estas ações podem ser realizadas caso determinadas premissas (pré-condições) sejam satisfeitas (WOOLDRIDGE, 2009).

Wooldridge (2009 *apud* MATEUS, 2010) destaca a necessidade de os agentes estarem preparados para tratar eventuais falhas que possam ocorrer durante suas interações com outros agentes e com o ambiente em que estão inseridos. O ambiente deve ser assumido como não-determinístico, de tal forma que um mesmo tipo de ação executada em dois momentos distintos, e em circunstâncias que se pareçam idênticas, pode produzir resultados bem diferentes, ou ainda não apresentar o resultado esperado pelo agente.

Além das características essenciais citadas por Wooldridge (2009), a saber, autonomia, reatividade, proatividade e habilidade social, Bonson (2013, p. 18) lista outras importantes vantagens proporcionadas pelos sistemas multiagente:

robustez, eficiência, capacidade para interoperações entre sistemas legados já existentes, e a habilidade para resolver problemas em que os

dados, o controle e o conhecimento estejam distribuídos.

De acordo com Silveira (2001), os sistemas multiagente podem ser classificados em duas abordagens principais, quanto ao modelo de funcionamento interno:

- SMA Cognitivos – Os agentes desta abordagem possuem uma forma explícita de representação de seus conhecimentos, e podem se comunicar e planejar suas futuras ações, através de sua capacidade de raciocínio sobre uma representação formal e interna do(s) conhecimento(s) que possuem.
- SMA Reativos – Os agentes desta abordagem têm como ênfase os seus comportamentos, sem maiores preocupações com a representação do conhecimento, e sim na representação do seu comportamento. A atividade de um agente é vista como o resultado da interação entre esta entidade e o ambiente em que está inserida.

Mateus (2010, p. 25) menciona que os agentes cognitivos e reativos são conhecidos também como agentes racionais e irracionais, respectivamente. Além disso, o autor cita que

os agentes reativos diferenciam-se dos agentes cognitivos pelo fato de não agirem deliberadamente e sim devido a eventos externos, ou seja, apenas reagem a situações, por isso o nome reativo. Por essa razão um agente reativo é mais simples que um agente cognitivo e por si só não realiza modificações significativas no ambiente onde atua.

Bittencourt (2006, p. 308) ressalta que “a diferença entre um agente reativo e um cognitivo pode ser também explicada como um compromisso entre eficiência e complexidade”. O comportamento dos agentes reativos é definido funcionalmente, o que permite a sua compilação e execução com atraso temporal pré-definido, característica ideal para aplicações que exigem respostas rápidas, onde o tempo seja uma importante restrição.

Entretanto a ausência de deliberação limita a complexidade dos comportamentos que o agente pode realizar. Desta forma, agentes mais

deliberativos tendem a exigir maior tempo de processamento e quantidade de memória, porém possibilitam comportamentos mais sofisticados (WOOLDRIDGE, 2009).

O modelo de agentes desenvolvido neste trabalho possui tanto agentes reativos, quanto agentes cognitivos (racionais). O capítulo 4 (“Modelo Proposto: ILOMAS”) apresentará detalhadamente o referido modelo de agentes.

### 2.5.2.1 Modelo BDI

Conforme mencionado na seção anterior, um dos agentes que compõem o sistema multiagente, apresentado nesta dissertação, é classificado como racional (cognitivo). Um importante modelo para agentes racionais presente na literatura da área de SMA é a teoria BDI (em inglês: *Belief, Desire, Intention*).

Agentes BDI são compostos de estados mentais: Crenças (*Beliefs*), Desejos (*Desires*) e Intenções (*Intentions*), e são também chamados de agentes cognitivos. As ações deste tipo de agente são guiadas para o alcance de seus objetivos (um subconjunto de seus desejos).

A dinamicidade da troca de informações entre o LMS e o OA durante a sessão de aprendizagem é uma característica inerente ao contexto da seleção de objetos apresentada nesta dissertação. Diante deste cenário, optou-se por utilizar o modelo BDI na concepção teórica do agente cognitivo que representa um Objeto Inteligente de Aprendizagem (OIA).

O modelo BDI foi inspirado na teoria da filosofia e ciência cognitiva sobre o pensamento humano e suas ações, proposta por Michael Bratman (1987). Na teoria de Bratman, as intenções têm um papel importante na tentativa de explicar a mente humana e as ações realizadas pelos indivíduos com base no raciocínio prático.

Os estados mentais humanos, tais como acreditar, almejar e decidir, são fatores que direcionam aquilo que fazemos. O processo do raciocínio prático humano consiste em decidir qual meta realizar (deliberação) e de que maneira esta ação será efetivamente executada (meio-fim) (BRATMAN, 1987).

Os estudos propostos para integrar a teoria de Bratman aos sistemas multiagente permitiram o desenvolvimento de agentes racionais que possuem os seguintes estados mentais:

- Crenças – As informações que o agente possui sobre o mundo (ambiente), sobre outros agentes e sobre si mesmo.



- Desejos – As situações (estados) que o agente deseja alcançar, ou seja, suas motivações.
- Intenções – O conjunto de ações que o agente se compromete a realizar para alcançar um determinado desejo, de acordo com as crenças que possui.
- Objetivos – Um subconjunto de desejos que podem ser alcançados pelo agente, não podendo ser contraditórios entre si.
- Planos – Para cumprir uma intenção o agente se utiliza de planos específicos, os quais são funções pré-definidas que podem ser aplicadas caso certas condições sejam satisfeitas.

Agentes BDI atuam de acordo com o processo de raciocínio prático, composto de deliberação de objetivos e do raciocínio meio-fim, o qual resulta na seleção de planos, que são a forma concreta de um agente atingir um determinado objetivo (WOOLDRIDGE, 2009; POKAHR *et al.*, 2014).

Wooldridge (2009) destaca um dilema relacionado à frequência com que um agente deve reavaliar suas intenções. Caso o agente não reconsidere suas intenções com uma frequência suficiente, estas podem não mais ser plausíveis (devido a alterações no ambiente).

Entretanto, se o agente revisar suas intenções muito frequentemente, estará dispendendo tempo insuficiente para alcançar a intenção (com risco de nunca ser alcançada). Ademais, o processo de deliberação é custoso e caso seja executado em uma frequência exagerada, resultará em desperdício de recursos computacionais. Logo, estas questões devem ser levadas em consideração no desenvolvimento do projeto do sistema multiagente.

Por fim, além das questões teóricas citadas nesta seção, ressalta-se que os detalhes sobre o emprego da abordagem BDI no contexto desta dissertação serão descritos em capítulos posteriores (capítulos 4 e 5).

### 2.5.2.2 Arquitetura PRS

A literatura da área de SMA apresenta algumas arquiteturas que têm como base o modelo de agentes BDI. A primeira implementação de um sistema baseado no framework conceitual do modelo BDI foi denominada IRMA (*Intelligent Resource-bounded Machine Architecture*). Posteriormente, surgiram diferentes arquiteturas, tais como

a PRS, a DMARS, entre outras. (HÜBNER *et al.*, 2004; WOOLDRIDGE, 2009).

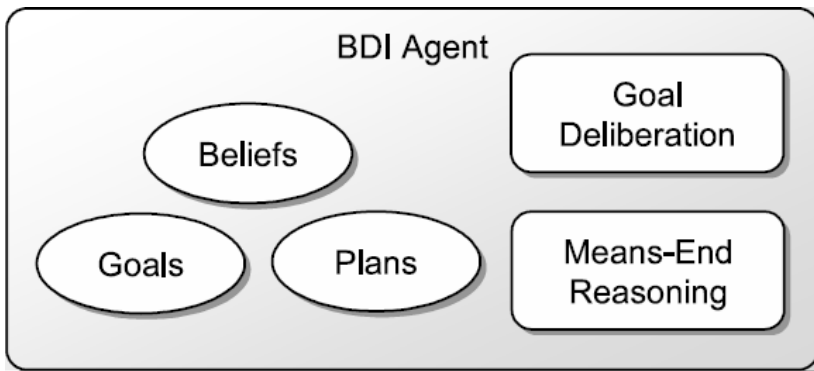
Dentre estas estruturas, destaca-se a arquitetura PRS (*Procedural Reasoning System*), considerada a mais conhecida implementação do modelo BDI. Esta arquitetura foi reimplementada diversas vezes em sistemas multiagentes ou plataformas de desenvolvimento de agentes, resultando em alterações e adaptações ao longo dos anos (WOOLDRIDGE, 2009).

Uma das implementações específicas da arquitetura PRS que pode ser apontada, devido a sua utilização no contexto deste trabalho, é a versão presente na plataforma Jadex, a qual desenvolveu uma variação à arquitetura PRS tradicional (POKAHR *et al.*, 2014).

Esta versão PRS do Jadex apresenta uma abordagem híbrida com capacidade deliberativa (proatividade) e reativa (reatividade), permitindo o desenvolvimento de agentes com raciocínio dirigido a objetivos, bem como com comportamentos de resposta a eventos (POKAHR *et al.*, 2014).

A Figura 11 ilustra elementos da estrutura e do processo de raciocínio BDI, com base na arquitetura PRS implementada na plataforma Jadex. Maiores detalhes sobre esta implementação serão mencionados no capítulo “Resultados” (Capítulo 5) desta dissertação.

Figura 11 – Estrutura e processo de um agente BDI



Fonte: Pokahr *et al.* (2014)

Cabe ressaltar o porquê da opção por detalhar as características da implementação PRS do framework Jadex apenas no capítulo de

resultados. Isto se deve ao fato de que tais características influenciam diretamente o fluxo da seleção dinâmica de objetos de aprendizagem implementada no protótipo. Desta forma, estas informações são apresentadas no referido capítulo, concomitantemente à descrição do processo de seleção de OA.

### 2.5.2.3 Desenvolvimento de Sistemas Multiagente

Existem modelos consolidados para o desenvolvimento de Sistemas Multiagente, dentre estes se destacam os padrões estabelecidos pela organização FIPA, a qual visa produzir padrões de software para agentes heterogêneos e interativos (SILVEIRA, 2001). Um SMA do padrão FIPA, entre outras características, possibilita a utilização de ontologias para troca de informações entre os agentes. No caso, a ontologia representa um contexto conceitual através do qual os agentes conseguem ter o mesmo entendimento sobre os termos e conceitos envolvidos durante a comunicação.

Há também metodologias específicas para o desenvolvimento de sistemas baseados em agentes, que fazem parte de uma área conhecida como Engenharia de Software Orientada a Agentes (em inglês, AOSE). Este tipo de metodologia permite realizar a análise e o projeto do sistema levando em considerações características próprias da abordagem SMA (HÜBNER e SICHMAN, 2003; BOISSIER *et al.*, 2013). Dentre as metodologias AOSE mais conhecidas, podem ser citadas: O-MaSE, Gaia, Tropos, INGENIAS, PASSI e Prometheus (DAM e WINIKOFF, 2013; UEZ e HÜBNER, 2013).

A metodologia Prometheus é comumente adotada na comunidade de SMA, por ser fácil de utilizar, não exigir grande experiência na área de agentes e possuir uma ferramenta que auxilie o uso do método (UEZ e HÜBNER, 2013). Devido a estas características, optou-se por utilizar a metodologia Prometheus para o desenvolvimento do modelo de agentes apresentado nesta dissertação (Capítulo 4, “Modelo Proposto: ILOMAS”).

Com relação ao desenvolvimento de sistemas multiagente, segundo Silva (2007), embora seja possível realizar a implementação do SMA em qualquer linguagem de programação existente, e da forma que o desenvolvedor preferir, existem IDEs, APIs e frameworks específicos para a abordagem de agentes.

Silva (2007, p. 31) aponta que estas ferramentas facilitam a concepção do sistema, “permitindo aos projetistas se focarem mais no problema em que o SMA deve resolver”. Desta forma, o desenvolvedor

não precisará se preocupar com detalhes internos da plataforma onde os agentes irão habitar, atuar e se comunicar, pois tais detalhes são de responsabilidade do ambiente de execução.

A comunidade de pesquisa em agentes produziu (e ainda produz) diversas ferramentas e ambientes que facilitam o processo de desenvolvimento de sistemas multiagente, possibilitando inclusive que sejam empregados padrões (de projeto, de arquitetura, de programação, de comunicação, etc.). As seguintes plataformas podem ser destacadas: FIPA-OS, ZEUS, JADE, JACK, Jadex, Jason e JaCaMo (SILVA, 2007; BOISSIER *et al.*, 2013; POKAHR *et al.*, 2014).

O protótipo implementado para instanciar o modelo teórico proposto nesta dissertação foi desenvolvido através do framework Jadex. A versão atual desta plataforma (BDI V3) permite a criação de agentes através de classes Java com o mecanismo de anotações (*annotations*) para a designação dos elementos BDI.

Conforme mencionado na seção anterior, a plataforma Jadex implementa uma versão modificada da arquitetura PRS. Maiores detalhes sobre esta implementação serão mencionados no capítulo “Resultados” (Capítulo 5) desta dissertação, durante a descrição do processo de seleção dinâmica de objetos de aprendizagem.

### **2.5.3 Representação do conhecimento**

Esta seção tem por objetivo descrever resumidamente conceitos relacionados à representação do conhecimento, que são citados em algumas discussões e referências a trabalhos futuros durante o texto desta dissertação. Não se almeja realizar uma explanação detalhada sobre esta subárea da IA, mas apenas apresentar uma visão geral sobre alguns conceitos relacionados que são abordados no presente trabalho.

A literatura da IA aponta que em sistemas capazes de raciocinar de maneira inteligente, há a necessidade se representar internamente o conhecimento sobre o mundo, o que pode ser feito através da construção de bases de conhecimento (BITTENCOURT, 2006; RUSSEL e NORVIG, 2013).

Uma das formas de se representar conhecimento em sistemas com comportamento inteligente é a Lógica Descritiva (em inglês, DL). Segundo Brandão (2005), lógicas descritivas (ou lógicas de descrição) são formalismos utilizados para representação do conhecimento de um

domínio de aplicação (o mundo, ou ambiente), de acordo com a definição de conceitos relevantes do domínio em questão (a terminologia).

A DL permite a especificação de propriedades dos objetos e indivíduos que compõem este domínio, com base nos conceitos que foram definidos. Brandão (2005) ressalta que a semântica das lógicas descritivas é baseada em lógica de primeira ordem, e também destaca a capacidade de dedução de uma DL, o que permite inferir conhecimento implícito a partir de conhecimento explicitamente representado, além de classificar hierarquicamente os conceitos que compõem a terminologia.

Outras formas de representação de conhecimento são: sistemas de produção, redes semânticas, quadros (*frames*), redes neurais, lógica proposicional, lógica temporal, lógica *fuzzy*, ontologias, etc.

A literatura de sistemas multiagente apresenta diversas estratégias de representação do conhecimento dos agentes, seja por uma base de conhecimento (KB) única e compartilhada pelas entidades, ou por uma base de crenças (BB) individual para cada agente. Além disso, a forma de estruturação do conhecimento (seja em uma KB ou BB) é implementada através de diferentes abordagens de representação de conhecimento previamente mencionadas, por exemplo: regras de produção, lógica de primeira ordem, lógica descritiva e ontologias (BINAS e IOERGER, 2004; KLAPISCAK e BORDINI, 2009; HOLMES e STOCKING, 2009; CARRERA *et al.*, 2014; KADHIM *et al.*, 2014).

A escolha por uma forma de representação de conhecimento deve levar em consideração as características específicas do sistema, bem como deve ser avaliada a relação entre o custo e o benefício de se utilizar uma estratégia mais ou menos expressiva, do ponto de vista de representação, e mais ou menos custosa, do ponto de vista de recursos computacionais (memória e tempo de resposta).

### 2.5.3.1 Ontologias

No modelo teórico proposto nesta dissertação, conforme apresentado no capítulo 4 (“Modelo Proposto: ILOMAS”), há a possibilidade de se desenvolver uma integração com ontologias de domínio educacional e de padrões de OA, com o intuito de expandir semanticamente a busca por objetos de aprendizagem.

Uma ontologia é um modelo de categorização e definição de conceitos dentro de um domínio, permitindo representar atributos destes conceitos e relacionamentos entre conceitos. Russell e Norvig (2013, p. 269) definem que “a palavra ontologia representa uma teoria específica sobre a natureza de ser ou existir”. Grubber (1993 *apud* RIOS, 2005, p.

2) cita que uma ontologia é “a especificação formal explícita de uma conceitualização compartilhada”.

As ontologias são importantes elementos para a composição de uma base de conhecimento, pois provêm um conjunto de conceitos ou termos com o objetivo de descrever semanticamente algum domínio. Logo, uma ontologia é o alicerce sobre o qual o conhecimento pode ser construído (RIOS, 2005). Além disso, como podem ser expressas através de lógica descritiva (DL), é possível realizar raciocínio sobre os elementos existentes em uma base de conhecimento ontológica.

De acordo com Santos *et al.* (2006), as ontologias fornecem meios de se explicar conceitos, e relacionamentos entre estes conceitos, permitindo que agentes inteligentes interpretem o significado destas conceitualizações de maneira flexível e sem ambiguidades.

Segundo Bonson (2013), uma ontologia geralmente utiliza os seguintes conceitos:

- Indivíduos – instâncias de classes, objetos.
- Classes – conceitos, conjuntos, tipos de objetos.
- Propriedades – atributos que indivíduos ou classes podem possuir.
- Relações – relacionamentos entre classes e indivíduos.
- Restrições – devem ser verdadeiras para que um axioma (sentença) seja adicionado.
- Regras – sentenças na forma *se* → *então* que descrevem inferências lógicas.

A concepção de uma ontologia pode ser realizada através de uma metodologia específica para este fim, tais como 101 e Methontology, fornecendo diversos instrumentos que possibilitam a representação do conhecimento de maneira iterativa (não-linear). As metodologias permitem ao engenheiro do conhecimento reutilizar e ou estender a ontologia, e geralmente estão integradas a ferramentas de edição de ontologias, como o Protégé (PRIMO, 2013; SALAZAR OSPINA, 2014).

Primo (2013) destaca as etapas da metodologia 101, que podem ser revisitadas pelo engenheiro de ontologias sempre que necessário: (1) determinar o domínio e o escopo da ontologia; (2) investigar o reuso de ontologias existentes; (3) listar termos importantes; (4) definir classes; (5)

identificar a hierarquia de classes; (6) definir restrições e axiomas das classes.

Dentre as tecnologias mencionadas, destaca-se que no contexto desta dissertação, foram utilizadas a metodologia 101 e a ferramenta Protégé, para a concepção da ontologia apresentada no capítulo 4 (“Modelo Proposto: ILOMAS”). Esta ontologia serviu como uma terminologia, a qual foi usada na definição das crenças do agente que representa um Objeto Inteligente de Aprendizagem (OIA).

### 2.5.3.2 *Web Semântica*

Embora no contexto desta dissertação não tenha sido empregada integração direta entre o modelo proposto e tecnologias da *Web Semântica*, esta possibilidade é descrita como trabalhos futuros. Desta forma, julga-se oportuno descrever alguns conceitos associados à *web semântica*, ainda que de maneira sucinta.

Conforme Santos *et al.* (2009), a *Web Semântica* (também conhecida como *web* de dados) pode ser vista como uma evolução da *web* atual, de forma que as informações recebam um significado computacional. O intuito é permitir que agentes de software sejam capazes de interpretar o conteúdo das páginas.

A *web semântica* difere da *web* tradicional onde as informações são disponibilizadas em linguagem natural, adequado apenas para a compreensão por seres humanos. Com as características proporcionadas pela *web semântica*, aplicações autônomas podem consumir e gerar conhecimento em favor do usuário (SANTOS *et al.*, 2009; PRIMO, 2013; CAMPOS, 2014).

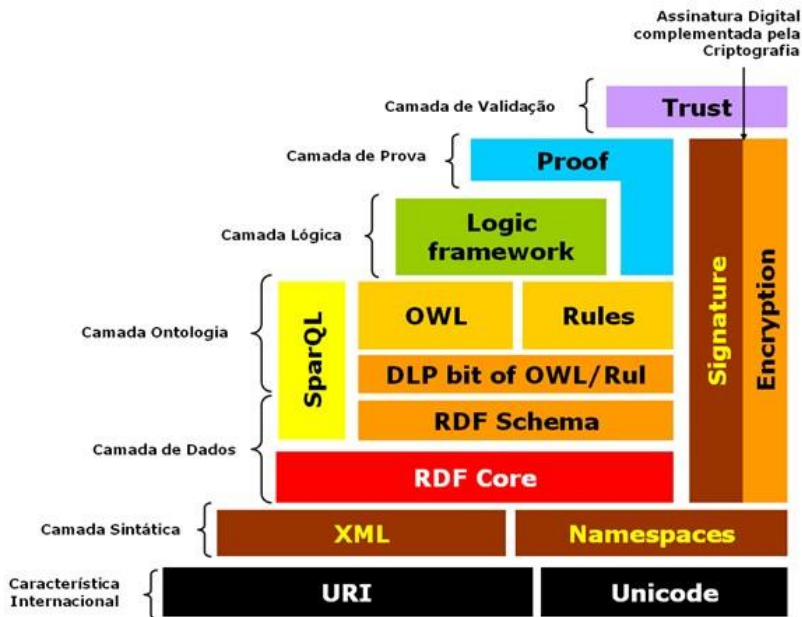
O W3C (*World Wide Web Consortium*) desenvolve um conjunto de recomendações para o desenvolvimento da *Web Semântica*. Entre estas tecnologias estão:

- XML – definição da sintaxe para os documentos *web*.
- RDF – modelo de representação de recursos (nós) e declarações sobre os recursos (relações), servindo de base para modelos de dados semânticos.
- OWL (*Web Ontology Language*) – uma linguagem de marcação semântica (baseada em RDF e DL) que permite a criação de ontologias, através da descrição de classes (conceitos), propriedades e instâncias, além de relacionamentos entre classes, cardinalidade, etc., por meio de ferramentas como o Protégé

(SANTOS *et al.*, 2006). A OWL possui três sub-linguagens: (1) *OWL-Lite*, (2) *OWL-DL* e (3) *OWL-Full*, cada uma apresentando um grau de expressividade diferente (BRANDÃO, 2005).

A Figura 12 exibe as camadas que compõem a pilha de tecnologias da *Web Semântica* definidas pelo W3C.

Figura 12 – Pilha da linguagem da *Web Semântica*



Fonte: Santos e Alves (2009)

Existem diversas pesquisas referentes à aquisição de conhecimento por parte de agentes inteligentes, destacando-se àquelas que relatam o uso das crenças dos agentes como insumo para a descoberta de conhecimento relacionado (às crenças) na *web* de dados, através de consultas SPARQL e ontologias (SANTOS *et al.*, 2009).

Nessah e Kazar (2013) propuseram uma abordagem para filtragem de recursos na *web*, através de um sistema multiagente integrado a uma base de conhecimento ontológica, permitindo a busca de documentos conforme uma medida de similaridade semântica. Saleena e Srivatsa (2014) apresentam um sistema de ensino on-line adaptativo, o qual



recomenda conteúdos instrucionais de acordo com o nível de conhecimento do usuário, baseado na similaridade de conceitos entre ontologias.

Suresh *et al.* (2014) desenvolveram um modelo para criação de ontologias de domínio para *e-learning*, com base em técnicas de data mining ontológico, extraindo os conceitos automaticamente a partir de dados textuais. Há também o trabalho de Casali *et al.* (2013) que propõem uma abordagem baseada em ontologia para o preenchimento de metadados de recursos instrucionais.

Ressalta-se ainda a existência de trabalhos, que propõem a integração de semântica às tecnologias educacionais, como por exemplo: a descrição dos elementos do padrão de metadados de objetos de aprendizagem LOM em RDF (NILSSON *et al.*, 2003) e outras linguagens de marcação baseada em ontologia (HSU, 2013) para a busca inteligente de objetos de aprendizagem em repositórios (CAMPOS, 2013; PRIMO, 2013).

Ao longo desta dissertação, serão mencionadas possibilidades de utilização de ontologias e outras tecnologias da *Web Semântica* no modelo proposto, principalmente como trabalhos futuros.

## 2.6 SISTEMA TUTOR INTELIGENTE BASEADO EM AGENTES COMPATÍVEL COM O PADRÃO SCORM

A última seção deste capítulo de fundamentação teórica tem por objetivo descrever brevemente os trabalhos anteriores do grupo de pesquisa IATE – UFSC associados ao conceito dos Objetos Inteligentes de Aprendizagem (SILVEIRA *et al.*, 2006; SILVA *et al.*, 2008; BAVARESCO e SILVEIRA, 2009), além do que já havia sido mencionado no capítulo introdutório (Capítulo 1, “Introdução”).

Os referidos trabalhos propuseram a seleção de objetos de aprendizagem através de um sistema multiagente com base em recursos educacionais segundo o modelo SCORM (SILVA *et al.*, 2008; BAVARESCO e SILVEIRA, 2009). Os trabalhos supracitados serviram de base para o modelo proposto nesta dissertação.

A proposta é estender os trabalhos mencionados, para propor a seleção dinâmica de objetos de aprendizagem em sistemas de gerenciamento de aprendizagem, segundo características do aluno e dados dos objetos relacionados (AMORIM JR. e SILVEIRA, 2015).

Esta evolução ao modelo é suportada por uma arquitetura de agentes cognitivos, os quais se utilizam de (1) dados do aluno no ambiente virtual (LMS), (2) de metadados dos objetos de aprendizagem e (3) de

dados da interação do aluno com o recurso instrucional, segundo o padrão de integração SCORM. O modelo proposto será detalhado nos próximos capítulos desta dissertação.

### 3 TRABALHOS RELACIONADOS

Como ponto de partida desta pesquisa, uma revisão sistemática da literatura foi realizada no portal de periódicos da CAPES/MEC, observando trabalhos existentes na área de sistemas de adaptação de conteúdos educacionais. O portal da CAPES disponibiliza acervos de diversas bases de periódicos e artigos, como por exemplo: IEEEExplore, ACM Digital Library e Springer Link.

O período definido na busca através do sistema da CAPES foi entre 2005 e 2015. Os termos utilizados na pesquisa (tanto em inglês, quanto em português) estavam relacionados a adaptação de conteúdo educacional, seleção dinâmica de conteúdo *e-learning*, seleção dinâmica de objetos de aprendizagem, ambientes educacionais inteligentes, entre outros termos correlatos (em inglês).

Inicialmente, mais de quatrocentos artigos foram encontrados. Na sequência, realizou-se uma filtragem dos trabalhos pelo título, pela leitura do resumo (abstract), e pela a leitura dos artigos que pareciam estar fortemente relacionados ao tema desta pesquisa. Após esta etapa de análise dos artigos restantes, chegou-se a quantidade de doze trabalhos que apresentam alguma correlação com esta pesquisa.

Posteriormente, realizou-se busca com critérios semelhantes no sistema de pesquisa dos Anais do Simpósio Brasileiro de Informática na Educação e da Revista Brasileira de Informática na Educação. Ambas as publicações são bem classificadas no *Qualis* da CAPES, sendo o principal evento e periódico, respectivamente, na área da informática aplicada a educação no Brasil. Como resultado, foram identificados outros seis estudos similares, cujo objetivo é propiciar adaptatividade e reuso de materiais instrucionais para sistemas educacionais.

Finalmente, quatro trabalhos foram acrescentados à revisão de literatura, por ter se tomado ciência da existência destas pesquisas relacionadas ao tema desta dissertação, através da sinalização do orientador do trabalho e de um membro da banca, durante a realização do exame de qualificação. Os trabalhos adicionados foram uma tese de doutorado, duas dissertações de mestrado e um artigo do periódico *Interdisciplinary Journal of E-learning and Learning Objects* (IJELLO).

Algumas das pesquisas selecionadas apresentam sistemas construídos como uma extensão do LMS, usando diferentes estratégias de adaptação do ambiente de aprendizagem. Há ainda trabalhos não integrados ao LMS, e que utilizam formas distintas de adaptação da experiência instrucional, segundo o modelo do aluno. Existem pesquisas que propõem sistemas de recomendação de conteúdo educacional de

forma adaptável. Por fim, alguns trabalhos analisados combinam o paradigma de Sistemas Multiagente (SMA) ao LMS para permitir que o ambiente tenha um nível maior de adaptação.

Na sequência, serão listados cada um dos trabalhos identificados no levantamento bibliográfico realizado, citando também a estratégia que utilizam para dotar de inteligência o sistema educacional.

### 3.1 SISTEMAS DE APOIO AO ENSINO ADAPTATIVOS

Conforme mencionado no Capítulo 2 (“Fundamentação Teórica”), a literatura apresenta diversas pesquisas que se utilizam de abordagens e técnicas propostas pelo campo da Inteligência Artificial para incluir comportamento inteligente em softwares educacionais. Entre estas abordagens, destacam-se àquelas baseadas em Sistemas Multiagente, devido ao emprego desta estratégia no modelo proposto nesta dissertação.

Alencar e Netto (2011) desenvolveram um Sistema Multiagente que utiliza dados obtidos da atividade fórum no Moodle para exibir recursos e outras atividades aos estudantes. Frade *et al.* (2014) apresentam um ambiente 3D para recomendação de OA baseado no perfil do aluno, usando informações da base de dados do Moodle.

Mateus (2010) propõe um framework baseado em SMA para promover um acompanhamento da aprendizagem dos alunos por meio da análise do desempenho individual durante um curso em um ambiente virtual de aprendizagem. A estratégia de adaptação é realizada por agentes BDI, cujas crenças são modeladas com base na lógica *fuzzy*.

Mikic-Fonte *et al.* (2010) apresentam um módulo de tutoria inteligente para uma plataforma de ensino on-line, baseado em agentes BDI. Os agentes monitoram os estudantes e sugerem objetivos e caminhos de aprendizagem, de acordo com o respectivo perfil do aluno.

Leung *et al.* (2013) projetaram um sistema multiagente BDI para uma plataforma de aprendizagem baseada em jogos. Os agentes guiam o fluxo do jogo 3D, através da interação com o aluno e de acordo com uma ontologia, objetivando aumentar o engajamento dos estudantes.

Giuffra e Silveira (2013) desenvolveram um ambiente adaptativo, baseado em agentes BDI capazes de identificar o perfil cognitivo e no desempenho do estudante, e que é integrada ao LMS Moodle. Finalmente, Bremgartner *et al.* (2014) apresentam um sistema multiagente BDI apto a selecionar dinamicamente recursos construtivistas (FILATRO, 2010) no ambiente Moodle.

Komlenov *et al.* (2010) e Despotovic *et al.* (2012) produziram *plug-ins* (extensões) no LMS Moodle (2015), os quais exibem um

determinado material previamente criado no ambiente, dependendo do estilo cognitivo do aluno, através do sequenciamento condicional do conteúdo e de técnicas de data mining, respectivamente.

Peter e Bacon (2010) desenvolveram um STI *standalone* que seleciona qual recurso será exibido ao aluno dependendo do seu estilo de aprendizagem identificado. Os recursos específicos para cada estilo de aprendizagem devem constar na base de conhecimento que é representada através de uma ontologia.

Bachari *et al.* (2011) desenvolveram um software de conteúdo *e-learning* onde o professor monta a sequência de objetos de aprendizagem apresentados ao aluno, dependendo do tipo em que este estudante é classificado, utilizando redes bayesianas.

Chen (2008) aborda a utilização de um algoritmo genético para definir a trilha de aprendizagem mais adequada para cada perfil de aluno, enquanto Santos *et al.* (2014) utilizam a lógica difusa (*fuzzy*) para definir se formas de ajuda serão exibidas ao aluno e para sinalizar ao professor sobre a situação do estudante durante a experiência de aprendizagem.

Moura e Fernandes (2012) e Kurilovas *et al.* (2014) utilizam o modelo de inteligência de enxames (*swarm intelligence*) para selecionar os cenários de aprendizagem, isto é, os agrupamentos de objetos de aprendizagem (adicionados previamente na ferramenta) que serão apresentados ao estudante, dependendo das suas preferências.

Por fim, alguns trabalhos apresentam sistemas de recomendação de objetos de aprendizagem de acordo com as características do aluno. Seguindo esta estratégia, Primo (2013) propõe um modelo de representação de conhecimento baseado em ontologia para apoiar sistemas de recomendação educacionais. A solução utiliza informações do perfil do usuário e do objeto de aprendizagem para recomendar o conteúdo instrucional.

Ritrovato (2014) e Vesin *et al.* (2013) apresentam as ferramentas IWT (*Intelligent Web Teacher*) e Protus, respectivamente, as quais são baseadas em ontologias. O material instrucional de cada modelo (estilo de aprendizagem) de aluno precisa ser previamente criado em uma ferramenta de autoria para então ser disponibilizado (configurado) na solução. No trabalho de Vesin *et al.* (2013), a adaptação do conteúdo se dá através da classificação do aluno em grupos de perfis e da mineração de regras de associação.

Sabitha *et al.* (2014) propõem a integração de um sistema de recomendação de objetos de aprendizagem a um LMS, com base na técnica de *clustering* (agrupamentos automáticos segundo semelhanças) da mineração de dados (*data mining*), conforme o perfil de aprendizagem

do estudante. Pereira *et al.* (2014) utilizam dados de redes sociais para definir o perfil do estudante e então recomendar objetos de aprendizagem a estes alunos.

Carvalho *et al.* (2014) apoiam-se em um sistema de regras de produção para definir quais objetos recomendar de acordo com o tipo cognitivo do estudante. Salazar Ospina (2014) propõe um modelo de sistema multiagente ubíquo, adaptativo e sensível ao contexto para realizar a oferta personalizada de recursos educacionais, com base em ontologias.

### 3.2 COMPARAÇÃO ENTRE OS TRABALHOS IDENTIFICADOS

A proposta de pesquisa que está sendo apresentada tem o intuito de desenvolver um modelo que permita a produção de cursos e experiências de ensino-aprendizagem adaptáveis e com favorecimento ao reuso de materiais instrucionais.

A adaptatividade da solução baseia-se na possibilidade de que novos Objetos de Aprendizagem (OAs) sejam incluídos no LMS, sem a necessidade prévia de configuração do curso, assim que for identificada a necessidade de reforçar um conceito ao aluno. Essa necessidade de reforço é identificada automaticamente através da verificação do desempenho do aluno (nota) nas avaliações de cada unidade instrucional.

A reutilização é alcançada pelo reaproveitamento de Objetos de Aprendizagem existentes relacionados ao assunto que o aluno apresentar dificuldades, evitando a necessidade do DI ou professor confeccionar novo material similar ao objeto de aprendizagem pré-existente e validado. A estruturação dos cursos também se torna mais flexível, pois não é necessário configurar todos os conjuntos possíveis de sequenciamento de materiais para cada perfil de aluno.

Levando-se em conta estas características, há questões que devem ser observadas com o intuito de identificar pontos onde os trabalhos relacionados podem ser aprimorados. Desta forma, foi realizado um estudo comparativo entre as pesquisas correlatas, resultando na classificação dos trabalhos segundo critérios específicos.

Com base neste estudo foi possível realizar a comparação entre as pesquisas correlatas e o modelo proposto nesta dissertação, com relação: (1) à forma como é definido o Modelo do Aluno; (2) ao conceito de Objeto de Aprendizagem, e à utilização de padrões; (3) ao acesso aos dados

decorrentes da interação do aluno com o ambiente; (4) à integração com LMS; (5) e ao tipo de adaptatividade obtido.

O Quadro 3 exibe a comparação entre os trabalhos relacionados, conforme os critérios elencados. Ao final deste quadro, inclui-se a proposta desta dissertação para efeitos comparativos.

Quadro 3 – Comparação entre os trabalhos relacionados e esta proposta

<b>Trabalho</b>	<b>Modelo do Aluno</b>	<b>Padrão OA</b>	<b>Monitora Interação</b>	<b>Acessado no LMS</b>	<b>Âmbito de Adaptação</b>
<b>Peter e Bacon, 2010</b>	Questionário	Não	Não	Não	Ambiente
<b>Ritrovato, 2014</b>	Questionário	Não	Não	Não	Ambiente
<b>Moura e Fernandes, 2012</b>	Questionário	Não	Não	Não	Ambiente
<b>Vesin <i>et al.</i>, 2013</b>	Questionário	SCORM	Não	Não	Ambiente
<b>Chen, 2008</b>	Questionário	SCORM	Não	Não	Ambiente
<b>Kurilovas <i>et al.</i>, 2014</b>	Questionário	LOM	Não	Não	Ambiente
<b>Carvalho <i>et al.</i>, 2014</b>	Questionário	LOM	Não	Não	Ambiente
<b>Komlenov <i>et al.</i>, 2010</b>	Questionário	LOM	Não	Sim	Ambiente
<b>Bachari <i>et al.</i>, 2011</b>	Questionário	LOM	Não	Sim	Ambiente
<b>Despotovic <i>et al.</i>, 2012</b>	Questionário	Não	Não	Sim	Ambiente
<b>Alencar e Netto, 2011</b>	Fórum	Não	Não	Sim	Ambiente
<b>Giuffra e Silveira, 2013</b>	Avaliações	Não	Não	Sim	Ambiente
<b>Bremgartner <i>et al.</i>, 2014</b>	Questionário, Avaliações e Frequência	Não	Não	Sim	Ambiente
<b>Pereira <i>et al.</i>, 2014</b>	Facebook	LOM	Não	Não	Ambiente
<b>Frade <i>et al.</i>, 2014</b>	Contexto	SCORM	Não	Não	Ambiente
<b>Mikic-Fonte <i>et al.</i>, 2010</b>	Objetivos de Aprendizagem e Progresso	LOM	Não	Sim	Ambiente
<b>Mateus, 2010</b>	Atividades de Aprendizagem e Avaliações	Não	Não	Não	Ambiente

<b>Trabalho</b>	<b>Modelo do Aluno</b>	<b>Padrão OA</b>	<b>Monitora Interação</b>	<b>Acessado no LMS</b>	<b>Âmbito de Adaptação</b>
<b>Primo, 2013</b>	Questionários, Avaliações e Histórico de Acesso	OBAA e LOM	Não	Não	Ambiente
<b>Salazar Ospina, 2014</b>	Questionário, Avaliações e Contexto	LOM	Não	Não	Ambiente
<b>Sabitha et al., 2014</b>	Avaliações e Progresso	Extensão LOM	Não	Sim	Ambiente
<b>Leung et al., 2013</b>	Avaliações	Não	Sim	Não	Ambiente
<b>Santos et al., 2014</b>	Interação	Extensão SCORM	Sim	Não	Ambiente
<b>Amorim Jr., 2015</b>	<b>Avaliações e Interação</b>	<b>LOM e SCORM</b>	<b>Sim</b>	<b>Sim</b>	<b>Objeto de Aprendizagem</b>

Fonte: Autor (2015)

Nota-se que algumas pesquisas definem o Modelo do aluno segundo questionários no início do curso, o que pode ser considerado intrusivo e distrair os estudantes (VESIN *et al.*, 2013). Verifica-se também que alguns estudos não consideram a utilização de padrões de empacotamento e/ou metadados na definição de objetos de aprendizagem, dificultando o reuso de recursos e a interoperabilidade com outros ambientes virtuais.

Ressalta-se que alguns trabalhos não provêm integração com o LMS, resultando na perda das funções ofertadas por este tipo de sistema e exigindo que o aluno mude o seu foco de atenção (acesso ao objeto de aprendizagem fora do LMS), com possível distração (ARDITO *et al.*, 2006; WINTER *et al.*, 2010).

Destaca-se ainda que a maioria dos trabalhos analisados não realiza o monitoramento em tempo real da interação entre o aluno e o Objeto de Aprendizagem (OA). Obter dados da interação instantaneamente permite



que o sistema se adapte ao aluno imediatamente, e não apenas após o fim da sessão de aprendizagem.

Na sequência, serão abordados detalhadamente os critérios de comparação, bem como as questões levantadas.

### **3.2.1 Análise das Questões Elencadas**

Conforme mencionado anteriormente, há questões que devem ser observadas com o intuito de identificar pontos onde os trabalhos relacionados podem ser aprimorados.

A primeira questão se refere à forma como é determinado o Modelo do Aluno. Em sua maioria, os estudos identificados obtêm o perfil do estudante através de questionários específicos aplicados no início do curso, os quais os estudantes devem responder. Este passo extra pode ser considerado intrusivo e distrair os alunos (VESIN *et al.*, 2013). As exceções são os trabalhos que utilizam como critérios de adaptação do ambiente informações obtidas de forma automática com base em dados do ambiente ou da experiência de aprendizagem (MIKIC-FONTE *et al.*, 2010; ALENCAR e NETTO, 2011; PEREIRA *et al.*, 2014; FRADE *et al.*, 2014; BREMGARTNER *et al.*, 2014; GIUFFRA e SILVEIRA, 2013; SANTOS *et al.*, 2014; LEUNG *et al.*, 2013).

A segunda questão está relacionada ao favorecimento do reuso de objetos de aprendizagem através do emprego de padrões na construção de conteúdos instrucionais. Padrões de OA permitem o uso dos recursos em diferentes ambientes que sejam compatíveis com estes modelos (NASH, 2005; SABITHA *et al.*, 2014). O objetivo é padronizar metadados e a forma de comunicação entre os objetos de aprendizagem e os LMS, possibilitando a reutilização dos recursos criados em vários cursos e em diferentes sistemas de gerenciamento de aprendizagem.

Um padrão mundialmente difundido e utilizado em cursos on-line de instituições de ensino públicas, privadas e escolas de governo é o SCORM, o qual define uma estrutura específica de empacotamento de recursos instrucionais, e os protocolos (API) de comunicação com os principais LMS (WANG e HSU, 2006). Ferramentas de autoria tais como eXeLearning (2015) e Articulate (2015) são capazes de empacotar conteúdo no referido padrão. Desta forma, é recomendável que o sistema de aprendizagem possibilite o aproveitamento de OA no padrão SCORM.

A reutilização de objetos de aprendizagem é facilitada também pela padronização dos metadados destes recursos. Metadados são elementos que descrevem informações sobre os objetos, tornando

possível sua busca e localização em repositórios e ambientes virtuais (CASALI *et al.*, 2013).

O padrão LTSC/IEEE *Learning Object Metadata* (LOM) é um exemplo de estrutura de metadados para objetos de aprendizagem, permitindo que sejam obtidas informações sobre como usar o OA, e em que condições (CARVALHO *et al.*, 2014). Além disso, a descrição dos metadados de maneira padronizada beneficia também a integração com ontologias para representação semântica dos objetos de aprendizagem em um ambiente virtual ou repositório (PEREIRA *et al.*, 2014).

O terceiro ponto de melhoria refere-se à forma de acesso aos dados da interação do aluno com o ambiente. A capacidade de se monitorar em tempo real (on-line) a interação do estudante com o objeto de aprendizagem, sem precisar aguardar a finalização desta interação para obter a sua situação e os seus resultados, traria benefícios, tais como, a possibilidade de rápido *feedback* ao estudante (DALMON *et al.*, 2011).

A obtenção de dados instantâneos referente à interação do aluno com o OA possibilita que o sistema se adapte ao estudante imediatamente, e não somente após a finalização de uma sessão de aprendizagem, o que potencializa a aprendizagem por parte do aluno, e favorece a sua motivação quanto à atividade educacional (VAN SETERS *et al.*, 2012).

Entre os trabalhos identificados, as propostas de Santos *et al.* (2014) e Leung *et al.* (2013), são as únicas que permitem o monitoramento de alguns itens configurados pelo professor durante o acesso do estudante ao objeto de aprendizagem. Estes itens podem ser, respectivamente: o tempo sem interagir e a quantidade de ajudas solicitadas (SANTOS *et al.*, 2014), ou o desempenho em um jogo e os estados emocionais do aluno (LEUNG *et al.*, 2013).

O quarto possível ponto de melhoria refere-se à forma de disponibilização do acesso ao objeto de aprendizagem para o estudante nos trabalhos identificados. Segundo Tarouco *et al.* (2003 *apud* DA SILVA *et al.*, 2010), objetos de aprendizagem são aproveitados de maneira mais eficiente quando possibilitam a integração com sistemas de gerenciamento de aprendizagem (LMS).

Dentre as pesquisas relacionadas que foram analisadas, percebe-se que algumas desenvolveram um ambiente próprio (independente) para exibir os OA, com o intuito específico de implementar os modelos propostos, não sendo incorporados a um LMS largamente difundido nas instituições educacionais ao redor do mundo, tais como: Moodle, Blackboard (2015), Amadeus (2015) e Claroline (2015).

A não integração com um LMS faz com que seja perdido todo o conjunto de funcionalidades disponibilizadas por este tipo de sistema.

Além disso, o ato de saída de um ambiente e acesso ao outro faz com que o estudante precise mudar o seu foco de atenção, interrompendo o fluxo de aprendizagem e resultando em possíveis distrações (ARDITO *et al.* 2006; WINTER *et al.*, 2010).

Por fim, o quinto e último ponto de aperfeiçoamento identificado, relacionado ao tipo de adaptação obtido nos sistemas analisados, é a possibilidade de acoplar novos recursos de aprendizagem (OA) dinamicamente à estrutura do curso no ambiente virtual.

Nos trabalhos observados, a forma de configuração do conteúdo no LMS exige que o professor (ou designer instrucional) precise configurar previamente todos os possíveis caminhos do curso para cada estilo de estudante, o que pode ser difícil e levar muito tempo para realizar (BROWN *et al.*, 2005; BACHARI *et al.*, 2011).

Além disso, a adição de um novo objeto de aprendizagem envolve a modificação da estrutura do curso no LMS, o que remete a um limitado grau de adaptação. A ausência da possibilidade de dinamicamente adicionar novos recursos instrucionais (OA) ao sistema ou ambiente faz com que a inteligência dos modelos analisados seja dependente da implementação de cada sistema (especificação explícita dos comportamentos dinâmicos), ou da configuração realizada pelo professor ou *designer* instrucional (DI), corroborando com a constatação feita por Brown (2005).

Como resultado da ausência desta característica nos trabalhos verificados, o professor (ou DI) precisa configurar explicitamente o novo objeto de aprendizagem no ambiente/sistema, ou então é necessário que os analistas de sistemas modifiquem o software educacional para acrescentar o novo OA, tornando estes modelos menos flexíveis, isto é, com adaptatividade e reuso restritos (SABITHA *et al.*, 2014).

Diante desta situação, seria interessante permitir a inclusão de novos objetos de aprendizagem ao ambiente sem que o professor (ou designer instrucional) precisasse reconfigurar o curso.

De certo modo, os trabalhos que utilizam a estratégia de sistemas de recomendação de recursos armazenados em repositórios de objetos de aprendizagem proporcionam esta característica. Sempre que um novo objeto é adicionado ao repositório, este passa a ser elegível para ser sugerido ao aluno, dependendo do assunto a que ele esteja relacionado (VESIN *et al.*, 2013; PEREIRA *et al.*, 2014).

Entretanto, os sistemas de recomendação apenas identificam a lista de objetos relacionados a determinado assunto existentes em um ou mais

repositórios, disponibilizando esta relação aos usuários (DA SILVA *et al.*, 2010; PRIMO, 2013; SABITHA *et al.*, 2014).

Cabe destacar que estes sistemas não possuem a capacidade de monitorar a interação do aluno com o objeto de aprendizagem sugerido. Desta forma, a lista de objetos recomendados pode ser modificada dependendo do perfil do aluno, porém não há mudança da experiência educacional durante a interação do aluno com o OA.

Fazer com que o comportamento de um recurso instrucional pareça se modificar durante a interação do aluno, resultaria em uma adaptação em nível de objeto, ou seja, a granularidade da adaptação seria o objeto de aprendizagem. Neste caso, a adaptação seria do conteúdo do objeto de aprendizagem, não sendo necessário realizar modificações no ambiente virtual (LMS).

### **3.2.2 Características da Proposta**

A pesquisa apresentada nesta dissertação contribui com avanços no estado da arte do campo da inteligência computacional aplicada à educação e visa suprir as questões detectadas, através do desenvolvimento de um modelo para seleção de objetos de aprendizagem de forma dinâmica, que:

- Favorece o reuso de materiais instrucionais compatíveis com padrões de recursos educacionais, em um LMS, sem a necessidade de pré-configuração de todos os recursos no ambiente virtual.
- Monitora a interação do estudante com o objeto de aprendizagem, obtendo de maneira automática as informações sobre este aluno durante a experiência de ensino.
- Permite a adaptação do conteúdo apresentado pelo objeto de aprendizagem conforme o nível de conhecimento (desempenho) apresentado pelo estudante, e de acordo com os dados dos recursos instrucionais (metadados).

A proposta se baseia em estudos anteriores do grupo de pesquisa IATE – UFSC, sendo uma extensão do modelo de Objetos Inteligentes de Aprendizagem (SILVEIRA *et al.*, 2006; SILVA *et al.*, 2008; BAVARESCO e SILVEIRA, 2009), que correspondem a objetos de aprendizagem com comportamento inteligente. Maiores detalhes da proposta são apresentados no Capítulo 4 (“Modelo Proposto: ILOMAS”).

## 4 MODELO PROPOSTO: ILOMAS

Esta dissertação apresenta detalhes da pesquisa que foi realizada durante o mestrado, a qual pretende suprir a limitação identificada nos trabalhos correlatos (conforme descrito no Capítulo 3) através da abordagem de Sistemas Multiagente integrada a objetos de aprendizagem (OA), acessados via ambiente virtuais (LMS).

O modelo teórico proposto possibilita a adaptação do conteúdo educacional de acordo com o desempenho do aluno, e a inclusão de recursos instrucionais na experiência de aprendizagem mesmo que estes não façam parte previamente do LMS, adicionando comportamento inteligente ao ambiente.

Com o intuito de expandir a capacidade da abordagem de objetos de aprendizagem, trabalhos anteriores realizados pelo grupo de pesquisa IATE propuseram a junção entre esta tecnologia e os Sistemas Multiagente (SMA). Esta estratégia foi denominada Objetos Inteligentes de Aprendizagem (OIA, em inglês: ILO) (SILVEIRA *et al.*, 2006; SILVA *et al.*, 2008; BAVARESCO e SILVEIRA, 2009).

Segundo Silveira *et al.* (2006), um OIA é um agente capaz de desempenhar o papel de um objeto de aprendizagem, agregando características potenciais de adaptação do conteúdo educacional, seguindo o estilo cognitivo, as características do estudante e de acordo com seu desempenho.

Além disso, um OIA pode adquirir novos conhecimentos através da interação com os alunos ou outros OIA (troca de informações entre agentes). A inteligência surge da comunicação e cooperação dos agentes no ambiente, permitindo ao OIA identificar a necessidade de um aluno reforçar (ou aprofundar) seus conhecimentos em determinado conceito ou disciplina. Portanto, a abordagem dos Objetos Inteligentes de Aprendizagem permite a disponibilização de situações de ensino mais completas e reutilizáveis, tornando o processo de aprendizagem do aluno com alto valor agregado.

Propõe-se uma extensão ao modelo OIA prévio, alterando-se a modelagem do agente que cumpre o papel de um OA, com base em uma arquitetura de agentes cognitivos e com a capacidade de comunicação com um LMS (o Moodle, por exemplo) e com objetos de aprendizagem compatíveis com um padrão de integração de recursos educacionais (tais como o SCORM) e com um padrão de metadados (ex.: IEEE-LOM).

Desta forma, os agentes podem deliberar sobre a necessidade de complementação de uma experiência de aprendizagem (ou sessão de aprendizagem), de acordo com os elementos do modelo de dados

transmitidos entre o OA e o LMS. Estes dados da interação do aluno com o OA serão mapeados para crenças dos agentes (modelo de interação). A estrutura completa de informações utilizadas pelos agentes será detalhada na próxima seção (4.1, “Modelo Teórico”).

De acordo com o comportamento e o desempenho do estudante representado na base de crenças, o agente irá deliberar a manutenção da experiência de aprendizagem atual, ou a necessidade de se apresentar um novo objeto de aprendizagem relacionado ao mesmo tema de ensino, ou a um assunto pré-requisito.

Conforme mencionado no Capítulo 1 (“Introdução”), deseja-se facilitar o reuso e a adaptatividade de materiais instrucionais em ambientes de aprendizagem. Propõe-se a adaptação do LMS através da seleção dinâmica de objetos de aprendizagem por meio da integração a um SMA, devido às características de agentes, tais como a capacidade de representação de informações sobre usuários, podendo definir o modelo do aluno, e de agregar comportamento inteligente ao sistema (BREMIGARTNER *et al.*, 2014). Estas características mencionadas favorecem o emprego da estratégia de agentes no desenvolvimento de sistemas de apoio ao processo educacional.

O uso de agentes cognitivos na implementação de ambientes dinâmicos de ensino permite que informações obtidas durante a interação do aluno com o OA possam ser utilizadas no processo de raciocínio dos agentes, como critérios de decisão para a seleção de novos objetos de aprendizagem conforme o desempenho do aluno. Além disso, os agentes podem adquirir novos conhecimentos e comportamentos com base nas informações obtidas de cada experiência de aprendizagem (SILVA *et al.*, 2008).

A ideia conceitual de um objeto de aprendizagem, representando um objeto composto que agrega outros recursos mais elementares (como animações, textos, vídeos, etc.) (DA SILVA *et al.*, 2010), também se aplica à abordagem de Objetos Inteligentes de Aprendizagem (OIA).

Desta forma, um OIA (agente) pode ser visto como uma composição dinâmica de objetos de aprendizagem autônomos, ou seja, um conjunto de materiais instrucionais pré-existentes, compatíveis com algum padrão de recursos digitais educacionais. Esta composição (agregação) realizada por um OIA é definida durante a interação do aluno

com os objetos que são ofertados, de acordo com o desempenho que o estudante apresenta.

#### 4.1 MODELO TEÓRICO

A proposta de solução apresentada nesta dissertação está ancorada na possibilidade de se desenvolver ambientes de aprendizagem dinâmicos, permitindo a disponibilização de situações de ensino mais completas e reutilizáveis.

Novos objetos passam a compor a trilha de aprendizagem dinamicamente, fazendo com que o aluno possa expandir, solidificar ou suprir seus conhecimentos em um determinado conteúdo, recebendo os objetos mais adequados, o que potencializa a melhora de seu desempenho educacional (ABBAD, 2007; MATEUS, 2010; DALMON *et al.*, 2012).

Objetos Inteligentes de Aprendizagem (OIA) enriquecem o processo de ensino-aprendizagem ao permitir que vários objetos de aprendizagem simples, em conjunto, possam identificar necessidades de reforço por parte do aluno, através do desempenho deste em uma avaliação, por exemplo (BAVARESCO e SILVEIRA, 2009).

Com a extensão proposta à abordagem de OIA anterior, ocorre a potencialização do processo de ensino-aprendizagem, possibilitando a apresentação de conteúdos diversos e em diferentes estratégias de ensino, sem que para isso o professor ou designer instrucional precisem explicitamente configurar o ambiente virtual de aprendizagem.

Objetos de aprendizagem que venham a ser incorporados no(s) repositório(s) acessíveis pelo agente, inclusive após o início do curso, podem ser adicionados ao LMS (trilha de aprendizagem) mesmo que não tenham sido explicitamente adicionados à estrutura inicialmente montada pelo DI ou professor no LMS, favorecendo o reuso de recursos.

A dinâmica do processo de adaptação é realizada pelo agente que representa um Objeto Inteligente de Aprendizagem, com base em suas crenças sobre o LMS, sobre o objeto de aprendizagem e sobre a interação do aluno com o objeto.

Ao ser identificado que determinado assunto (ou pré-requisito) precisa ser melhor compreendido por parte do aluno (base de crenças), o agente pode modificar o fluxo da experiência de aprendizagem para exibir um novo recurso ao aluno, potencializando o entendimento do indivíduo sobre o assunto específico.

A nota do aluno no curso obtida do LMS e o assunto relacionado à experiência de aprendizagem (tópico) definem os critérios iniciais para o

agente ativar a sessão de aprendizagem, passando a exibir o objeto e monitorar a interação do aluno com este recurso instrucional.

Os elementos (e respectivos valores) definidos no modelo de dados do padrão de integração, para troca de informações entre o conteúdo (OA) e o sistema de aprendizagem (LMS), permitem que o resultado da interação do aluno com o objeto seja avaliado independente do domínio (assunto) relacionado ao artefato instrucional.

Além disso, os metadados do objeto de aprendizagem devidamente preenchidos possibilitam a identificação de qual conteúdo educacional está relacionado ao recurso, e desta forma, os agentes têm a capacidade de identificar, dependendo do desempenho apresentado pelo estudante:

- O assunto que o aluno precisa reforçar seu entendimento ou suprir a lacuna de conhecimento (mesmo conceito ou pré-requisito).
- O assunto que o aluno pode se aprofundar (OA com maior nível de dificuldade).

As informações que os agentes (OIA) utilizam para realizar a adaptação dos recursos instrucionais são armazenadas em suas crenças com base em dados de três elementos envolvidos na experiência (ou sessão) de aprendizagem, que no contexto desta pesquisa simboliza o momento enquanto o aluno acessa o objeto inteligente de aprendizagem.

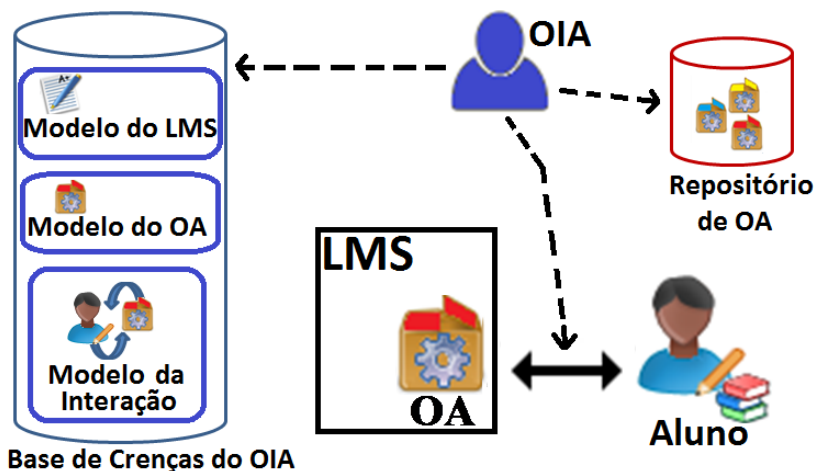
Assim, o processo de decisão do agente que representa um objeto inteligente de aprendizagem não se limita a dados do LMS ou dos próprios objetos de aprendizagem (metadados), mas vai além, utilizando informações obtidas durante a interação do aluno com o OA (“*on the fly*”), segundo um padrão de empacotamento e sequenciamento de recursos.

O primeiro elemento que integra a abstração da experiência de aprendizagem é o LMS (ou ambiente virtual de aprendizagem), o segundo componente é o objeto de aprendizagem propriamente dito (recurso instrucional) e o terceiro elemento é a própria interação do aluno com o



objeto. A Figura 13 ilustra os elementos da arquitetura abstrata do modelo teórico proposto.

Figura 13 – Elementos da Arquitetura do Modelo Teórico



Fonte: Autor (2015)

Os dados referentes ao LMS (Modelo do LMS) representam informações sobre o estudante no ambiente virtual, tais como: nome do aluno, tópico (assunto) no qual o aluno se encontra atualmente no curso e seu desempenho no curso (nota). Tais informações são utilizadas para inicializar uma sessão de aprendizagem, fazendo com que o OIA (agente) passe a exibir o objeto e monitorar a interação do aluno com este recurso educacional.

Os dados referentes ao Objeto de Aprendizagem (Modelo do OA) advêm dos metadados deste objeto, permitindo a identificação de detalhes sobre o recurso instrucional, tais como: título, palavras-chave, nível de dificuldade, tipo de recurso educacional, nível de interatividade, contexto educacional de uso, formato, tamanho, propósito e classificação taxonômica de assuntos e/ou pré-requisitos.

Os dados da interação do aluno com o objeto de aprendizagem (Modelo da Interação) são os elementos (e respectivos valores) definidos no modelo de dados do padrão de integração, para troca de informações entre o conteúdo e o sistema de aprendizagem (LMS), como por exemplo: tempo da sessão de aprendizagem, medida de progresso nos itens que

compõem o objeto de aprendizagem (completude), resultados de respostas a questões avaliativas (questionários) e desempenho obtido no OA (sucesso ou falha).

Com o intuito de ampliar o Modelo do OA (objeto de aprendizagem), propõe-se que os elementos definidos nas crenças do agente sejam utilizados em conjunto com ontologias para a busca por objetos em repositórios com base em: (1) conceitos semanticamente similares, através de ontologias de domínio (GASCUEÑA *et al.*, 2006; BOYCE e PAHL, 2007), e (2) por informações qualificadas dos metadados de objetos de aprendizagem, com base em ontologias de padrões de metadados (GHEBGHOUB *et al.*, 2008; GLUZ e VICARI, 2011). Inclusive estas ontologias poderiam ser integradas a outras ontologias já desenvolvidas pela comunidade, e também aos conceitos de *Linked Data* e *Web Semântica* (ex.: DBPedia) (JAIN *et al.*, 2010).

As ontologias de domínio permitirão a expansão da busca por objetos cujos assuntos sejam sinônimos aos valores definidos nos metadados do OA. Um exemplo seria um objeto de aprendizagem sobre a história do Brasil, cujo assunto definido no elemento “título” ou “palavra-chave” do seu metadado fosse “Guerra dos Farrapos”. A busca por objetos relacionados com base na ontologia de domínio poderia obter recursos que fossem identificados por “Revolução Farroupilha” (busca semântica).

Já a ontologia que defina a estrutura dos elementos que compõem a especificação do respectivo padrão de metadados permitirá a busca por elementos relacionados, ou por valores dos elementos que sejam semanticamente associados, compondo links semânticos com outras bases (repositórios) de objetos de aprendizagem.

Para exemplificar esta funcionalidade, pode-se citar a busca por um tipo de objeto de aprendizagem que seja composto de um diagrama sobre fotossíntese. A pesquisa por objetos relacionados não iria apenas retornar àqueles que possuíssem a palavra “diagrama” definida no elemento “tipo de recurso educacional”, mas também retornaria objetos com tipos de recurso que semanticamente tivessem significado semelhante, tais como: “gráfico” ou “uso de figuras e sequenciamento”.

Além disso, poderia se buscar por objetos de aprendizagem sobre “Lei de Ohm”, onde a ontologia permitisse expandir o significado da palavra de busca, de tal forma que fossem pesquisados objetos que tivessem em seus metadados “Lei de Ohm” como “assunto”, como “tópico”, como “objetivo”, e assim por diante.

Pode-se também empregar ontologias de padrões de metadados de objetos de aprendizagem existentes e publicadas na literatura, para

expandir a forma de busca por objetos implementada no protótipo que será detalhado no Capítulo 5 (“Resultados”) (BRASE e NEJDL, 2004; WANG, 2008; LEE *et al.*, 2008; GLADUN *et al.*, 2009; DAG e ERKAN, 2010).

A ideia é futuramente realizar a consulta não só pelas palavras-chave (*keywords*), título (*title*) e assunto (*subject*) como *strings* “fixas”, mas também buscar por elementos de metadados que sejam variações semânticas (sinônimos) a estes termos, como por exemplo: “assunto” como “descrição”, “tópico”, “item”, “unidade”, e até mesmo a união destes e de outros termos.

Entretanto, destaca-se que no protótipo implementado para validar o modelo teórico proposto, não será realizada a integração com ontologias para expandir a busca semântica por objetos de aprendizagem. Estas restrições do protótipo serão detalhadas no próximo capítulo (Capítulo 5, “Resultados”).

Com base nas informações descritas, o novo modelo de agentes, agora denominado ILOMAS (acrônimo em inglês para Sistema Multiagente de Objetos Inteligentes de Aprendizagem: *Intelligent Learning Object Multi-Agent System*), possui características que possibilitam a seleção dinâmica de objetos de aprendizagem, no contexto do LMS, conforme o desempenho do aluno, tanto na interação interna com o OA (elementos do padrão de integração de objetos de aprendizagem) quanto na externa (realizada dentro do curso através do LMS), sem a necessidade de inclusão de todos os possíveis recursos instrucionais (OA) na estrutura do curso.

O fluxo da adaptação realizada por um OIA segundo o novo modelo será detalhado no decorrer desta dissertação, principalmente no capítulo que descreve a implementação do modelo teórico apresentado (Capítulo 5, “Resultados”).

## 4.2 MODELAGEM DO SMA

O modelo do sistema multiagente foi especificado segundo a metodologia de Engenharia de Software Orientada a Agentes (AOSE) Prometheus (DAM e WINIKOFF, 2013), a qual define um processo detalhado para especificação, projeto e implementação de sistemas de agentes inteligentes, baseados na arquitetura BDI.

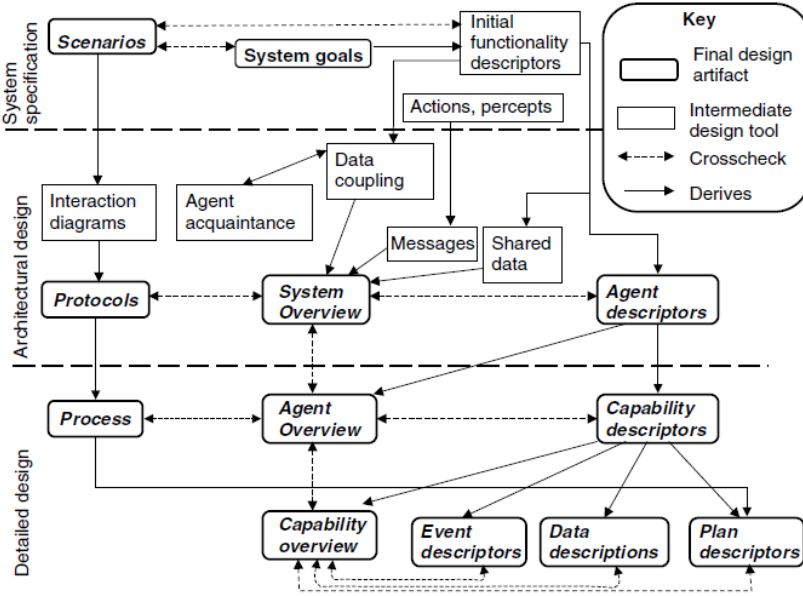
A metodologia Prometheus é composta de diversas atividades divididas em três fases: (1) especificação do sistema, (2) projeto da arquitetura (ou projeto arquitetural) e (3) projeto detalhado. Cada fase apresenta resultados intermediários, tais como diagramas e documentos

de especificação, os quais permitem o desenvolvimento de aplicações orientadas a agentes de forma iterativa (PADGHAM e WINIKOFF, 2004).

A adoção da metodologia Prometheus foi favorecida também pela existência da ferramenta PDT (Prometheus *Design Tool*), um *plug-in* para a IDE Eclipse que permite o desenho dos diagramas que compõem a metodologia (SUN *et al.*, 2010). A ferramenta PDT facilita o processo de desenvolvimento de sistemas multiagente, possibilitando que o projetista se concentre em uma fase do projeto de cada vez, identificando gradualmente todos os componentes do sistema, porém com a opção de retornar a fases anteriores, de forma iterativa.

Embora a metodologia permita o desenvolvimento de diversos artefatos em cada uma de suas fases, não há a exigência de que todos os tipos sejam produzidos, ficando ao critério e à necessidade do projetista a decisão de quais artefatos utilizar em sua modelagem. A Figura 14 ilustra as fases da metodologia Prometheus, bem como as interligações entre os elementos produzidos em cada uma das fases.

Figura 14 – Fases da Metodologia Prometheus



Seguindo o processo da metodologia Prometheus, na fase de especificação do sistema, inicialmente foram identificados:

- Objetivos – que representam aquilo que o sistema (como um todo) e os agentes (individualmente) devem fazer.
- Papéis – são os comportamentos esperados do agente que cumprir o papel especificado, englobando “pedaços” de funcionalidades, tais como objetivos, percepções e ações.
- Agentes – são as entidades inteligentes do sistema.
- Interfaces – são ligações entre os agentes e o ambiente, em termos de mensagens, percepções, ações e dados externos (acesso a um banco de dados, por exemplo).

O resultado da modelagem através da metodologia Prometheus levou ao desenvolvimento da arquitetura do ILOMAS, que contempla os seguintes agentes: (1) LMSAgent e (2) ILOAgent.

O primeiro tipo de agente modelado foi o LMSAgent. Este agente é a representação do LMS, e fica responsável por:

- Receber a requisição do aluno para uma nova experiência de aprendizagem.
- Identificar qual o tema que o estudante deve aprender no momento (objetivo *IdentifyLearningExperienceSubject*).
- Passar o controle da interação com o aluno para um novo agente do tipo ILOAgent. Suas crenças advêm de informações da base de dados do LMS, tais como estudantes inscritos, o curso e o assunto que o aluno deve aprender. A comunicação entre este agente e o LMS é feita através de uma API de acesso a banco de dados (ex.: Java JDBC, PHP PDO, etc.) ou via *Web Service*.

O segundo tipo de agente modelado foi denominado ILOAgent. Este agente representa o OIA preconizado no modelo teórico, podendo ser considerado uma interface para acesso a objetos de aprendizagem físicos. Estes recursos instrucionais físicos são exibidos ao estudante

conforme o modelo de interação do aluno com o objeto, modelo este representado nas crenças do ILOAgent.

Além de possuir a competência de exibir objetos e monitorar a interação de um estudante com estes recursos ofertados, este agente tem a capacidade de se comunicar com repositórios de objetos de aprendizagem, para a obtenção de novos materiais educacionais que complementam a experiência de aprendizagem.

O processo de exibição da experiência educacional ao estudante, bem como seu monitoramento, pode ser descrito da seguinte forma:

- Inicialmente o agente deve apresentar o objeto de aprendizagem relacionado ao assunto definido na crença do agente (obtida da comunicação com o LMSAgent).

Para isso, o ILOAgent precisa obter, no repositório de objetos, um OA relativo ao tema que o aluno precisa estudar, para então exibi-lo ao estudante (objetivo *SearchLearningObject*). Os metadados dos objetos de aprendizagem são critérios para a escolha de qual recurso instrucional será ofertado.

- Além disso, este tipo de agente deve monitorar a interação entre o estudante e o objeto de aprendizagem (objetivo *PlayLearningObject*), sendo esta a principal tarefa deste agente, e a característica mais importante proporcionada pelo SMA, do ponto de vista da seleção dinâmica de conteúdo.

O mecanismo de deliberação do ILOAgent irá definir se o aluno precisa complementar a experiência atual com um objeto extra (se houver demora no acesso ao objeto, por exemplo), alterando-se o conteúdo do curso dinamicamente, ou se o estudante deve continuar interagindo com o OA atual.

Essa decisão é tomada com base nos dados obtidos da interação, ou seja, os elementos do modelo de dados do padrão de integração de recursos instrucionais (ex.: duração da sessão, nota no questionário, grau de progresso no OA, etc.), bem como no histórico das interações prévias.

Assim, o ILOAgent pode ser entendido como um meta-objeto que se apropria de um ou mais objetos de aprendizagem físicos. Este agente mantém sua existência durante a sessão de aprendizagem, que é o período de interação do aluno com os objetos que são ofertados através do LMS. Os dados de cada objeto ofertado durante a experiência de aprendizagem

(sessão de aprendizagem) do respectivo aluno são armazenados na base de crenças do ILOAgent.

Julga-se oportuno mencionar uma alternativa a esta estratégia adotada, quanto à correspondência entre um ILOAgent (OIA) e os objetos de aprendizagem físicos. Ao invés de um ILOAgent representar uma agregação de objetos físicos, poderia ser modelado o ILOAgent como a representação de um único objeto físico.

Com esta abordagem alternativa, ao ser identificada a necessidade se ofertar um objeto complementar, o agente OIA sairia de cena (finalizaria sua execução) e passaria o controle (bem como todas as informações por ele colhidas) para um novo ILOAgent. Uma das vantagens desta alternativa seria que cada ILOAgent seria especialista em um objeto de aprendizagem, e seria capaz de aprender sobre o conceito associado ao objeto.

Entretanto, por razões de facilidade de implementação, principalmente quanto à transferência de informações entre agentes, essa segunda alternativa não foi utilizada, sendo uma possibilidade a ser melhor avaliada em trabalhos futuros.

Com relação aos dois agentes modelados, é oportuno classificá-los quanto à abordagem teórica, à arquitetura e à plataforma de implementação. Do ponto de vista teórico, o LMSAgent pode ser classificado como reativo. Seu comportamento é simplificado, servindo basicamente para obtenção de informações da base de dados do LMS, e não participando do processo de seleção dinâmica de objetos de aprendizagem propriamente dito.

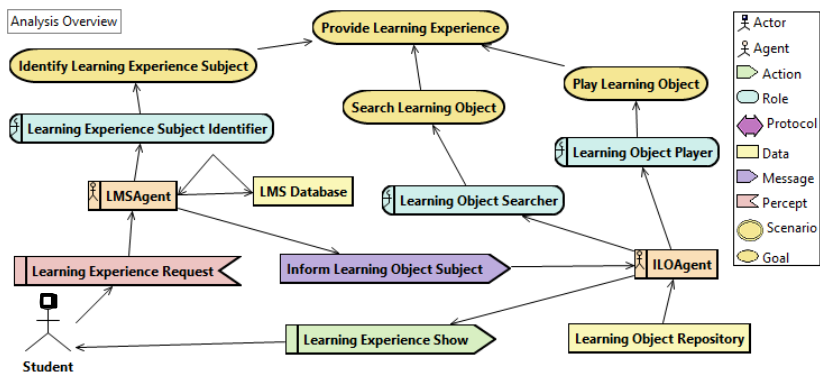
Em contrapartida, o ILOAgent é a entidade principal do modelo teórico proposto, pois conforme mencionado previamente, a adaptação do conteúdo educacional é realizada por este agente (OIA). O ILOAgent é classificado como racional (cognitivo), segundo a teoria (modelo) BDI, baseado na arquitetura PRS (com algumas alterações) e implementado segundo a plataforma Jadex. Maiores detalhes sobre a implementação do ILOAgent serão descritos no capítulo 5 (“Resultados”).

Prosseguindo no detalhamento das etapas de análise e projeto do sistema multiagente, serão descritos na sequência os subprodutos gerados em cada uma das fases da metodologia Prometheus.

Os diagramas confeccionados na etapa de especificação do sistema através da ferramenta PDT foram o diagrama de visão geral da análise, o

diagrama de objetivos do sistema e o diagrama de visão geral dos papéis do sistema. A Figura 15 exibe o diagrama de visão geral da análise.

Figura 15 – Diagrama da Visão Geral da Análise



Fonte: Autor (2015)

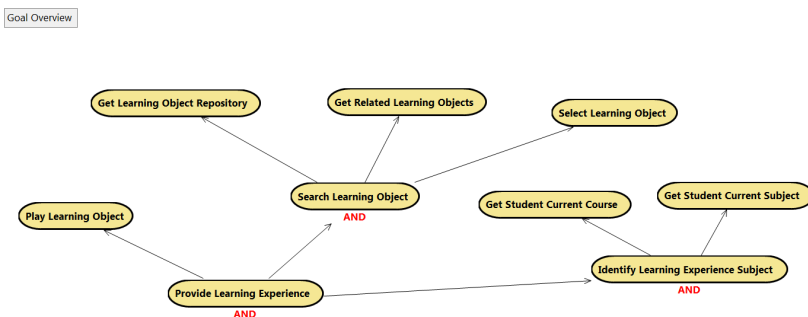
Com relação aos objetivos elencados, cabe ressaltar que estes foram modelados segundo uma hierarquia de objetivos, conforme possibilitado pela metodologia Prometheus. Desta forma, um objetivo mais genérico foi subdividido em outros mais específicos, com o intuito de decompor cada responsabilidade da forma mais específica possível.

Como exemplo pode ser destacado o objetivo de se identificar o assunto da experiência de aprendizagem (*Identify Learning Experience Subject*), o qual foi decomposto em dois sub-objetivos: obter o curso atual que o estudante está matriculado (*Get Student Current Course*), e obter o



assunto atual que o estudante deve aprender (*Get Student Current Subject*). A Figura 16 ilustra o diagrama de objetivos do sistema.

Figura 16 – Diagrama de Objetivos do Sistema



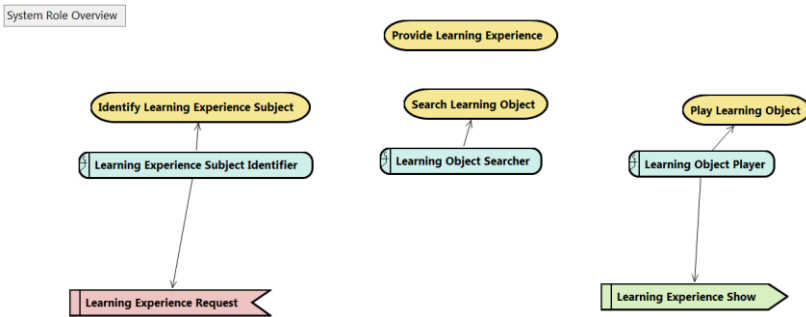
Fonte: Autor (2015)

O diagrama de papéis do sistema descreve os objetivos que devem ser alcançados por cada papel, além de identificar as percepções e ações relacionadas com um papel. Como exemplo, um dos papéis existentes no Sistema ILOMAS é o de identificador do assunto relacionado à experiência de aprendizagem (papel *Learning Experience Subject Identifier*), o qual tem como objetivo a identificação deste assunto. Este

papel está relacionado à percepção de que um aluno requisitou uma experiência de aprendizagem (percepção *Learning Experience Request*).

O assunto (tópico) inicial da experiência de aprendizagem é obtido do sistema de gerenciamento de aprendizagem (Modelo do LMS). A Figura 17 exibe o diagrama de visão geral dos papéis do sistema.

Figura 17 – Diagrama de Visão Geral dos Papéis



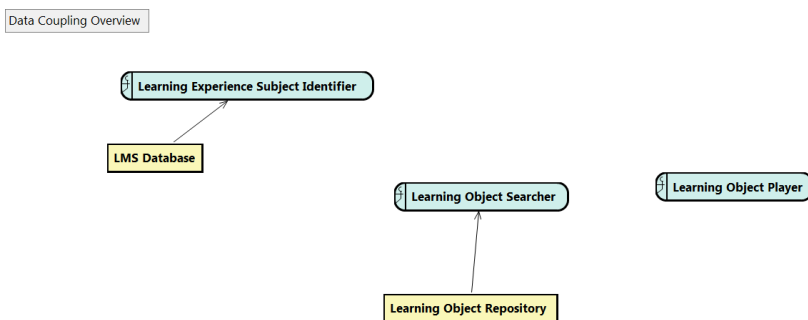
Fonte: Autor (2015)

A segunda fase da metodologia Prometheus, denominada projeto arquitetural, possibilita a definição de mais detalhes da arquitetura do sistema com base nos agentes e nas interações entre eles, visando o agrupamento de funcionalidades.

O primeiro diagrama confeccionado na fase do projeto arquitetural foi o diagrama de acoplamento de dados, o qual relaciona os papéis identificados na fase anterior com as fontes de dados externas do sistema. O papel identificador do assunto relacionado à experiência de aprendizagem (papel *Learning Experience Subject Identifier*) obtém informações da base de dados do LMS, enquanto o papel de busca do objeto de aprendizagem (papel *Learning Object Searcher*) faz uso do

repositório de objetos de aprendizagem. A Figura 18 ilustra o diagrama de acoplamento de dados.

Figura 18 – Diagrama de Acoplamento de Dados



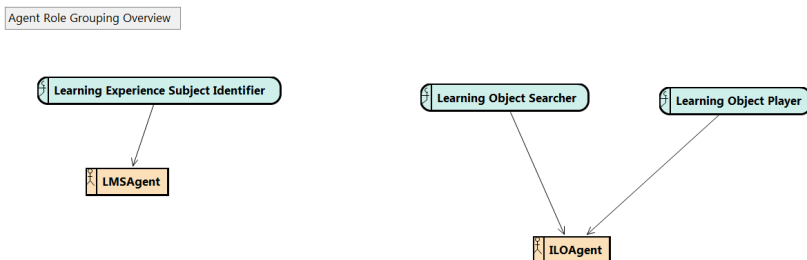
Fonte: Autor (2015)

O segundo diagrama gerado na fase do projeto arquitetural foi o diagrama de relacionamento entre papéis e agentes, o qual apresenta o papel (ou os papéis) que cada agente modelado realiza. O agente LMSAgent realiza o papel de identificador do assunto da experiência de aprendizagem (papel *Learning Experience Subject Identifier*).

O agente ILOAgent realiza o papel de buscador do objeto de aprendizagem, além do papel que tem a incumbência de exibir o objeto

de aprendizagem ao aluno (papéis *Learning Object Searcher* e *Learning Object Player*, respectivamente, conforme exibido na Figura 19).

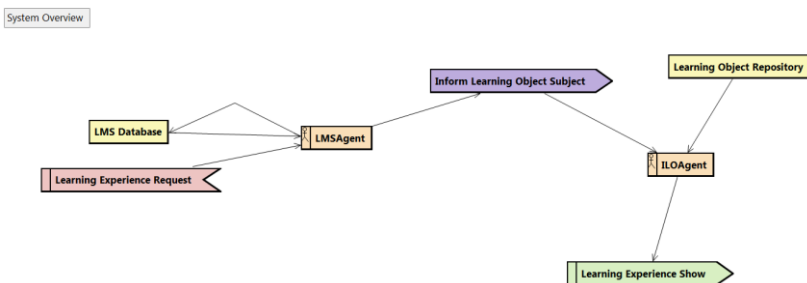
Figura 19 – Diagrama de Relacionamento entre Papéis e Agentes



Fonte: Autor (2015)

O último diagrama desenvolvido na fase do projeto arquitetural foi o diagrama de visão geral do sistema, o qual (como o próprio nome diz) descreve a estrutura geral do sistema proposto, exibindo os agentes modelados, suas respectivas interações (mensagens), percepções e ações. A Figura 20 ilustra o diagrama de visão geral do sistema.

Figura 20 – Diagrama de Visão Geral do Sistema



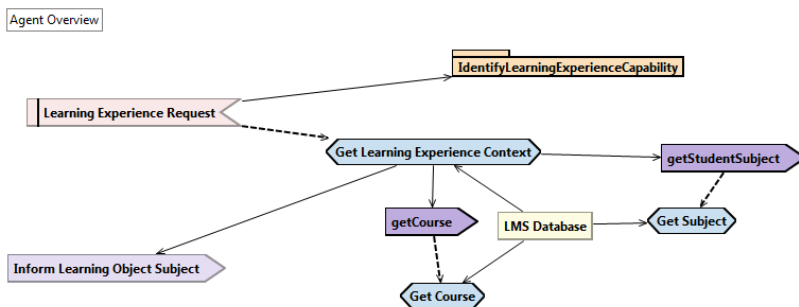
Fonte: Autor (2015)

A terceira e última fase da metodologia Prometheus é denominada projeto detalhado. Esta etapa resultou na confecção dos diagramas de visão geral dos agentes LMSAgent e ILOAgent. Este tipo de diagrama traz detalhes mais internos de funcionamento do agente, apresentando planos, percepções, ações, mensagens internas, além de *capabilities*

associadas ao agente. Os planos são a maneira concreta de realizar os objetivos que cada agente deve alcançar.

A Figura 21 exibe o diagrama de visão geral do LMSAgent.

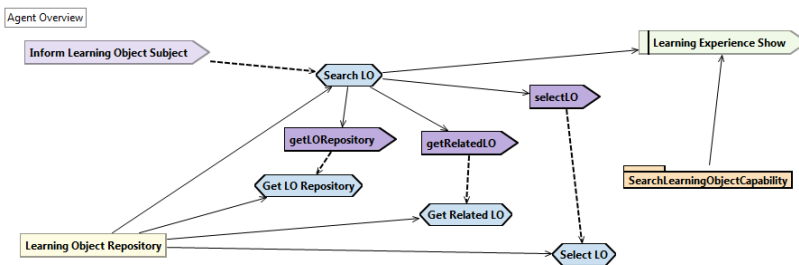
Figura 21 – Diagrama de Visão Geral do Agente – LMSAgent



Fonte: Autor (2015)

A Figura 22 ilustra o diagrama de visão geral do ILOAgent.

Figura 22 – Diagrama de Visão Geral do Agente – ILOAgent



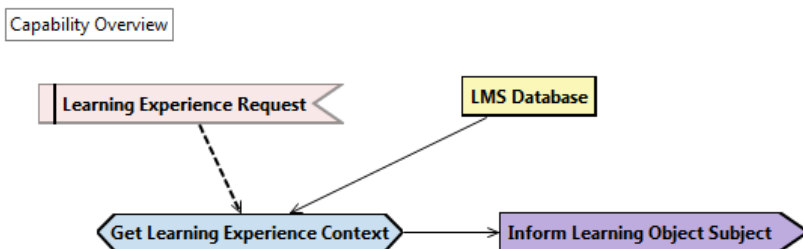
Fonte: Autor (2015)

Um dos resultados da etapa do projeto detalhado foi a confecção do diagrama de *capabilities*. Este diagrama exibe os detalhes de cada *capability* dos agentes, demonstrando os planos associados, além das fontes de dados, mensagens, percepções e ações. A ideia principal do uso de *capabilities* é modularizar funcionalidades dos agentes, permitindo o reuso de características comuns entre agentes distintos que possuem

algum comportamento específico em comum (como planos, objetivos, crenças, etc.).

A Figura 23 exibe o diagrama da visão geral da *capability* denominada *IdentifyLearningExperienceCapability*.

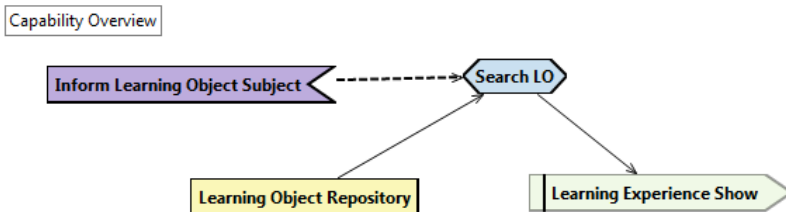
Figura 23 – Diagrama de *Capability – IdentifyLearningExperienceCapability*



Fonte: Autor (2015)

A Figura 24 ilustra o diagrama da visão geral da *capability* denominada *SearchLearningObjectCapability*.

Figura 24 – Diagrama de *Capability – SearchLearningObjectCapability*



Fonte: Autor (2015)

Seguindo a fase do projeto detalhado, foram modelados os planos associados a cada um dos objetivos identificados nas etapas anteriores. Os planos são conjuntos ordenados de ações que o agente deve executar para atingir um determinado objetivo (UEZ, 2014).

A especificação de um plano envolve a definição das situações que devem ser verdadeiras (contexto) para que o plano possa ser executado. Para os planos vinculados aos objetivos previamente elencados, as

condições de disparo (pré-condições) estão atreladas à condição para seleção do objetivo relacionado. A descrição completa do plano é realizada na etapa de implementação, de acordo com a sintaxe da ferramenta (framework) de desenvolvimento de agentes utilizada.

Além dos planos associados aos objetivos detectados na fase de especificação do sistema multiagente, foram projetados planos para tratamento de falhas que possam ocorrer durante o processo de oferta do objeto de aprendizagem ao aluno, na sessão de aprendizagem, enquanto o aluno está interagindo com o recurso instrucional. Tratar falhas pode ser considerado uma forma de adaptação, resultando na ampliação do escopo de atuação do agente, e em um sistema mais robusto.

Os planos para o tratamento de falhas foram divididos em dois tipos: (1) para falhas técnicas e (2) para falhas educacionais. Estes tipos de planos estão associados aos objetivos: tratar falha técnica (*Handle Technical Error*) e tratar falha educacional (*Handle Educational Failure*), respectivamente. Tais objetivos foram incorporados ao agente ILOAgent (representa o Objeto Inteligente de Aprendizagem).

Destaca-se que os objetivos e respectivos planos associados às falhas técnicas e educacionais não foram incluídos na estrutura do LMSAgent, devido à característica basicamente reativa deste tipo de agente. Desta forma, os planos que serão apresentados na sequência compõem o modelo do ILOAgent (OIA).

Os planos de falhas técnicas estão relacionados a algum problema operacional que impede o fluxo de execução normal do sistema e que inviabiliza o acesso do estudante ao objeto de aprendizagem por questões tecnológicas. O Quadro 4 exhibe os planos modelados para tratamento de

falhas técnicas (em abstrações de mais alto nível), descrevendo as pré-condições de disparo (contexto) do plano e as ações resultantes.

Quadro 4 – Planos para tratamento de falhas técnicas

<b>Plano</b>	<b>Situação (Contexto)</b>	<b>Ação</b>
trataErroTecnicoConexao	Sem conexão com o Repositório escolhido	Estabelecer conexão com outro repositório
trataErroTecnicoFaltaOA	Repositório escolhido não possui OA relacionado aos critérios de busca	Buscar em outro repositório
trataErroTecnicoJavaScript	Erro JavaScript	Plano genérico para erros técnicos (tratarErroTecnico)
trataErroTecnicoBrowser	Erro de compatibilidade no navegador com algum <i>plug-in</i> (ex.: flash, wav, quick time, etc.)	Plano genérico para erros técnicos (tratarErroTecnico)
trataErroTecnicoGenerico	Invocação (explícita) do plano	Exibir mensagem de erro e solicitar que o aluno tente o acesso novamente

Fonte: Autor (2015).

Os planos de falhas educacionais estão relacionados a alguma situação não prevista no processo de ensino-aprendizagem detectada pelos agentes, ou ainda a algum outro tipo de problema que não seja de ordem tecnológica, mas que também impeça o funcionamento correto do processo de oferta de objetos de aprendizagem de acordo com o desempenho do aluno. O Quadro 5 apresenta os planos modelados para tratamento de falhas educacionais (em abstrações de mais alto nível),



descrevendo as pré-condições de disparo (contexto) do plano e as ações resultantes.

Quadro 5 – Planos para tratamento de falhas educacionais

<b>Plano</b>	<b>Situação (Contexto)</b>	<b>Ação</b>
trataFalhaEducRepositorio LocalSemNovoOA	Todos os objetos existentes no repositório local já foram ofertados ao aluno	Buscar OA em outro repositório (externo)
trataFalhaEducRepositorios SemNovoOA	Todos os objetos existentes no repositório local, e em outros acessíveis, já foram ofertados ao aluno	Ofertar o OA inicial que já havia sido exibido ao aluno
trataFalhaEducSemMelhora DeDesempenho	Objetos relacionados aos critérios de busca atual não estão melhorando desempenho do aluno	Buscar OA com base em critérios semânticos (buscaSemanticaOA)
trataFalhaEducSemNovo OACriterioAtual	Não há novos objetos relacionados aos critérios de busca atual a serem ofertados ao aluno	Buscar OA com base em critérios semânticos (buscaSemanticaOA)
trataFalhaEducSemNovo OADiferentesCriterios	Não há novos objetos (nem mesmo de critérios distintos) a serem ofertados ao aluno	Sinalizar ao aluno que ele deve entrar em contato com o tutor ou professor do curso
buscaSemanticaOA	Invocação (explícita) do plano	Buscar OA com base em outros critérios (busca semântica segundo elementos LOM)

Fonte: Autor (2015).

Ressalta-se que outros tipos de tratamento de falhas poderiam ser identificados e desenvolvidos, principalmente para questões educacionais, tais como a busca por novos objetos de aprendizagem com base em critérios que não sejam definidos nos metadados IEEE-LOM, como por exemplo, o estilo de aprendizagem do estudante

(KARAGIANNIDIS e SAMPSON, 2004; DEL MORAL *et al.*, 2013; ANITHA e DEISY, 2013; CARVALHO *et al.*, 2014; SABITHA *et al.*, 2014).

Desta forma, caso o aluno não esteja apresentando melhora no seu desempenho educacional e os objetos de aprendizagem que estejam sendo ofertados possuam informações sobre o estilo de aprendizagem associado, pode-se tratar essa situação como uma falha de ensino, e realizar a busca por objetos relacionados a outros estilos de aprendizagem.

Embora exista essa possibilidade (e com certeza outras), por motivos de limitação de escopo e prazo foram modelados (e implementados) planos para tratamento de apenas dois tipos de falhas: (1) o plano referente ao erro técnico genérico (trataErroTecnicoGenerico) e (2) o plano relacionado à situação onde não há novos objetos a serem ofertados ao estudante (trataFalhaEducRepositoriosSemNovoOA).

Após ser realizada a modelagem do Sistema ILOMAS, com a elaboração dos respectivos subprodutos (diagramas e especificações), foi iniciada a etapa de implementação do modelo, cujos detalhes serão apresentados no próximo capítulo (Capítulo 5, “Resultados”).

### 4.3 REPRESENTAÇÃO DAS CRENÇAS

O modelo conceitual do agente OIA (ILOAgent) utilizado no seu processo de deliberação foi definido conforme uma abordagem de representação de conhecimento. A literatura menciona algumas estratégias possíveis, conforme será detalhado na sequência.

A abordagem de utilização de uma base de conhecimento externa ao agente, a qual centraliza as informações sobre conceitos do domínio do sistema, é comumente utilizada na literatura (HINDRIKS, 2009; HOLMES e STOCKING, 2009). Hindriks (2009) afirma que, em geral, uma base de conhecimento é uma estrutura que mantém fatos de domínio que não se modificam durante a execução do agente. Já a base de crenças mantém as representações internas do agente sobre o mundo, contendo informações que são mutáveis (dinâmicas) durante a execução do agente.

A literatura apresenta ainda estratégias híbridas que utilizam algumas informações armazenadas em bases de conhecimento e outras como crenças dos agentes (DIBLEY *et al.*, 2011). A decisão de qual abordagem utilizar depende das características do sistema multiagente, da dinamicidade das informações e questões de facilidade de projeto e restrições de recursos computacionais.

Cabe destacar que a forma de representar as informações utilizadas pelo OIA (ILOAgent) no processo de deliberação foi definida como

crenças do agente (base de crenças), de tal forma que estes dados foram internalizados na estrutura do ILOAgent e são descritos conforme a sintaxe da plataforma de desenvolvimento de sistemas multiagente adotada (Capítulo 5, “Resultados”).

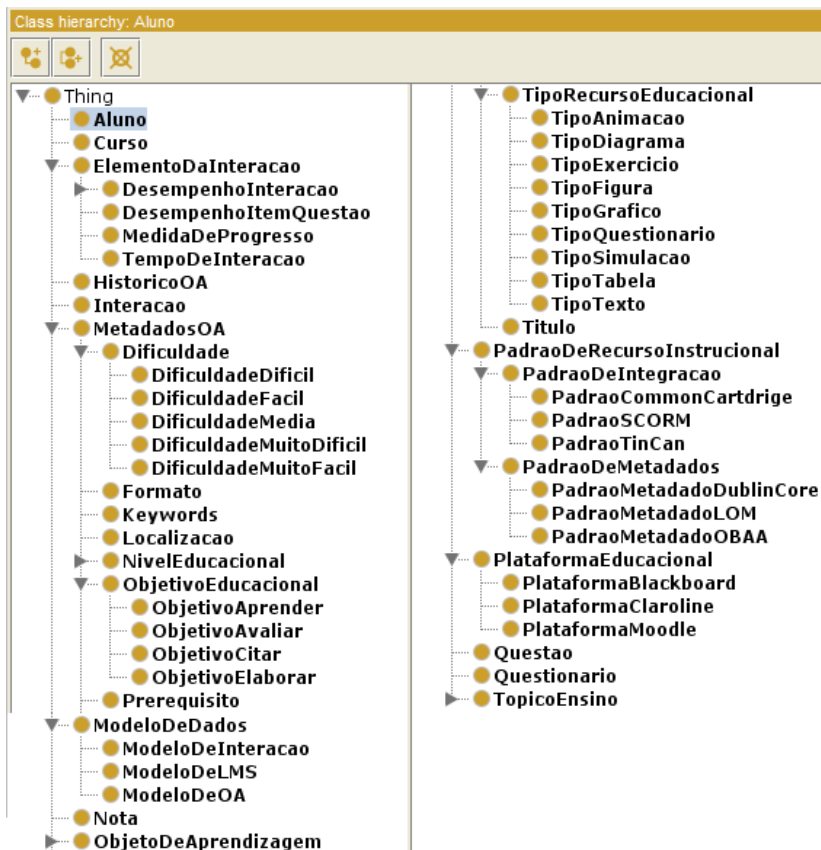
Os elementos do Modelo Conceitual que define a estrutura das crenças do OIA (ILOAgent) foram modeladas através de uma representação ontológica, desenvolvida na ferramenta Protégé. Além de possibilitar a construção das entidades e axiomas envolvidos da ontologia na linguagem OWL, o Protégé possibilita também a geração de código em Java, ou seja, classes (da linguagem de programação) que representam os elementos desta ontologia.

Seguindo as etapas da metodologia 101 (PRIMO, 2013), foram definidos o escopo da ontologia, os termos importantes, as classes e as propriedades, com base nos elementos que compõem o Modelo Conceitual do OIA: o Modelo de LMS, Modelo de OA e o Modelo de Interação (conforme apresentado anteriormente na Figura 13). O intuito é utilizar esta representação ontológica para estruturar as informações que

um ILOAgent utiliza no seu processo de raciocínio, com base apenas nos conceitos e suas relações (TBox).

A Figura 25 exibe as classes modeladas na ontologia.

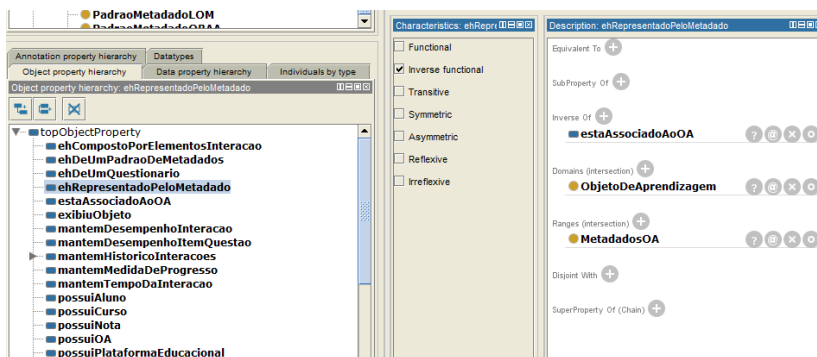
Figura 25 – Classes da ontologia do modelo conceitual do OIA



Fonte: Autor (2015).

A Figura 26 ilustra algumas propriedades identificadas para as classes da ontologia.

Figura 26 – Propriedades da ontologia do modelo conceitual do OIA



Fonte: Autor (2015).

Existem trabalhos em andamento no grupo de pesquisa IATE, além de outras pesquisas da comunidade de sistemas multiagente já publicadas, que propõem modelos e arquiteturas para a junção de agentes e ontologias. Alguns indicam o mapeamento de ontologias para crenças dos agentes, através de classes (em linguagem Java, por exemplo) que encapsulam os conceitos da ontologia como atributos (crenças) do agente (TOMAIUOLO *et al.*, 2006, BREMGARTNER *et al.*, 2014; CAMPOS, 2014).

Há outros estudos que se preocupam com a mudança da programação dos agentes, levando-se em consideração o conhecimento contido na ontologia (representação em lógica descritiva) no processo de raciocínio, como por exemplo, para o disparo de planos do agente (HOLMES e STOCKING, 2009; KLAPISCAK e BORDINI, 2009; SANTOS *et al.*, 2009; REINISCH *et al.*, 2010).

Conforme será descrito no capítulo sobre a implementação do modelo, a inclusão/exclusão de indivíduos na ontologia (ABox) ou a utilização de um motor de inferências (*Reasoner*) está fora do escopo desta dissertação, podendo ser tratado em trabalhos futuros, como uma evolução ao modelo proposto.

Assim, cabe destacar que a ontologia desenvolvida foi utilizada basicamente como uma terminologia (através do OWL) dos elementos do modelo conceitual (TBox), servindo como um meio para se obter as

crenças do ILOAgent. Na implementação do protótipo, a representação das crenças foi elaborada através da ferramenta específica do Protégé que gera uma biblioteca de classes Java (pacote) contendo as entidades, propriedades e relacionamentos da ontologia. O próximo capítulo (Capítulo 5, “Resultados”) apresentará maiores detalhes do protótipo.

Entretanto, ressalta-se que uma vantagem extra advinda da definição da ontologia é a possibilidade futura de utilização no ILOMAS dos conceitos desta ontologia para expandir semanticamente o domínio de busca por objetos de aprendizagem, contextualizando a informação com base nos elementos de metadados, e apoiado em tecnologias da *Web Semântica* (como OWL, SPARQL, etc.).

#### 4.4 ESTRATÉGIAS ADOTADAS NO MODELO TEÓRICO

Esta seção apresenta as premissas que motivaram as decisões de projeto tomadas e as estratégias adotadas para a concepção do modelo teórico proposto nesta dissertação.

Segundo Padgham e Winikoff (2004), em aplicações realísticas, agentes geralmente possuem limitados recursos computacionais e capacidade de perceber seu ambiente, de tal forma que conceitos de uma arquitetura cognitiva, como BDI, auxiliam na eficiência do sistema.

Dentre os elementos que compõem a arquitetura BDI, Padgham e Winikoff (2004) salientam a importância das crenças, as quais permitem ao agente representar as informações obtidas do ambiente e construir seu conhecimento sobre o mundo (ambiente) ao longo do tempo. Além disso, os autores destacam que questões computacionais e de representação são simplificadas através do desenvolvimento de planos pré-definidos como meio prático (ações) de se atingir os objetivos (desejos) do agente.

A literatura do campo da informática na educação apresenta diversos sistemas inteligentes de apoio ao ensino, cuja arquitetura está baseada em uma sociedade de agentes que trabalham de forma colaborativa, composta por agentes puramente reativos, puramente cognitivos ou ainda em uma abordagem híbrida (WILGES *et al.*, 2012). Estes agentes podem interagir diretamente com os usuários (alunos) ou atuar em segundo plano (*background*) compondo a estrutura do sistema instrucional (JAQUES, 2004).

Mateus (2010) cita que a ideia central da adoção dos estados mentais do modelo BDI se concentra no fato de que o agente cognitivo possui estados internos relacionados ao estado do ambiente com o qual interage. Mikic-Fonte *et al.* (2010) mencionam que agentes BDI são

indicados para sistemas que possuam um ambiente variável e que recebam informações continuamente.

Segundo Leung *et al.* (2013), a abordagem BDI permite que interesses individuais, motivações e objetivos dos agentes possam ser facilmente monitorados, manipulados e modificados para ajustar-se a dinamicidade do ambiente virtual, incluindo as atividades dos usuários.

Wooldridge (2002 *apud* BONSON, 2013), Móra (1999 *apud* JAQUES, 2004) e Mikic-Fonte *et al.* (2010) ressaltam as vantagens em se utilizar uma arquitetura BDI, devido às principais qualidades apresentadas por esta abordagem, tais como:

- A forma de raciocínio que a abordagem BDI descreve é bastante intuitiva para nós seres humanos.
- A abstração dos estados mentais facilita a tarefa de modelagem de um agente, a qual pode ser realizada pelos projetistas com baixa complexidade.
- A analogia com estados mentais também fornece descrições concisas sobre sistemas complexos, auxiliando a compreensão do comportamento destes sistemas.
- As abstrações podem ser utilizadas pelos próprios agentes para raciocinar sobre eles mesmos e sobre outros agentes.
- Os sistemas multiagente BDI fornecem uma decomposição clara de suas partes.
- Abstrações de mais baixo nível podem ser evitadas, devido à possibilidade de se desenvolver diferentes planos para tratar um mesmo objetivo que o agente deve alcançar.

A decomposição do sistema em partes menos complexas, as abstrações fornecidas pelos estados mentais e o desenvolvimento de planos foram características que facilitaram a elaboração do modelo de agentes proposto nesta dissertação.

Embora a habilidade de os agentes raciocinarem sobre eles mesmos, ou sobre outros agentes, tenha sido citada como uma vantagem do modelo BDI, cabe destacar que os agentes projetados não possuem tal capacidade. Entretanto, esta característica poderá ser acrescentada futuramente, como uma evolução à modelagem atual.

Ainda com relação às características do modelo BDI, Jaques (2004) sinaliza que esta abordagem é apropriada também quando o sistema está inserido em um ambiente variável e possui acesso a

informações parciais deste ambiente, como é o caso dos sistemas educacionais. Ainda segundo Jaques (2004, p. 39), “as crenças são formas de representar os estados deste mundo onde as informações são dinâmicas e quando o sistema tem uma visão local do mundo”.

Além disso, podem ser citadas como vantagens em se adotar a abordagem BDI na concepção do modelo proposto:

- A dinamicidade e frequência das informações sobre a interação do aluno com o objeto de aprendizagem.
- A capacidade de os agentes aprenderem sobre o estudante durante a experiência de aprendizagem, e também entre diferentes experiências (sessões).
- A característica de adaptação do conteúdo de forma descentralizada fornecida pelos agentes cognitivos que representam um OIA (deliberação).
- A independência de domínio educacional, pois os agentes são especialistas em monitorar a interação do aluno com o objeto, seja qual for o assunto relacionado ao recurso.
- A possibilidade de se agregar novas funcionalidades (planos) e capacidades aos agentes com base em soluções já definidas na literatura (ex.: busca semântica por objetos com base em ontologias).
- A facilidade em se tratar erros de aprendizagem por parte do aluno e falhas operacionais (falta de conexão, OA não existente no repositório, etc.).

Dentre estas vantagens supracitadas, destaca-se que o modelo proposto não apresenta a capacidade de aprender sobre o aluno entre experiências de aprendizagem distintas (com o passar das experiências), podendo esta característica ser acrescentada ao modelo futuramente.

A forma como o modelo foi projetado, e como o protótipo foi implementado, permitirão também que futuramente sejam acrescentados, com relativa facilidade, outras funcionalidades aos agentes através do desenvolvimento de novos planos, como por exemplo, para a busca semântica por objetos de aprendizagem com base em ontologias.

Após serem descritos detalhes das estratégias adotadas no modelo teórico, com relação à arquitetura de agentes cognitivos, é importante



mencionar também outra estratégia empregada quanto à forma de modelagem dos dados.

O modelo de dados do sistema foi projetado com base no modelo de interação do aluno com um recurso instrucional, e não apenas a abordagem comumente adotada nos sistemas tutores inteligentes (STI) apresentados na maioria das pesquisas de informática na educação: o modelo do aluno (PALOMINO, 2013; VESIN *et al.*, 2013; BREMGARTNER *et al.*, 2014).

A ideia do modelo de interação é obter informações mais apuradas, frequentes e atualizadas da dinâmica de comunicação entre o estudante e o objeto de aprendizagem, monitorando todos os estágios do contato do aluno com o recurso educacional (SILVEIRA, 2001).

Esta ideia é semelhante ao que foi descrito por Bolzan e Giraffa (2002) sobre a ampliação da arquitetura clássica de STI proposta por Self (1999 *apud* BOLZAN e GIRAFFA, 2002, p. 8), modelando “a interação que ocorre ao longo de uma sessão de trabalho entre o aluno e o ambiente”.

Segundo Goulart (2000 *apud* BOLZAN e GIRAFFA, 2002, p. 8),

o módulo do estudante não mais relaciona somente as informações sobre a análise das interações do aluno com o domínio, mas busca uma contextualização maior destas interações em função das ações do aluno, o contexto em que elas ocorrem e a estrutura cognitiva do aluno naquele momento (Modelo de Interação).

Assim sendo, o processo de deliberação dos agentes pode ser enriquecido com os dados obtidos da interação do estudante com o objeto de aprendizagem, potencializando a adaptação da experiência educacional às características mais atuais apresentadas pelo aluno durante a sessão de aprendizagem (ex.: desempenho em um questionário), bem como possibilitando um feedback mais rápido ao estudante (DALMON *et al.*, 2011).

O próximo capítulo (Capítulo 5, “Resultados”) descreve a implementação de um protótipo com base no modelo teórico apresentado neste capítulo. Assim, serão apresentados os detalhes da implementação do sistema ILOMAS, mencionando as tecnologias utilizadas, as decisões tomadas na etapa de programação, os artefatos desenvolvidos e as avaliações do modelo realizadas com base na execução do protótipo em simulações de curso.



## 5 RESULTADOS

Conforme mencionado em capítulos anteriores, o modelo teórico proposto é uma extensão à abordagem de Objetos Inteligentes de Aprendizagem (OIA) apresentada em trabalhos prévios. Este modelo teórico foi instanciado através da implementação de um protótipo de aplicação, projetado para utilizar um subconjunto dos elementos disponíveis em cada um dos padrões que compõem o modelo.

O protótipo implementado permite a integração da abordagem OIA com recursos instrucionais no padrão SCORM, através de uma arquitetura de agentes BDI, possibilitando a tomada de decisão pelos agentes, conforme o desempenho apresentado pelo aluno durante a interação com o recurso realizada dentro do LMS Moodle.

Desta forma, o Modelo de Dados implementado, o qual representa a base de crenças de um OIA, possui a seguinte configuração:

- O Modelo de LMS se baseia em algumas informações da base de dados (tabelas) do LMS Moodle.
- O Modelo de Objeto de Aprendizagem foi especificado segundo o padrão de metadados IEEE-LOM.
- O Modelo de Interação (Aluno-OA) seguiu o modelo de dados do padrão de integração de objetos de aprendizagem SCORM.

A próxima seção apresenta as premissas que motivaram as decisões de implementação tomadas e as tecnologias adotadas no desenvolvimento do protótipo.

### 5.1 TECNOLOGIAS UTILIZADAS NA IMPLEMENTAÇÃO

O desenvolvimento do protótipo que implementa o modelo de agentes cognitivos foi realizado com base na plataforma Jadex, a qual é citada em diversos trabalhos na literatura de sistemas multiagente (BALTHAZAR *et al.*, 2010; DIBLEY *et al.*, 2010; POKAHR e BRAUBACH, 2013), inclusive para soluções educacionais (MIKIC-FONTE *et al.*, 2010; VALDERRAMA *et al.*, 2013).

A escolha pelo framework Jadex se deu principalmente pela compatibilidade desta ferramenta com tecnologias *web* (como serviços) e dispositivos móveis (POKAHR e BRAUBACH, 2013), favorecendo a

interoperabilidade entre sistemas, e potencializando a utilização do modelo em ambientes educacionais ubíquos.

A implementação da plataforma Jadex é baseada na arquitetura PRS, com uma pequena variação, permitindo a definição de objetivos e de respectivas condições em que estes podem ser elencados durante o processo de deliberação (DIBLEY *et al.*, 2011). Embora a aderência aos aspectos formais da abordagem BDI, com relação ao raciocínio sobre as crenças, não seja tão desenvolvida no Jadex em comparação a outras plataformas tais como o Jason (HÜBNER *et al.*, 2004), existe a liberdade do programador realizar a implementação desta característica segundo a estratégia que desejar.

Existe também a possibilidade de se manter uma clara correspondência entre o projeto e a implementação do sistema através do mecanismo de declaração de objetivos (*goals*) e deliberação explícita do Jadex. Este mecanismo do Jadex permite que sejam declarados explicitamente através de anotações (*annotations*) quais os elementos Java (classes, atributos e métodos) representam crenças, objetivos e planos relacionados ao processo de deliberação intrínseco do framework.

Desta forma, pode-se definir explicitamente nas anotações: (1) quais crenças devem ser monitoradas, e com que frequência; (2) quais crenças estarão associadas à criação, à manutenção, à interrupção ou ao abandono de um objetivo; bem como (3) qual planos estão relacionados a um determinado objetivo.

Além disso, a definição prévia dos planos que poderão ser executados pelos agentes, com suas respectivas pré-condições e objetivos correspondentes, permite que o processo de raciocínio apresente melhor desempenho e eficiência do ponto de vista computacional em comparação com outras plataformas BDI, como o Jason (POKAHR *et al.*, 2010; FRANTZ *et al.*, 2012; KRAVARI e BASSILIADES, 2015).

Outra vantagem de se utilizar o Jadex está relacionada à afirmação de Dibley *et al.* (2011) sobre a facilidade de implementação pelo programador propiciada pelo framework. No caso do autor desta dissertação, isto foi verificado também pelo fato de (na versão atual do Jadex) os agentes serem construídos com tecnologia Java pura (POKAHR *et al.*, 2014). Desta forma, a tarefa de implementação do modelo foi simplificada pela *expertise* na referida linguagem de programação, adquirida durante sua vida profissional como desenvolvedor de software.

Quanto à escolha do LMS a ser utilizado na implementação do modelo, a preferência natural foi pelo Moodle, devido à ampla aceitação

desta plataforma na comunidade nacional e internacional, inclusive no mundo acadêmico (KUMAR *et al.*, 2011; DO PRADO *et al.*, 2014).

Outra motivação em se optar pelo Moodle se deve à familiaridade do autor desta dissertação com a ferramenta, inclusive já tendo desenvolvido profissionalmente personalizações e extensões (*plug-ins*) para uma instância do ambiente em uma instituição pública de âmbito estadual. Assim, a tarefa de desenvolver a integração entre o sistema multiagente e o LMS se tornou menos complexa.

Além disso, existem diversos trabalhos no campo da informática na educação que desenvolveram sistemas que se integram ao Moodle, expandindo as funcionalidades padrão da ferramenta (KOMLENOV *et al.*, 2010; ALENCAR e NETTO, 2011; DESPOTOVIC *et al.*, 2012; PALOMINO, 2013; RODRIGUES *et al.*, 2013; BREMGARTNER *et al.*, 2014; FRADE *et al.*, 2014; DO PRADO *et al.*, 2014).

Quanto ao padrão de metadados dos objetos de aprendizagem, optou-se pelo IEEE-LOM pelo fato já mencionado no Capítulo 2 (“Fundamentação Teórica”) de este ser um dos padrões de metadados de recursos educacionais mais aceitos e utilizados mundialmente em instituições de ensino, existindo vários objetos já catalogados segundo este padrão (VICARI *et al.*, 2010).

Por fim, justifica-se a utilização do modelo de integração de objetos de aprendizagem SCORM. O padrão SCORM define o protocolo de comunicação entre o LMS e o objeto de aprendizagem. Por meio deste protocolo, o estudante interage com o objeto através do LMS. Desse modo, o aluno visualiza o recurso instrucional e controla o sequenciamento de itens que fazem parte do pacote SCORM.

Além das características mencionadas no Capítulo 2 (“Fundamentação Teórica”), ressalta-se que o modelo SCORM é bastante utilizado em cursos on-line de instituições públicas nacionais (como os Tribunais de Justiça estaduais), e que existem vários recursos empacotados no padrão SCORM já produzidos e com os respectivos metadados preenchidos segundo o padrão IEEE-LOM (principalmente os elementos mais básicos).

Destaca-se também que a convergência de objetos de aprendizagem no padrão SCORM com agentes, permite que seja obtido um comportamento proativo do OA, o que não existiria na abordagem SCORM tradicional. Esta pró-atividade possibilita a adaptação da sessão de aprendizagem de maneira dinâmica, enriquecendo a abordagem SCORM tradicional com a capacidade de deliberação dos agentes.

Desta forma, durante a interação do aluno com o objeto, elementos do modelo de dados SCORM são obtidos pelo agente e utilizados no

processo de raciocínio, tais como: preferências do aluno, interações, objetivos de aprendizagem e pontuação em questionário (IEEE, 2005).

Com relação ao protótipo desenvolvido para validar o modelo proposto, integrado ao LMS Moodle, foram utilizados objetos SCORM existentes em cursos on-line oferecidos pela Academia Judicial do Tribunal de Justiça de Santa Catarina (TJSC).

A Academia Judicial é uma instituição que compõe o Poder Judiciário do estado de Santa Catarina, sendo responsável pela capacitação de servidores e membros da sociedade em geral, tanto através de cursos presenciais quanto de cursos on-line.

Os objetos SCORM cedidos (arquivos zip) são parte de um curso introdutório de Direito Previdenciário. Estes materiais contêm animações, sequenciamento de telas e questionários associados aos tópicos do curso supracitado. Ressalta-se que os referidos objetos foram cedidos com restrição de uso, apenas para o desenvolvimento deste projeto.

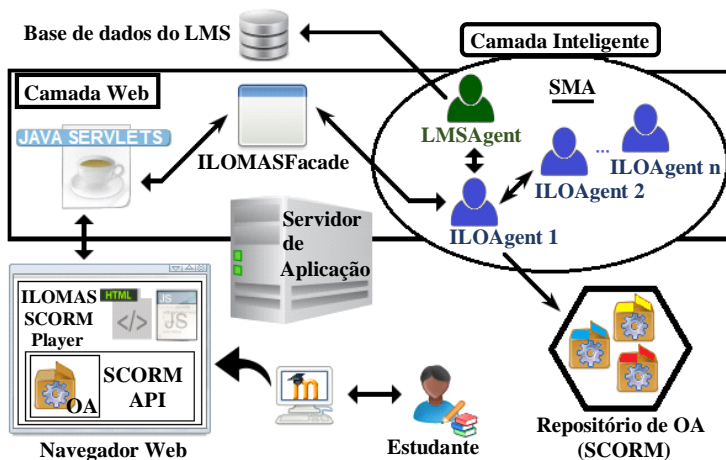
Na sequência do capítulo serão descritos os detalhes da implementação, com relação à arquitetura do sistema e a plataforma de agentes adotada, à integração com a tecnologia SCORM e ao modelo de dados utilizado no processo de deliberação do objeto inteligente de aprendizagem (OIA).

## 5.2 ARQUITETURA DO SISTEMA

O projeto da arquitetura do ILOMAS levou em consideração as características de cursos *e-learning* disponibilizados em um LMS (como o Moodle), no que se refere à forma de acesso a estes ambientes, que geralmente é feita via *web browsers* (navegadores). Como resultado, a infraestrutura da plataforma ILOMAS foi modelada levando em consideração requisitos de comunicação entre navegadores (clientes) e servidores *web*.

A Figura 27 exhibe o projeto arquitetural do sistema ILOMAS, com suas camadas *web* e inteligente (SMA).

Figura 27. Arquitetura do sistema



Fonte: Autor (2015)

Diante desse cenário, optou-se pela implementação do sistema ILOMAS através da plataforma Jadex (conforme já mencionado), pois este framework possibilita o desenvolvimento de aplicações orientadas a agentes racionais (BDI) compatíveis com tecnologias *web* e móveis. A versão mais atual do Jadex apresenta o *Kernel* (núcleo) BDI em sua terceira versão, o qual modificou a forma de desenvolvimento dos agentes, em comparação com a maneira utilizada na versão antiga.

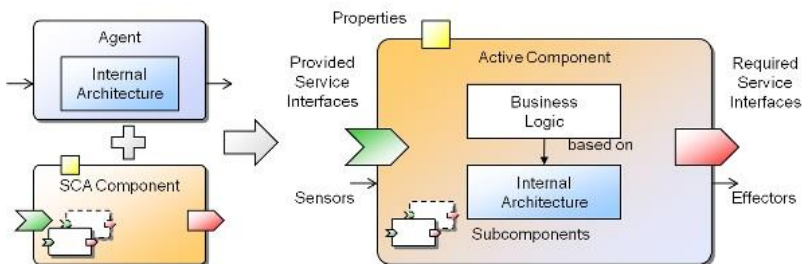
Anteriormente, os agentes eram desenvolvidos através de arquivos XML, os quais continham *tags* que definiam as características dos agentes (crenças, desejos, etc.), e definindo os planos (execução dos objetivos) através de classes Java. Com o Jadex BDI V3, o desenvolvimento de agentes é feito totalmente através de classes Java (não mais por arquivos XML), fazendo uso de anotações (*annotations*) para definir as crenças, os objetivos, as *capabilities* e os planos dos agentes.

Outra mudança trazida pela nova versão do Jadex é a possibilidade de se desenvolver componentes ativos (*Active Components*). Esta abordagem permite a associação de duas tecnologias: Agentes e SCA (modelo para composição de aplicações seguindo princípios da

arquitetura orientada a serviços - SOA). O modelo SCA foi proposto por algumas empresas como IBM, ORACLE, SAP, Siemens, etc., com o objetivo de promover e facilitar a interoperabilidade entre aplicações.

A Figura 28 ilustra a arquitetura de componentes ativos (*Active Components*) utilizada na plataforma Jadex.

Figura 28 – Arquitetura de Componentes Ativos



Fonte: Pokahr e Braubach (2013)

Com isso, a forma típica de comunicação entre os agentes Jadex BDI V3 se dá através do acesso a serviços. Os agentes BDI podem ser provedores de serviços, bastando implementarem as respectivas interfaces dos serviços e utilizarem as anotações (*annotations*) associadas. Podem também ser consumidores de serviços de outros agentes, de outros componentes padrão (não inteligentes), de *web services*, ou outros sistemas legados implementados nas mais diversas tecnologias, desde que estes tenham sido adaptados (*wrapper*) para responderem como agentes.

Conforme a arquitetura da implementação do sistema ILOMAS (ilustrada na Figura 27), a interface de interação entre o ambiente de agentes e o ambiente virtual tem como alicerce as tecnologias Java Servlets e JSP, beneficiando-se da estrutura de comunicação de serviços fornecida pelo Jadex BDI V3. A tecnologia de Servlets permite a execução de serviços e classes Java no servidor *web*, a partir de requisições dos navegadores clientes.

Outro benefício trazido pela característica de integração com tecnologias *web* apresentada pelo Jadex está relacionada a utilização do padrão SCORM como modelo de empacotamento e sequenciamento de objetos de aprendizagem no sistema ILOMAS.

A comunicação entre o LMS e o OA, definida pelo padrão SCORM (via ambiente de *Run-Time*, RTE), ocorre através de funções JavaScript,



cuja linguagem é a tecnologia mais utilizada mundialmente para desenvolvimento de programação na camada cliente (*Browser*) na *web*, adicionando comportamento dinâmico (ex.: AJAX) a conteúdos estáticos (ex.: arquivos HTML) (MORRISON, 2008).

Dentre as funções disponíveis na API SCORM, as que são mais importantes para este trabalho são `LMSSetValue` e `LMSGetValue`. A primeira função permite que o conteúdo SCORM envie informações à base de dados do LMS. A segunda função possibilita que o conteúdo SCORM obtenha informações do LMS.

Para implementar a interface entre o motor SCORM e o ambiente ILOMAS foi necessário desenvolver um player SCORM específico (baseado em arquivos HTML e JavaScript), compondo a camada *web* do sistema ILOMAS. Esta estratégia permite que se execute o processamento necessário entre o código SCORM e a camada inteligente (agentes).

O arquivo `LMSSCORMILOMASFunctions.js` sobrescreve a API SCORM, adicionando o código para envio dos dados da interação do objeto SCORM com o LMS Moodle ao ambiente ILOMAS (na função `LMSSetValue`), através de uma requisição AJAX.

Como resultado desta requisição, o sistema ILOMAS retorna o resultado da deliberação do agente `ILOAgent` à camada cliente. Caso seja identificada a necessidade de exibição de um novo OA ao estudante, o player SCORM desenvolvido irá receber a informação sobre qual o caminho (endereço) para o novo objeto.

A camada servlet delega o tratamento da requisição do cliente *web* (*browser*) para a classe (não-agente) `ILOMASFacade`, a qual segue o padrão de projeto Fachada, de acordo com as melhores práticas da Engenharia de Software. Este padrão de projeto fornece uma interface unificada e simplificada para um subsistema, promovendo o baixo acoplamento (GAMMA *et al.*, 1995).

Desta forma, a classe `ILOMASFacade` oferece acesso aos serviços dos agentes, mantendo a separação entre a camada SMA e os itens externos (servlets e *front-end*), evitando acoplamento desnecessário, e permitindo que possam ser feitas mudanças em uma camada sem que se precise alterar todo o sistema.

O trecho de código descrito na sequência demonstra algumas funções da API SCORM que foram sobrescritas através do arquivo `LMSSCORMILOMASFunctions.js` permitindo a integração com a camada SMA do ILOMAS.

```
function SCORM_ILOMAS_API() {
    function dispatchLMSGetValueServlet(varname){
```

```

statusBarSCORMElement = document.getElementById("iloirstatusbar");
var text = "";
text = text.replace(/\\s/g, '&nbsp;');
statusBarSCORMElement.innerHTML = text;
var xhttpLMSGetValue;
if (window.XMLHttpRequest){xhttpLMSGetValue = new XMLHttpRequest();}
else{xhttpLMSGetValue = new ActiveXObject("Microsoft.XMLHTTP");}
var appcontextlocationInput =
    document.getElementById("appcontextlocation");
if(appcontextlocationInput == null ||
    appcontextlocationInput.value == null)
{
    statusBarSCORMElement.innerHTML = "Can't get the Application"+
        " Location";
}
else {
    var appcontextlocationValue = appcontextlocationInput.value;
    var iloirIdInput = document.getElementById("iloirid");
    if(iloirIdInput == null || iloirIdInput.value == null) {
        statusBarSCORMElement.innerHTML = "Can't get the Intelligent"+
            " Learning Object Interface Resource Location";
    }else{
        var stdIdInput = document.getElementById("stdid");
        if(stdIdInput == null || stdIdInput.value == null) {
            statusBarSCORMElement.innerHTML = "Can't get the Student ID"+
                " Location";
        }else{
            var parameters = "varname="+varname+"&iloirid="+
                iloirIdInput.value+"&stdid="+
                stdIdInput.value;
            xhttpLMSGetValue.open("POST", appcontextlocationValue +
                "/GetLMSElementServlet", false);
            xhttpLMSGetValue.setRequestHeader("Content-type",
                "application/x-www-form-urlencoded; charset=UTF-8");
            xhttpLMSGetValue.send(parameters);
            if(debug){
                alert("dispatchLMSGetValueServlet status: " +
                    xhttpLMSGetValue.status);
                alert("dispatchLMSGetValueServlet response: " +
                    xhttpLMSGetValue.responseText);
            }
            if (xhttpLMSGetValue.status === 200) {
                return xhttpLMSGetValue.responseText;
            }
        }
    }
}
return "";
}

function LMSGetValue(varname) {
    ...

    errorCode = '0';
    var lmsReturnedValue = dispatchLMSGetValueServlet(varname);
    if(lmsReturnedValue && lmsReturnedValue != "" &&
        lmsReturnedValue != "-1")
    {
        cache[varname] = lmsReturnedValue;
    }
}

```

```

    if (debug) {
        alert("LMSSCORMILOMASFunctions LMSGetValue varvalue: " +
            cache[varname]);
    }

    return cache[varname];
}

function LMSSetValue(varname,varvalue) {
    ...

    cache[varname] = varvalue;
    if (debug) {
        alert("SCORMILOMASFunctions LMSSetValue " + varname + " cache: " +
            cache[varname]);
    }

    if(varname != "cmi.core.exit" && varname != "cmi.core.session_time")
    {
        if(document.readyState == "complete" &&
            (document.getElementById("ilomaswebplayer").
                contentWindow.document != null &&
                document.getElementById("ilomaswebplayer").
                    contentWindow.document.readyState == "complete"))
        {
            dispatchLMSSetValueServlet(varname, cache[varname]);
        }
    }

    errorCode = '0';
    return "true";
}
...
}

```

A implementação da camada multiagente seguiu a estrutura de classes do framework Jadex BDI V3, cujos detalhes técnicos podem ser obtidos em Amorim Jr. *et al.* (2015). Nesta fase, foram declaradas as classes dos agentes, juntamente com os elementos BDI: crenças, objetivos (estados que se deseja alcançar e que guiam o comportamento do agente) e planos (forma concreta de atingir uma intenção) (WOOLDRIDGE, 2009).

Entre as anotações (*annotation*) disponibilizadas pelo Jadex para definir os elementos BDI, foram utilizadas (POKAHR *et al.*, 2014):

- *@Agent* – para declarar uma classe de agente, ou um atributo do tipo agente.
- *@Belief* – define que um atributo representa uma crença, e propriedades associadas, que permitem definir a frequência de atualização de uma crença, bem como monitorar esta atualização.

- *@Goal* – para declarar os objetivos de um agente. Este tipo de anotação permite que sejam definidas condições para disparar, manter, inibir, alcançar ou abandonar o objetivo, bem como a ordem de prioridade entre objetivos.
- *@Plan* – para definir que um método representa um plano, bem como as condições para disparo do plano (pré-condições, com base em objetivos ou crenças), condições de contexto (manutenção do plano) e ações de sucesso, abandono ou falha do plano.
- *@Capability* – definição de *capabilities* associadas ao agente. Conforme mencionado anteriormente nesta dissertação, uma *capability* possibilita o reuso de características BDI (crenças, objetivos e planos) comuns entre agentes distintos, de forma modular.

Quanto a representação das crenças, cabe destacar que são encontradas na literatura diversas estratégias para mapeamento de ontologias para crenças de agentes, ou para consulta de ontologias por agentes. Estes modelos possibilitam a representação dos elementos da ontologia em classes Java, ou em RDF e XML que podem ser interpretados por objetos Java, e geralmente são integrados a plataformas de desenvolvimento de agentes, como o JADE ou Jadex.

Algumas destas estratégias utilizam *plug-ins* da ferramenta Protégé para gerar as classes Java com base na ontologia, como o Bean Generator (BREMERTNER *et al.*, 2014) e o Jastor (DIBLEY *et al.*, 2011). Outras abordagens baseiam-se em bibliotecas que permitem a construção de objetos Java simples (Java Beans) com base na ontologia (TOMAIUOLO *et al.*, 2006; STEVENSON e DOBSON, 2011; OHLBACH, 2012).

Existem também outras alternativas como o uso da biblioteca OWL API para interagir e manipular ontologias (BECHHOFER *et al.*, 2003; HORRIDGE e BECHHOFER, 2011), ou outras bibliotecas como o Jena e OntoCAT (ADAMUSIAK *et al.*, 2011; DIBLEY *et al.*, 2011).

Por fim, destaca-se o trabalho de Campos (2014), o qual propõe um modelo para a representação de dados semânticos em agentes BDI, permitindo o armazenamento de informações, bem como a comunicação com outros agentes (semânticos e não-semânticos) e com *triplestores* da Web Semântica.

A pesquisa de Campos (2014) está sendo estendida por um outro integrante do grupo de pesquisa IATE, e futuramente poderá ser utilizada

pelo modelo ILOMAS para a integração das crenças dos agentes com a ontologia do modelo de dados e com ontologias de domínio educacional.

Devido a este cenário, e também por questões de limitação de escopo do protótipo, optou-se por utilizar a ontologia do Modelo Conceitual de Dados (apresentada no capítulo 4, “Modelo Proposto: ILOMAS”) basicamente como uma terminologia para definição dos elementos constituintes das crenças do ILOAgent (agente cognitivo).

Logo, ressalta-se que o protótipo implementado não está integrado a nenhum processo de raciocínio (inferência) sobre ontologias. Conforme será mencionado no último capítulo desta dissertação (Capítulo 6, “Considerações Finais”), a implementação da utilização de inferências sobre ontologias no ILOMAS poderá ser desenvolvida como um trabalho futuro.

Desta forma, para a implementação do protótipo, foi realizado o mapeamento direto dos elementos da ontologia supracitada para objetos Java (em uma estrutura de pacote - *package*), os quais representam os conceitos desta ontologia, com suas propriedades e suas relações (TBox). Este mapeamento foi realizado com base na funcionalidade de exportação da ontologia para código Java presente na ferramenta Protégé, e realizando algumas adequações manualmente.

Assim, a estrutura das crenças do ILOAgent foi desenvolvida a partir da ontologia mencionada, convertida para classes Java conforme o modelo do framework Jadex, e incorporando alguns elementos em coleções (*collections*) e outras estruturas de dados padrão da linguagem Java para facilitar a implementação.

Com isso, os elementos do modelo de dados foram incluídos na representação de conhecimento interna do ILOAgent como atributos da respectiva classe do agente, e estes atributos foram marcados conforme a sintaxe de declaração de crenças no framework Jadex (utilizando a *annotation @Belief* para definir que um atributo é uma crença).

Além disso, as crenças foram associadas a critérios de disparo de objetivos, através da propriedade *beliefs* da anotação *@GoalCreationCondition*, e também foram utilizadas como condições para determinar o fluxo de execução de planos (regras se-então). Desta forma, por razões de simplificação de implementação, cabe ressaltar que não foram definidos critérios mais elaborados para a escolha de objetivos e/ou planos.

O trecho de código-fonte transcrito na sequência descreve parte da implementação do ILOAgent, segundo a sintaxe do framework Jadex. O intuito desta reprodução não é descrever minuciosamente todo o arquivo fonte do agente, mas sim demonstrar a forma de programação de agentes

BDI segundo a plataforma Jadex, bem como permitir que se tenha uma noção mais concreta da estrutura do código do agente que foi desenvolvida.

```

@Agent
@Service
@ProvidedServices(
    {@ProvidedService(type=ILearningObjectPlayerService.class)})
public class ILOAgentBDI
    implements ILearningObjectPlayerService
{
    @Agent
    public BDIAgent agent;

    @Belief
    protected String currentLessonStatus;

    @Belief
    protected int requestForNewLOReceived;

    @Belief
    protected ObjetoDeAprendizagem learningObject;

    @Belief
    Protected DesempenhoInteracao currentAttempt;

    @Belief
    protected Map<LOMLearningObject, Map<Integer, HistoricoOA>>
        learningObjectsPerformanceHistory;

    ...

    @Goal(unique=true,
    deliberation=@Deliberation(inhibits={GoalRequestForNewLO.class}))
    public class GoalRequestForNewLO{
        @GoalCreationCondition(beliefs={"requestForNewLOReceived"})
        public GoalRequestForNewLO() {System.out.println("GRFNLO created");}
    }

    @Plan(trigger=@Trigger(goals=GoalRequestForNewLO.class))
    private void obtainNewLO() {
        System.out.println("Obtaining a new LO on ILOAgent " + this);
        if(this.learningObject!=null &&
            (this.currentAttempt.getGrade() > 0 ||
            this.currentAttempt.getScoreRaw() >= 0))
        {
            defineLOGrade();
        }

        // Search for a LO
        this.learningObject = searchLO();

        if(!this.hasPreviousError &&
            this.iLOInterfaceResource != null && this.learningObject != null)
        {
            ILOMASPlatform.getInstance().inform(LMSAgentBDI.class,
                ILOMASPlatform.INFORM_LO_SUBJECT_TO_STUDENT,
                this.iLOInterfaceResource.getRequester(),
                this.iLOInterfaceResource.getSubject(), this.learningObject);
        }
    }
}

```

```

this.learningObjectsPresented.add(this.learningObject);
Map<Integer, DesempenhoInteracao > learningObjectPerformanceList =
    this.learningObjectsPerformanceHistory.get(this.learningObject);
DesempenhoInteracao learningObjectPerformance =
    new DesempenhoInteracao (this.learningObjectsPresented.size() - 1,
        this.learningObject, null);
if(learningObjectPerformanceList != null){
    learningObjectPerformanceList.put (
        this.learningObjectsPresented.size() - 1,
        learningObjectPerformance);
}
else {
    learningObjectPerformanceList =
        new HashMap<Integer, DesempenhoInteracao>();
    learningObjectPerformanceList.put (
        this.learningObjectsPresented.size() - 1,
        learningObjectPerformance);
    this.learningObjectsPerformanceHistory.put (this.learningObject,
        learningObjectPerformanceList);
}

System.out.println(""+this.agent.getAgentName()+" Selected "+
    "Learning Object: " + this.learningObject);
this.hasPreviousError = false;
ILOMASPlatform.getInstance().addCreatedLOMLO(""+
    this.learningObject.getId(), this.learningObject);
System.out.println("The LO was obtained on ILOAgent " + this);
}

...
}

```

### 5.3 PROCESSO DE DELIBERAÇÃO DO OIA

Conforme mencionado anteriormente, dois tipos de agentes foram desenvolvidos: o agente responsável pelo acesso à base de dados do LMS (LMSAgent) e o agente que representa o OIA (ILOAgent). O agente LMSAgent fica responsável por acessar os dados armazenados nas tabelas do LMS Moodle que também são representados nas crenças do agente ILOAgent.

O agente ILOAgent representa um OIA (Objeto Inteligente de Aprendizagem), e seus principais objetivos são a seleção e oferta de um OA ao estudante, e o monitoramento da interação deste aluno com o objeto, para verificar a necessidade (ou não) de oferta de objeto(s) que complemente(m) a experiência de aprendizagem.

Os elementos do modelo de dados do ILOAgent são a base para definir o fluxo de comportamento do agente, e como consequência, a dinâmica do processo de seleção de objetos de aprendizagem ao aluno.

Todos estes elementos são as crenças do ILOAgent utilizadas no processo de adaptação do conteúdo do OA, de acordo com o desempenho do aluno.

As informações referentes ao LMS (Modelo do LMS) são compostas pela nota do estudante em um curso no Moodle e pelo assunto atual do curso que o aluno deve estudar. Conforme mencionado anteriormente, inicialmente o ILOAgent obtém estas informações a partir do LMSAgent.

O assunto que o aluno precisa aprender em uma determinada experiência de aprendizagem (uma das crenças do agente ILOAgent) pode ser denominado como objetivo de aprendizagem. Esta informação é configurada no LMS pelo professor. Entretanto, cabe destacar que o educador não precisa definir na estrutura do curso dentro do LMS qual (ou quais) objeto de aprendizagem estará disponível ao estudante.

Com relação ao Modelo do OA (objeto de aprendizagem) representado na base de crença do ILOAgent, destaca-se que este modelo é composto pelos metadados do OA, segundo o padrão LOM (IEEE Std 1484.12.1, 2002). Estes elementos de metadados do IEEE-LOM são utilizados de modo a permitir que se identifique qual é o tópico educacional relacionado ao objeto de aprendizagem, qual o nível de dificuldade do objeto, qual o tipo do recurso educacional e qual seu nível de interatividade, e também qual objetivo educacional e pré-requisitos associados.

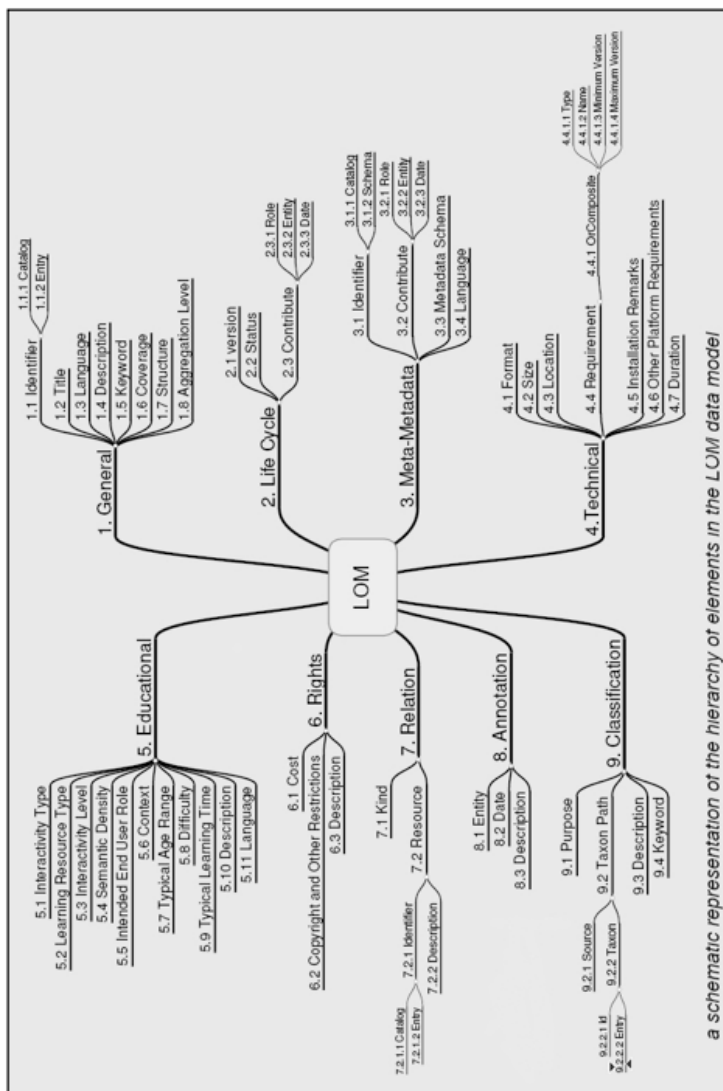
Assim, um objeto inteligente de aprendizagem (ILOAgent) utiliza as informações (elementos de metadados) que possui sobre o objeto atual para realizar a busca por objetos complementares no repositório de objetos de aprendizagem, caso o desempenho do estudante não estiver sendo satisfatório, potencializando o entendimento deste aluno com relação ao assunto em questão.

Além disso, conforme já mencionado nesta dissertação, os elementos de metadados (Modelo do OA) podem ser utilizados em conjunto com ontologias de domínio e de especificação dos metadados LOM para enriquecer a busca por objetos de aprendizagem em repositórios e em bases de dados ligados (*linked data*).



A Figura 29 exibe uma representação semântica da hierarquia dos elementos que compõem o padrão de metadados IEEE-LOM.

Figura 29. Elementos do Padrão de Metadados IEEE-LOM



A fonte de objetos de aprendizagem será o repositório de objetos local que o ambiente ILOMAS pode acessar, e apenas os objetos que estiverem associados ao tema definido pelo professor é que serão aptos a serem selecionados em uma determinada experiência de aprendizagem (no contexto desta dissertação, também denominada sessão de aprendizagem).

Precisamente, a busca por um OA no repositório de objetos de aprendizagem que foi implementada leva em consideração os seguintes elementos, os quais são declarados como crenças do ILOAgent:

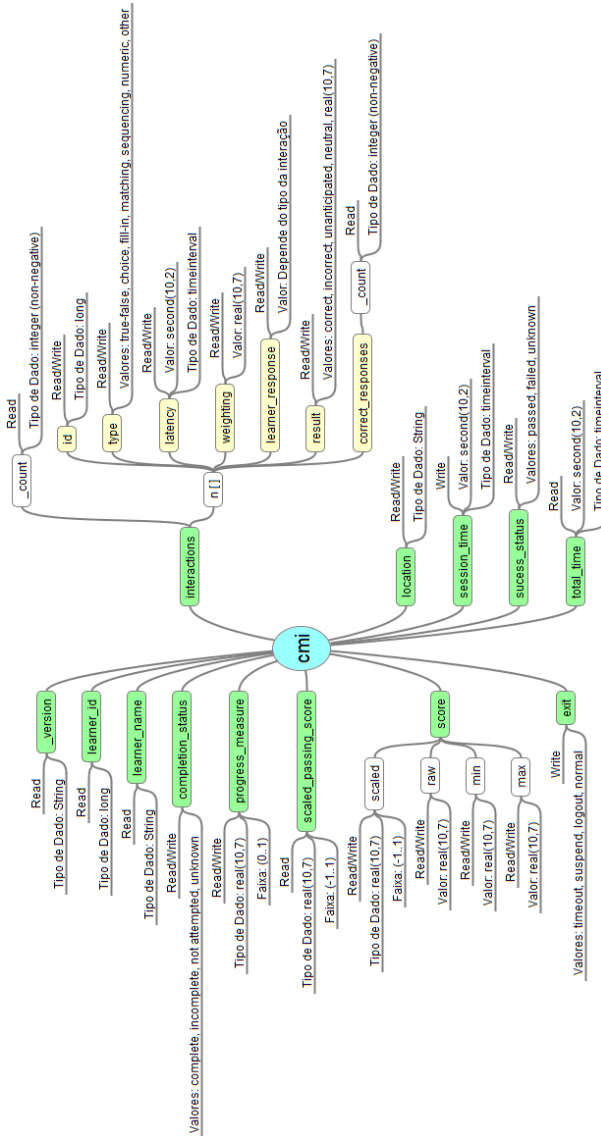
- O assunto do curso que o aluno precisa aprender no momento. Este dado é obtido através da troca de mensagem com o LMSAgent, o qual recupera a informação da base de dados do LMS Moodle.
- Os elementos de metadados do padrão LOM do OA que está sendo ofertado (ex.: palavras-chave, grau de dificuldade, pré-requisitos e formato).
- O histórico de objetos de aprendizagem já apresentados ao aluno, com o respectivo desempenho que o estudante obteve.

Além disso, outra crença importante deste tipo de agente é o histórico dos elementos da especificação SCORM (e seus respectivos valores) recebidos durante a interação com o estudante. Estes dados compõem a implementação do Modelo de Interação (Aluno-OA), juntamente com os elementos do modelo de dados SCORM recebidos na interação com o objeto de aprendizagem atual.

Cabe citar que os elementos do modelo de dados SCORM são definidos segundo o documento IEEE Std 1484.11.1 (IEEE, 2005).

A Figura 30 ilustra os elementos do modelo de dados SCORM.

Figura 30. Modelo de Dados SCORM



Fonte: Autor (2015).

Estas informações possibilitam que as características do aluno (Modelo de Interação do Aluno com o Objeto) sejam modeladas com base nos elementos recebidos durante a interação do estudante com o OA (sessão de aprendizagem), através da comunicação com a API SCORM.

Potencialmente, o agente ILOAgent é capaz de reconhecer e interpretar os valores de todos os elementos disponíveis no modelo de dados do padrão SCORM. Entretanto por questões de limitação de escopo, no protótipo foram implementados os seguintes elementos:

- O resultado de um questionário disponibilizado em uma interação (elemento **cmi.interactions.n.result**), cujos possíveis valores podem ser “correto”, “incorreto” ou um número real.
- O tempo decorrido (latência) desde o início da interação atual (elementos **cmi.interactions.n.latency** e **cmi.session\_time**).
- O peso do resultado (desempenho) obtido em uma interação com relação as outras interações (**elemento cmi.interactions.n.weighting**).
- A situação de conclusão de um item que compõe o objeto e qual a pontuação obtida pelo aluno (elementos **cmi.completion\_status**, **cmi.success\_status** e **cmi.score**).
- A medida de progresso no objeto atual (elemento **cmi.progress\_measure**).

Cabe ser feito o destaque sobre o critério de escolha dos elementos que compõem o subconjunto do modelo de dados SCORM implementado no protótipo. Os elementos do modelo SCORM relacionados a desempenho e tempo (duração), características estas inerentes ao problema, foram selecionados para compor o subconjunto.

Com as creanças especificadas através dos elementos (subconjunto) do modelo de dados SCORM, o ILOAgent consegue identificar:

- Se o aluno está demorando demais na interação com o objeto, ou se está avançando na sequência de recursos que compõem o OA.
- Se o estudante está respondendo corretamente as questões (questionários inseridos no OA).
- Se o aluno obteve resultado satisfatório na interação com o OA (acessou todos os conteúdos necessários, e/ou obteve nota mínima requerida no questionário).

- Se o estudante está recebendo objetos de aprendizagem muito aquém do seu nível de conhecimento (desempenho satisfatório com avanço muito rápido).
- Se o aluno está conseguindo suprir sua deficiência no(s) assunto(s) relacionado(s) ao OA, caso um (ou mais) OA(s) complementar(es) tenha(m) sido exibido(s) ao estudante durante a sessão de aprendizagem (reforço sobre o mesmo tema, ou sobre assunto pré-requisito).

Desta forma, os dados recebidos pelo ILOAgent (via camada *web*, durante a interação do aluno com o objeto SCORM) são utilizados no seu processo de raciocínio, como critério de escolha ou abandono dos seus objetivos, possibilitando a identificação da necessidade de reforço com maior precisão e rapidez.

Como já havia sido mencionado anteriormente, o protótipo foi implementado segundo a plataforma Jadex. Diante disso, cabe ser feita uma explanação mais detalhada sobre o processo de deliberação do framework Jadex, do ponto de vista prático.

A etapa de deliberação de objetivos no Jadex é realizada segundo o motor de deliberação padrão do framework, o qual segue uma estratégia que seleciona objetivos ativos (cujas condições de disparo são verdadeiras) e livres de conflito (estratégia *Easy Deliberation*). Em seguida, o referido motor monta a lista de planos aplicáveis para atingir o(s) objetivo(s) elencados (DIBLEY *et al.*, 2011; POKAHR *et al.*, 2014).

Os planos candidatos são selecionados e instanciados para a execução, de tal forma que todos os planos que cumprirem as condições são executados. Através do atributo *priority* da anotação de planos (@Plan), é possível definir a prioridade de um plano em relação a outros associados ao mesmo objetivo. O processo finaliza quando todos os objetivos são alcançados (segundo critério definido em cada objetivo) ou quando não há mais planos associados.

Os atributos marcados como crença (*annotation @Belief*) no Jadex são monitorados pelo processo de deliberação da plataforma. Com isso, ao ocorrer uma mudança de valor no atributo que representa uma crença, o referido mecanismo de deliberação do framework realiza as etapas de verificação. Estas verificações constituem-se basicamente da avaliação aos critérios de disparo de objetivos e da seleção dos planos relacionados.

Por razões de simplificação do protótipo e limitação de escopo, cada objetivo implementado possui apenas um plano associado, para evitar conflitos, e evitando-se a necessidade de definição da prioridade

entre planos. A mesma estratégia foi adotada com relação aos objetivos, de tal forma que não foram definidos objetivos conflitantes.

Além disso, foram implementados planos para o tratamento de dois tipos de falhas técnicas e educacionais. O primeiro plano deste tipo adicionado ao ILOAgent foi o trataFalhaEducRepositoriosSemNovoOA.

Este plano trata a situação em que todos os objetos existentes no repositório, relacionados ao critério de busca atual, já foram ofertados ao aluno. O corpo do método associado ao plano (anotado com @Plan), define que o comportamento do ILOAgent nesta situação é ofertar ao aluno o OA inicial da sessão de aprendizagem (que já havia sido exibido).

O segundo plano implementado para o tratamento de falhas foi o trataErroTecnicoGenerico, cuja ação é apresentar uma mensagem ao aluno (caixa de diálogo), solicitando que acesse outra vez o OA (iniciando uma nova sessão de aprendizagem).

Futuramente, outras falhas contornáveis (cuja ação de tratamento possa ser elaborada em tempo de desenvolvimento), poderão ser definidas como planos do OIA. Já em situações onde não seja possível definir previamente um comportamento (código) para tratar a falha, o ILOAgent poderá utilizar o plano genérico. Assim, será exibida uma mensagem ao aluno, por exemplo, para que entre em contato com o professor.

Após a descrição do motor de deliberação da plataforma Jadex e dos planos desenvolvidos, é oportuno detalhar o processo de seleção dinâmica de objetos implementado no protótipo, com relação aos critérios de disparo do objetivo relacionado à oferta de um novo OA.

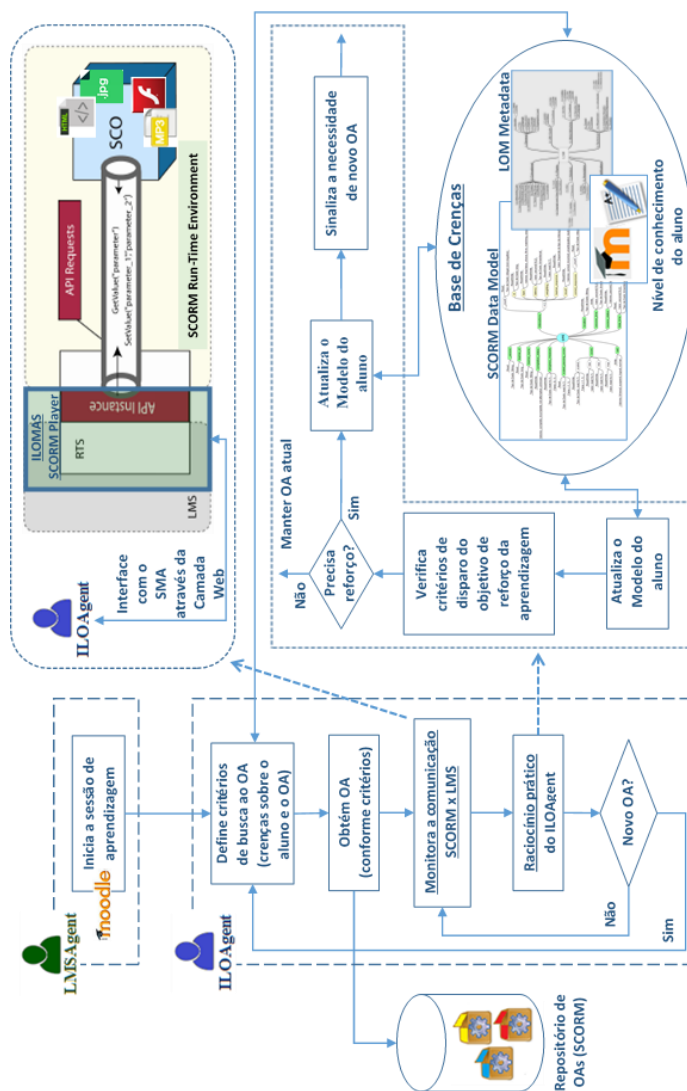
O objetivo que representa a necessidade de complementação (reforço) da aprendizagem pode ser disparado de acordo com alguns critérios, com base nas crenças sobre a interação do aluno com o OA, tais como: o estudante apresentar três erros em sequência em um questionário composto por cinco questões, ou ficar mais do que um determinado intervalo de tempo (ex.: dez minutos) sem interagir com o objeto.

Assim, não é necessário aguardar até o término da sessão de aprendizagem (saída do aluno do OA) para identificar que o estudante não está tendo um desempenho satisfatório na avaliação (respostas incorretas em sequência), ou que está levando muito tempo em avançar na navegação do objeto de aprendizagem.

Logo que a etapa de raciocínio do ILOAgent identifique a necessidade de exibir um novo OA ao aluno, a interação atual é interrompida e um novo ciclo de busca por objetos recomeça, porém de tal forma que este processo seja imperceptível ao estudante. O aluno nota apenas a alteração do conteúdo, mas na mesma interface do objeto de aprendizagem, sem mudanças abruptas de tela.

A Figura 31 descreve o fluxo de execução do sistema ILOMAS, o qual será detalhado na sequência.

Figura 31. ILOMAS – Fluxo de Execução



Fonte: Autor (2015).

A sessão de aprendizagem é iniciada pelo LMSAgent, através da interface entre o player SCORM no LMS e a camada web do ILOMAS. O LMSAgent obtém as informações iniciais da experiência de aprendizagem (sessão), que são a nota atual do estudante no curso (nível de desempenho) e o tópico educacional que deve ser estudado pelo aluno nesta sessão de aprendizagem.

Estas informações são repassadas ao ILOAgent, para que este agente busque um objeto de aprendizagem no repositório de objetos, com base nos critérios de seleção dos objetos disponíveis. Tais critérios são as crenças do ILOAgent que representam a nota do aluno no curso e o respectivo assunto (tópico) que o estudante deve aprender, ambas as informações recebidas do LMSAgent.

Na sequência, o ILOAgent seleciona no repositório um objeto que apresente em seus metadados o nível de dificuldade (elemento *difficult* do IEEE-LOM) compatível com o nível de desempenho do aluno no curso (nota), cujo assunto associado à sessão de aprendizagem esteja definido nos elementos palavras-chave (*keywords*) e título (*title*) do OA (segundo o padrão IEEE-LOM).

Conforme previamente mencionado no capítulo anterior (Capítulo 4, “Modelo Proposto: ILOMAS”), ressalta-se que no protótipo implementado não foi realizada integração com ontologias de domínio ou de modelo de dados de padrões de referência de objetos de aprendizagem, para expandir o processo de busca por OA no repositório.

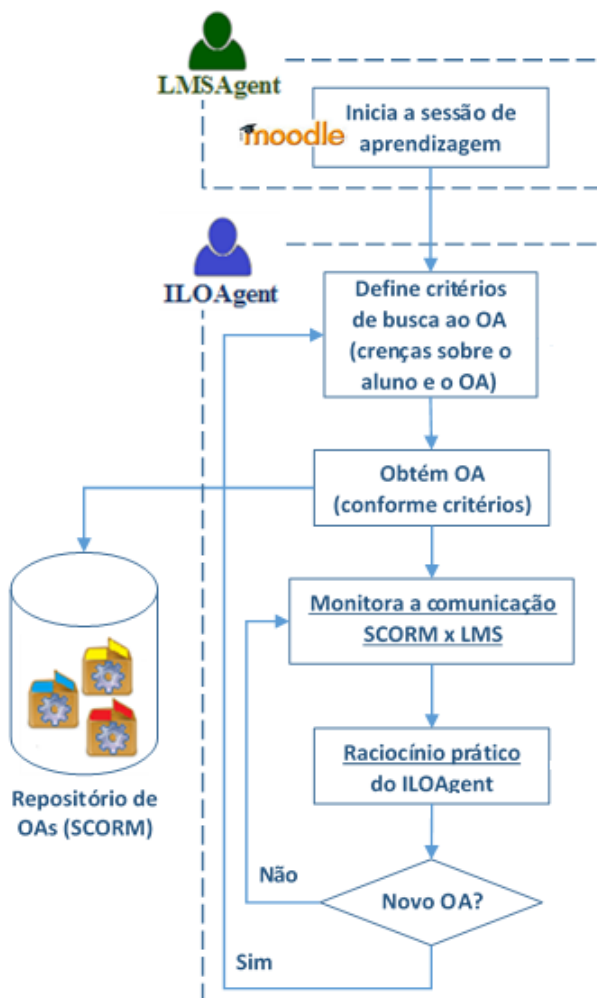
Desta forma, por questões de simplificação de implementação e restrição de escopo, o protótipo não apresenta esta característica de busca semântica por objetos de aprendizagem estabelecida no modelo teórico. Contudo, este aspecto poderá ser incorporado à implementação em trabalhos futuros.

Logo, durante o passo de obtenção de um OA no repositório implementado no protótipo, a busca é feita com base na correspondência



simples entre os termos (“*matching*” de *strings*). Esta etapa é ilustrada conforme Figura 32.

Figura 32. Início da sessão de aprendizagem e busca pelo OA

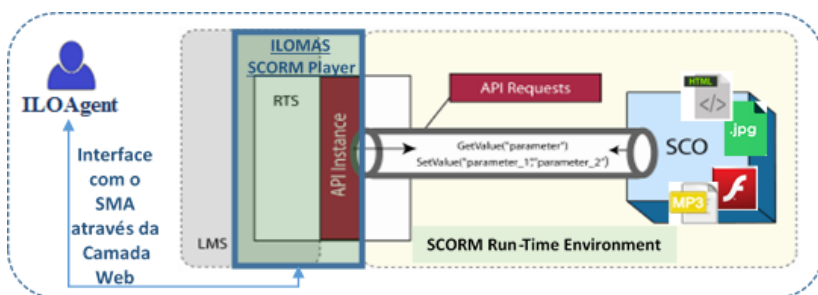


Fonte: Autor (2015).

Após o objeto de aprendizagem ser obtido pelo ILOAgent, é iniciada a exibição do recurso ao aluno no LMS, e o ILOAgent passa a

monitorar a interação entre o aluno e o OA, através da interface com o player SCORM implementado. A Figura 33 representa esta integração do ILOAgent com a API do ambiente de *Run-Time* do SCORM (SCORM RTE).

Figura 33. Interface ILOAgent e SCORM RTE



Fonte: Autor (2015).

Nesta etapa, a cada novo dado recebido da interface com o objeto SCORM são atualizadas as respectivas crenças do ILOAgent. Conforme mencionado anteriormente, estas informações são elementos do modelo de dados SCORM: o tempo da sessão de aprendizagem, o progresso de telas já visualizadas, o resultado obtido em uma questão de questionário, ou no questionário como um todo.

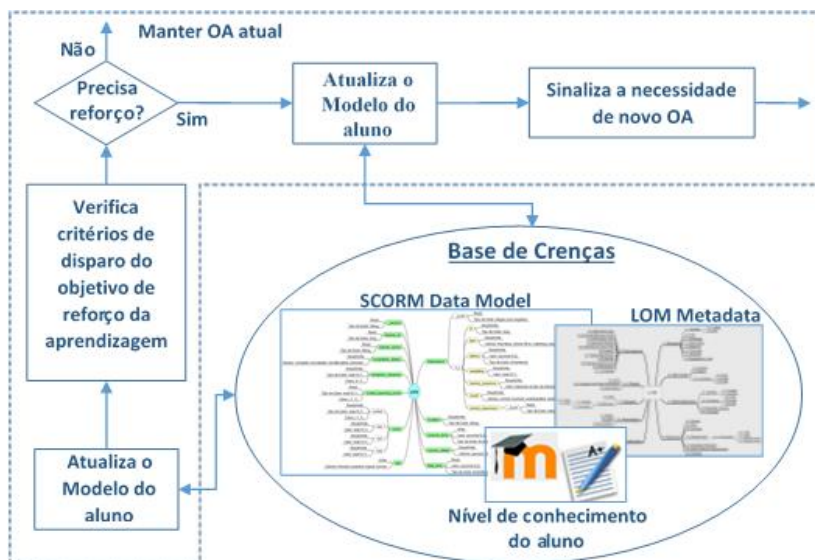
Conforme a implementação do ILOAgent na plataforma Jadex, sempre que o valor de uma das suas crenças é modificado, ocorre o processo de raciocínio deste agente. Como resultado, há a manutenção do objetivo corrente, que é continuar monitorando a interação do aluno com o objeto atual, caso os critérios de disparo do objetivo reforço de aprendizagem (*GoalRequestForNewLO*) não sejam satisfeitos.

Entretanto, se as condições para disparar o objetivo de reforço (complemento) da aprendizagem forem detectadas, é sinalizada a necessidade de se ofertar um novo objeto ao estudante, e se inicia um novo processo de seleção de objeto no repositório.

Destaca-se ainda, que tanto o histórico de objetos de aprendizagem oferecidos, quanto os desempenhos apresentados pelo estudante durante a sessão de aprendizagem, são armazenados na estrutura interna do ILOAgent (base de crença), para que estas informações sejam utilizadas

nos processos de raciocínio subsequentes, no decorrer da sessão de aprendizagem. A Figura 34 ilustra os passos da etapa de raciocínio.

Figura 34. Fluxo do processo de raciocínio do ILOAgent



Fonte: Autor (2015).

Com a descrição do processo de seleção de objetos de aprendizagem durante a interação do aluno com o recurso no LMS, é possível fazer a conexão entre o comportamento do sistema ILOMAS e as características de qualidade de softwares educacionais elencadas no capítulo inicial desta dissertação (Capítulo 1, “Introdução”): adaptatividade e reusabilidade.

Desta maneira, a adaptatividade do sistema está na possibilidade de modificar o fluxo de apresentação do OA ao estudante, dependendo do seu desempenho durante a experiência de aprendizagem, do histórico de interações e sugestões de objetos de aprendizagem a este indivíduo, e também de acordo com o seu grau de conhecimento (nota no curso), dependendo do intervalo de valores, podendo ser: “muito ruim”, “ruim”, “regular”, “bom” e “excelente”.

A forma como o sistema se adapta à característica do estudante é descrita na seqüência. Um aluno que apresente bom rendimento, no tema

que o professor definiu como objetivo de aprendizagem, terá uma experiência de aprendizagem concisa, recebendo apenas o OA necessário.

Por outro lado, o estudante que apresentar baixo desempenho no tema receberá mais de um OA, sendo do mesmo assunto definido como objetivo de aprendizagem pelo professor (reforço), ou então um OA de um assunto pré-requisito (caso o OA atual defina esta informação em seus metadados) para tentar suprir a lacuna de conhecimento do estudante.

Além disso, destaca-se também a possibilidade de se ofertar objetos de aprendizagem mais desafiadores para alunos que possuam um conhecimento mais avançado do assunto relacionado ao OA. Desta forma, caso o estudante esteja recebendo objetos muito básicos, e como resultado complete as experiências de aprendizagem muito rapidamente, o sistema pode oferecer objetos com maior nível de dificuldade, para evitar que o aluno fique entediado e por isso resolva desistir da aprendizagem.

A exibição de um novo OA para o estudante enquanto ocorre a interação (sessão de aprendizagem), conforme o desempenho apresentado pelo aluno, e sem que o professor tenha definido explicitamente na estrutura do curso que aquele OA era necessário, resulta em um comportamento dinâmico da experiência educacional, com uma adaptação no âmbito de objeto de aprendizagem (adaptação na granularidade OA).

Novos objetos de aprendizagem (associados ao tema definido no LMS) que forem adicionados ao repositório ficarão disponíveis automaticamente para os alunos, sem a intervenção do professor ou designer instrucional (DI) na configuração do curso, facilitando o reuso de recursos (potencializa a reusabilidade).

Com esta adaptação no âmbito de objeto, do ponto de vista do estudante (e mesmo do professor) o OA acessado é apenas um, porém com muitos conteúdos associados, caso seja identificada a necessidade de oferta de objetos extras ao aluno.

Este cenário representado por um objeto de aprendizagem composto de muitos conteúdos associados (um OA mais completo, composto por outros “sub-objetos”) é uma situação análoga à composição de blocos de montar, segundo a metáfora do brinquedo LEGO (DA SILVA *et al.*, 2010).

Por fim, cabe ser feito um destaque quanto à implementação do modelo proposto. Embora o protótipo produzido apresente um mapeamento reduzido de alguns elementos do SCORM (ex.: latência e resultado nos questionários), devido a questões de tempo, prazo e

limitação de escopo, ressalta-se que essa situação não invalida a verificação do modelo teórico.

#### 5.4 ILOIR: MOODLE *PLUG-IN*

Para integrar o ambiente ILOMAS ao LMS Moodle de forma facilitada ao aluno, sem que este precise sair do ambiente virtual e acessar outro sistema, evitando possíveis perdas de atenção (ARDITO *et al.* 2006; WINTER *et al.*, 2010), foi desenvolvida uma interface aos Objetos Inteligentes de Aprendizagem, denominado ILOIR (acrônimo em inglês: *Intelligent Learning Object's Interface Resource*).

O ILOIR é um *plug-in*, ou seja, uma extensão à versão padrão do Moodle (versão 2.8), e foi produzido segundo os padrões e convenções de desenvolvimento de novas funcionalidades para esta plataforma, conforme definido no site oficial da ferramenta: moodle.org.

A extensão desenvolvida acopla a camada *web* do sistema ILOMAS (composta por um player SCORM, um contêiner HTML e uma biblioteca JavaScript) à estrutura do LMS, permitindo que o objeto de aprendizagem (pacote SCORM) disponibilizado no Moodle estabeleça comunicação (indireta) com o ambiente dos agentes.

O ILOIR é o componente tecnológico (artefato) que permite disponibilizar as funcionalidades do sistema ILOMAS, e mais precisamente de um Objeto Inteligente de Aprendizagem, ao aluno dentro do LMS Moodle. É através do ILOIR que o estudante terá acesso ao objeto de aprendizagem e que a adaptação de conteúdo realizada pelo ILOAgent será percebida.

O ILOIR é disponibilizado no Moodle como um elemento do tipo recurso, que pode ser adicionado em um curso de ambiente virtual pelo professor (ou DI). Os itens associados ao *plug-in* que podem ser configurados são: (1) o assunto (tema) objetivo de aprendizagem daquela

unidade instrucional e (2) o endereço onde a plataforma ILOMAS está disponível.

A Figura 35 exibe a interface gráfica de configuração do *plug-in* ILOIR no LMS Moodle, com os respectivos dados que podem ser informados pelo professor ou *designer* instrucional.

Figura 35. Tela de Configuração do *Plug-in* ILOIR

**ILOIR** Atualizando um ILOIR em Tópico 1 ?

▼ Geral ► Expandir tudo

**Nome\*** ? Introdução ao Direito Previdenciário

**Descrição**

Objeto Inteligente de aprendizagem referente ao assunto da Unidade 3

**Assunto\*** ? Introdução ao Direito Previdenciário - Regimes Próprios

**URL da plataforma ILOMAS (Acesso aos Objetos de Aprendizagem)\*** ? http://posiate.inf.ufsc.br:8080/ILOMASWeb

► Nota ?

► Configurações comuns de módulos

Fonte: Autor (2015).

O trecho de código em PHP transcrito na sequência representa a tela de exibição do *plug-in* ILOIR quando do acesso pelo estudante:

```
require_once (dirname (dirname (dirname (__FILE__))) . '/config.php');
require_once (dirname (__FILE__) . '/lib.php');
require_once ($CFG->libdir . '/pagelib.php');
global $PAGE, $USER;
$id = optional_param ('id', 0, PARAM_INT);
$n = optional_param ('n', 0, PARAM_INT);

if ($n) {
    $iloir = $DB->get_record ('iloir',
        array ('id' => $n), '*', MUST_EXIST);
    $course = $DB->get_record ('course', array ('id' => $iloir->courseid),
        '*', MUST_EXIST);
    $cm = get_coursemodule_from_instance ('iloir', $iloir->id,
        $course->id, false,
        MUST_EXIST);
} else {
    error ('You must specify an instance ID');
}

require_login ($course, true, $cm);
$PAGE->set_url ('/mod/iloir/view.php', array ('id' => $cm->id));
$PAGE->set_title (format_string ($iloir->name));
```

```

$PAGE->set_heading(format_string($course->fullname));
echo $OUTPUT->header();
echo $OUTPUT->heading($iloir->name);
$table = new html_table();
$table->head = array(get_string('iloirsubject', 'iloir'),
                    get_string('iloirformat', 'iloir'),
                    get_string('iloirilomasaplocation', 'iloir'));
$table->data = array(array($iloir->subject, $iloirformat,
                          $iloir->ilomasaplocation));
echo html_writer::table($table);
$subject = $iloir->subject;
$iloirId = $iloir->id;
$scormcontainer = "/ilomasweb/html/" .
                  "lmsilomaswebcontainer.html?courseid=".
                  $PAGE->course->id."&stdid=".$USER->id.
                  "&iloirid=".$iloirId."&subject=".$subject;

$iloirframesrc = $iloir->ilomasaplocation.$scormcontainer;
$parmFrameId = array('id' => 'playerFrameID');
$parmcc = array('id'=>'learningObjectIFrame',
                'name' => 'learningObjectIFrame', 'width' => '1000',
                'height' => '890', 'src' => $iloirframesrc);
$htmlcontainercode = html_writer::start_tag('div', $parmFrameId);
$htmlcontainercode .= html_writer::start_tag('iframe', $parmcc);
$htmlcontainercode .= html_writer::end_tag('iframe');
$htmlcontainercode .= html_writer::end_tag('div');
echo $htmlcontainercode;

```

Fonte: Autor (2015).

Além da tabela principal do *plug-in* ILOIR (m\_iloir), foram criadas algumas tabelas auxiliares na base de dados do Moodle (ex.: m\_iloir\_attempt, m\_iloir\_lo\_presented e m\_iloir\_scorm\_element), com o objetivo de manter armazenado todo o histórico de objetos de aprendizagem exibidos ao aluno durante uma sessão de aprendizagem, os elementos SCORM recebidos e enviados durante a interação com o estudante, os critérios de busca usados para selecionar novos objetos de aprendizagem, e a situação (completa ou incompleta) de cada sessão de aprendizagem (caso o aluno faça acesso ao mesmo recurso mais de uma vez).

As tabelas citadas poderão ser utilizadas futuramente para o desenvolvimento de relatórios que permitirão aos professores examinar no LMS Moodle detalhes das interações do aluno com um recurso ILOIR, com base nos dados armazenados nas referidas tabelas.

Cabe ressaltar que a utilização de um recurso ILOIR em um curso no Moodle não impede que o professor adicione outros recursos e atividades nativas do LMS (ou outros *plug-ins*) à estrutura do curso. Também é possível adicionar ao curso quantos recursos ILOIR se desejar, cada recurso relacionado a um tema diferente. A nota associada a cada recurso ILOIR é definida automaticamente pelo ambiente ILOMAS, de

acordo com o resultado consolidado da experiência de aprendizagem do aluno (situação definida quando o aluno decide encerrar aquela sessão de aprendizagem).

Todas as demais funções associadas com os recursos nativos do Moodle (página, arquivo, url, etc.) também estão disponíveis no *plug-in* ILOIR, tais como: definir um texto explicativo do recurso (descrição), configurações de agrupamento, de liberação de acesso e ocultação, etc. Inclusive é possível o professor definir uma nota associada ao recurso manualmente, e também adicionar a nota do(s) recurso(s) ILOIR configurado(s) no ambiente ao quadro de notas do curso.

## 5.5 AVALIAÇÃO DO PROTÓTIPO

Para avaliar o modelo proposto e o protótipo implementado, foram realizados alguns experimentos, os quais serão detalhados na sequência. Esses experimentos seguem a abordagem da validação por cenários, a qual, segundo Ferreira *et al.* (2013), vem sendo utilizada pela comunidade científica para avaliação de ambientes sensíveis ao contexto, como é o caso desta pesquisa.

Destaca-se que o intuito desta etapa é avaliar o comportamento adaptativo do sistema, e não sua efetividade educacional. Por questões de prazo e dificuldade em se identificar uma oportunidade pedagógica que se encaixasse no cronograma desta pesquisa, as avaliações para verificar a efetividade educacional do modelo proposto serão realizadas como trabalhos futuros.

Desta forma, o modelo ILOMAS foi utilizado para a criação de um curso introdutório de Direito Previdenciário, oferecido por uma escola de governo de um órgão público de âmbito estadual, para fins exclusivos de verificação da proposta.

O Modelo de avaliação buscou averiguar dois tipos de cenários: um de configuração de cursos, na forma convencional e outro de interação de alunos com o recurso ILOIR. Para tanto, foram realizadas comparações entre dois cursos criados no LMS Moodle.

O primeiro curso (1) foi elaborado com três objetos de aprendizagem empacotados no padrão SCORM (disponíveis no repositório de objetos), cujos temas estão relacionados ao Direito Previdenciário. Estes objetos foram inseridos isoladamente na estrutura do curso no LMS, como recursos do tipo SCORM (*plug-in* nativo, incorporado ao pacote de instalação padrão do Moodle).

O segundo curso (2) foi configurado com apenas um recurso do tipo ILOIR (novo *plug-in* desenvolvido), cujo assunto refere-se ao tema



do primeiro objeto SCORM inserido no curso mencionado anteriormente. Com o intuito de facilitar a descrição, daqui em diante o curso com objetos SCORM será chamado “Curso 1” e o curso com o recurso ILOIR será denominado “Curso 2”.

A Figura 36 ilustra a estrutura resultante do “Curso 1” após a sua configuração ser realizada.

Figura 36. Estrutura do curso com SCORM padrão



The image shows a Moodle course interface for 'Direito Previdenciário'. The page header includes 'Moodle', 'Português - Brasil (pt\_br)', and 'Professor 1'. The course title 'Direito Previdenciário' is prominently displayed. Below the title, there is a navigation breadcrumb: 'Página inicial > Meus cursos > Direito > DirPrevSCORM', and an 'Ativar edição' button. A vertical sidebar on the left contains navigation options: 'Navegação', 'Administração', 'Meus Fóruns', 'Minhas notícias', and 'Meus eventos'. The main content area features a 'Fórum de Avisos' section, followed by three topics: 'Tópico 1', 'Tópico 2', and 'Tópico 3'. Each topic is accompanied by a folder icon and the text 'Introdução ao Direito Previdenciário - Unidade 1', '2', and '3' respectively. A red rounded rectangle highlights these three topics. Below the topics is a 'Material Complementar' section with a document icon and the text 'Sistema de Legislação da Previdência Social'.

Fonte: Autor (2015).

A Figura 37 exibe a estrutura do “Curso 2”, o qual foi configurado com o recurso ILOIR.

Figura 37. Estrutura do curso com o recurso ILOIR



Fonte: Autor (2015).

Na sequência, será realizada a comparação entre os dois cursos supracitados configurados no Moodle, quanto aos critérios: (1) quantidade de recursos (itens) incluídos explicitamente no LMS, (2) quantidade de objetos de aprendizagem incluídos à estrutura do LMS, (3) tempo de configuração (tanto individual de um recurso, como total do curso) e (4) quantidade de objetos de aprendizagens disponíveis por

recurso adicionado. O Quadro 6 demonstra o resultado destas configurações.

Quadro 6 – Resultados da configuração dos cursos no LMS

	<b>Recursos incluídos</b>	<b>OAs incluídos</b>	<b>Tempo de configuração por recurso</b>	<b>Tempo total de configuração</b>	<b>Quantidade de OAs disponíveis</b>
Curso 1	3	3	00:02:18	00:08:30	1
Curso 2	1	0	00:00:58	00:04:30	Todos os OAs existentes no Repositório

Fonte: Autor (2015).

No que se refere à quantidade de itens explicitamente incluídos no LMS Moodle, note-se que para a configuração do “Curso 1” foi necessário adicionar três recursos do tipo SCORM, que é nativo da plataforma Moodle. Já para o “Curso 2”, apenas um recurso ILOIR foi acrescentado no ambiente virtual, resultando em uma quantidade menor de itens adicionados ao LMS.

Quanto à quantidade de objetos de aprendizagem incluídos no LMS, o recurso SCORM nativo do Moodle corresponde ao próprio objeto instrucional (arquivo no formato .zip), logo, no caso do “Curso 1”, três objetos de aprendizagem foram adicionados ao LMS.

Com relação ao “Curso 2”, cabe destacar que nenhum objeto de aprendizagem foi explicitamente adicionado no ambiente virtual, pois o recurso ILOIR é apenas uma camada de acesso ao sistema ILOMAS, o qual disponibilizará ao aluno objetos de aprendizagem, conforme deliberado pelo ILOAgent.

A quantidade de recursos (itens) do Moodle adicionados pelo responsável em configurar o curso influencia diretamente o terceiro critério de comparação, que é o tempo de configuração. A inclusão de um recurso SCORM (nativo do Moodle) na estrutura do curso leva mais tempo do que acrescentar um recurso ILOIR, pois a ação de configurar o recurso SCORM nativo inclui realizar o upload de um arquivo (no formato zip).

Como foram adicionados três itens SCORM no “Curso 1”, o intervalo total de configuração deste curso também foi maior do que o tempo necessário para configurar o “Curso 2”, onde foi incluído um único recurso ILOIR. Além disso, destaca-se também que o uso do recurso ILOIR resulta em menor uso de espaço em disco, pois este *plug-in* não

realiza upload de arquivos para a área de armazenamento do LMS, diferentemente do recurso SCORM nativo.

Além da redução na quantidade de itens a serem adicionados à estrutura do curso e de um menor tempo de configuração obtida com o uso do recurso ILOIR, o ponto de maior destaque na comparação refere-se à quantidade de objetos que podem vir a ser utilizados em uma sessão (experiência) de aprendizagem pelo estudante.

Enquanto o “Curso 1”, que utiliza os objetos de aprendizagem explicitamente incluídos, possui apenas o objeto SCORM adicionado como possibilidade de exibição, o “Curso 2”, que utiliza o *plug-in* ILOIR (que dá acesso aos Objetos Inteligentes de Aprendizagem do ambiente ILOMAS), permite que (de maneira dinâmica) vários objetos de aprendizagem distintos sejam apresentados ao aluno durante a experiência, desde que estes objetos estejam relacionados ao tema configurado pelo professor no recurso ILOIR. Esta característica oferecida pelo *plug-in* facilita o reuso de objetos de aprendizagem.

Seguindo a etapa de avaliação computacional do modelo ILOMAS, foi realizado outro experimento com base em um novo cenário, o qual visa testar a integração do ILOMAS com o modelo de empacotamento e sequenciamento SCORM, para reproduzir diferentes situações de aprendizagem.

Estas situações de ensino são representadas pela simulação de comportamento de três tipos de alunos:

- Um estudante que seleciona todas as respostas corretas (Aluno 1).
- Outro estudante que erra todas as questões (Aluno 2).
- Um terceiro estudante que melhora seu entendimento sobre o assunto durante a interação (Aluno 3).

O intuito deste cenário é verificar o comportamento dinâmico da interação dos alunos com o sistema ILOMAS, através do *plug-in* ILOIR.

O Quadro 7 demonstra os resultados obtidos com este segundo cenário de avaliação.

Quadro 7 – Resultados da interação simulada dos estudantes com o recurso ILOIR quanto ao comportamento dinâmico

	Respostas erradas	OA configurados previamente (ILOIR)	Novos OA oferecidos	Comportamento dinâmico
Aluno 1	0	1	0	Não
Aluno 2	4	1	4	Sim
Aluno 3	1	1	1	Sim

Fonte: Autor (2015).

Foram realizadas simulações com estas três situações de aprendizagem representadas pelo comportamento dos três tipos de alunos mencionados anteriormente (“Aluno 1”, “Aluno 2” e “Aluno 3”), através do *plug-in* ILOIR em uma instância do LMS Moodle.

Cada vez que era simulado um aluno cometendo um erro, o sistema ILOMAS identificava o problema de entendimento e sugeria outro OA associado ao mesmo assunto, com o intuito de reforçar o conhecimento do estudante sobre o respectivo tópico de ensino.

Quanto maior era a quantidade de erros apresentada por um aluno, maior era a quantidade de novos objetos de aprendizagem oferecidos a este estudante, potencializando o entendimento do aluno no assunto associado ao objeto. Ressalta-se que cada erro levava à oferta de um novo OA, pois este foi o comportamento implementado para o ILOAgent, na versão intermediária do protótipo utilizada neste segundo cenário de avaliação.

Por fim, foi realizado um terceiro cenário de simulação para contribuir com a avaliação do modelo proposto, o qual foi implementado através do protótipo (versão final) previamente descrito nas seções anteriores deste capítulo.

Este outro cenário buscou comparar o desempenho do aluno quanto: (1) à quantidade de assuntos apresentados ao aluno, (2) ao tempo de atenção do aluno na sessão de aprendizagem, (3) à quantidade de erros demonstrados pelo estudante e (4) à quantidade de objetos de aprendizagem extras (reforço) exibidos.

Com relação aos cenários de interação de estudantes com o *plug-in* ILOIR simulados, novamente foram considerados três alunos com

graus distintos de entendimento, porém em um curso introdutório sobre Direito Previdenciário. Os alunos simulados apresentavam os seguintes conceitos (nota) no curso: “ruim”, “regular” e “bom”.

O recurso ILOIR (configurado no curso) ofertou objetos de aprendizagem do tipo SCORM, fornecidos pela Academia Judicial do Poder Judiciário de Santa Catarina, que estavam disponíveis no repositório de objetos de aprendizagem que o ambiente ILOMAS tinha acesso (configurado localmente).

Esses objetos de aprendizagem utilizados possuem como conteúdo:

- Animações – O aluno deve visualizar todo o conteúdo apresentado, avançando nas telas que compõem o objeto.
- Questionários – Compostos de cinco questões de múltipla escolha sobre temas do curso, com cada questão valendo dois pontos, totalizando o máximo de dez pontos para o questionário. O aluno deve acertar pelo menos três questões para ter um desempenho considerado satisfatório no objeto.

O Quadro 8 exhibe os dados comparados entre os três tipos de alunos, conforme a simulação realizada.

Quadro 8 – Resultados da interação dos estudantes com o recurso ILOIR quanto à oferta de objetos de aprendizagem extras, conforme o desempenho

	<b>Conceito parcial</b>	<b>Assuntos exibidos</b>	<b>Tempo de atenção do aluno</b>	<b>Respostas erradas</b>	<b>OAs extras (reforço)</b>
Aluno 1	Bom	1	00:09:46	0	0
Aluno 2	Regular	2	00:21:13	3	1
Aluno 3	Ruim	3	00:34:19	6	3

Fonte: Autor (2015).

Nota-se que, dependendo do desempenho simulado do aluno (quantidade de respostas erradas) no questionário presente no objeto de aprendizagem, ou da demora em interagir com o OA atual, objetos adicionais eram exibidos na mesma experiência (sessão) de aprendizagem ao estudante, para que este pudesse ter maiores condições de elucidar suas dúvidas, reforçar conceitos ou superar lacunas de assuntos que são pré-

requisitos ao tema que ele deve compreender (ou seja, o objetivo de aprendizagem configurado pelo professor no recurso ILOIR).

O aluno com maior grau de desempenho no curso, que apresentava conceito parcial (nota) melhor, acessou menos recursos, pois o objeto oferecido inicialmente já foi suficiente para este estudante compreender o conteúdo e obter um resultado satisfatório na sessão de aprendizagem.

Entretanto, os alunos com desempenho “regular” ou “ruim” precisaram interagir com objetos adicionais, além do inicialmente oferecido. O intuito deste comportamento do sistema é potencializar a aprendizagem destes estudantes, com maior possibilidade de compreenderem os assuntos associados, através de objetos diferentes.

Além do desempenho no objeto e do grau de conhecimento prévio do estudante no curso (nota), a oferta de um OA extra levou em consideração também os metadados configurados para cada um destes recursos instrucionais, para que não fosse ofertado qualquer objeto, mas sim um OA relacionado ao tópico do objeto atual (o mesmo assunto, ou um assunto pré-requisito).

Desta forma, durante a experiência simulada, obteve-se a adaptação do conteúdo do objeto de aprendizagem de modo dinâmico (enquanto ocorria o acesso ao OA), conforme deliberado pelo ambiente de agentes (camada inteligente do sistema), segundo o resultado obtido pelo aluno na sessão de aprendizagem, isto é, o desempenho apresentado pelo estudante durante a interação com o objeto de aprendizagem.

Para ilustrar o comportamento de adaptação dinâmica (adaptatividade) verificado durante a execução deste cenário de avaliação, serão demonstradas na sequência algumas capturas de tela do acesso simulado do Aluno 3 (o qual apresentava um conceito parcial no curso classificado como “ruim”) ao plug-in ILOIR no ambiente Moodle.

A Figura 38 exibe o momento durante a simulação em que foi identificada e sinalizada ao Aluno 3 a necessidade de ser ofertado novo objeto de aprendizagem, durante a exibição do OA referente à “Unidade 2”. Esta situação foi identificada pelo sistema, devido aos erros em sequência (baixo desempenho) cometidos pelo estudante no questionário que compõe o objeto.

A quantidade de erros obtidos não mais permitiria que o estudante alcançasse um desempenho satisfatório na interação com o OA. Desta

forma, para possibilitar uma nova oportunidade de aprendizado, foi sinalizado ao estudante que um objeto extra seria ofertado.

Figura 38. Diagnóstico e sinalização de novo OA ao Aluno 3 (tempo t1)

The screenshot shows a Moodle quiz page titled "Introdução ao Direito Previdenciário". A red box highlights a diagnostic message that says: "A página em questão não foi encontrada: 8080 diz: Dear Student, it was identified that you need to know more about this subject. If you want to accept our suggestion, press the OK button. Otherwise, press the Cancel button and keep viewing the current Learning Object." Below this, a quiz question is displayed: "1 - Dadas as assertivas" followed by four multiple-choice options (A, B, C, D) regarding social security benefits. The score is currently 0 out of 6. A red box labeled "t1" is in the bottom right corner. At the bottom, there are two buttons: "I need to learn more about this subject" and "Finalize this Learning Experience".

Fonte: Autor (2015).



A Figura 39 apresenta a oferta do novo objeto de aprendizagem referente à “Unidade 1” (pré-requisito da “Unidade 2”) ao Aluno 3, para que este reforce sua compreensão sobre o assunto base.

Figura 39. Oferta do novo OA (pré-requisito do OA prévio) ao Aluno 3 (tempo t2)

Moodle Português - Brasil (pt\_br) Aluno 3

## Introdução ao Direito Previdenciário

Enjoy your Learning Experience!

**Unidade I**

Pergunta 5 De 5 Valor do Ponto: 2

5. Dadas as assertivas abaixo, assinalar a alternativa incorreta.

A) Pode-se definir o Período de Graça como sendo o tempo em que a lei permite ao indivíduo continuar amparado pelo Regime Geral de Previdência Social em caso de infortúnios, mesmo não estando a exercer atividade que o enquadre como segurado obrigatório, nem contribuir mensalmente, como facultativo, mantendo todos os direitos inerentes à condição de segurado.

B) O recebimento conjunto de mais de um benefício previdenciário não sofre limitações pela Lei n. 8.213/91.

C) Considera-se tempo de contribuição, para fins previdenciários, o tempo, contado de data a data, desde o início até a data do requerimento ou do desligamento de atividade abrangida pela Previdência Social, descontados os períodos legalmente estabelecidos como de suspensão de contrato de trabalho.

Pontuação até o momento: 8 pontos de 8 ENVIAR

Caution: May be necessary to install some required plugins to watch a Learning Object (i.e.: mp4 player plugin). Some objects are not shown in IE Browser. If you can't watch some Learning Object, please try it in another browser (Google Chrome, Mozilla Firefox, etc.).

I need to learn more about this subject Finalize this Learning Experience

t2

Fonte: Autor (2015).

## 5.6 LIMITAÇÕES

Julga-se necessário esclarecer algumas limitações existentes no trabalho apresentado nesta dissertação. Inicialmente, cabe destacar que o modelo proposto não utiliza informações relativas ao perfil (ou estilo) de aprendizagem do estudante (ex.: textual, visual, interativo) (DEL MORAL *et al.*, 2013; VESIN *et al.*, 2013) como um dos critérios de seleção dos objetos de aprendizagem. O escopo da pesquisa considera

essencialmente os resultados obtidos pelo aluno ao interagir com o objeto de aprendizagem (segundo modelo IEEE 1484.11.1, 2005).

Quanto à implementação do modelo, por questões de simplificação e limitação de escopo, foram levadas em consideração as seguintes informações no processo de adaptação do objeto de aprendizagem:

- A nota do aluno no curso do LMS Moodle e o assunto associado ao recurso instrucional.
- Alguns elementos do modelo de dados do padrão IEEE-LOM, tais como título (*title*), palavras-chave (*keywords*) e dificuldade (*difficulty*).
- O modelo da interação do aluno com o objeto de aprendizagem considera apenas alguns elementos do modelo de dados do padrão IEEE 1484.11.1 (2005), tais como resultados de questionários e tempo de interação.

Além disso, para que os objetos de aprendizagem possam ser acoplados ao ambiente multiagente (e conseqüentemente ao LMS), resultando na transformação do objeto simples em um OIA, é necessário que:

- O repositório dos recursos instrucionais descreva-os através de um padrão de metadados de objetos de aprendizagem.
- Estes objetos de aprendizagem tenham os respectivos elementos de metadados corretamente preenchidos (pelo menos os itens mais comuns).
- Os materiais instrucionais sejam compatíveis com um padrão de integração de objetos de aprendizagem que permita a obtenção de dados da interação do aluno com o recurso incorporado a um LMS.

Diante dessas premissas, optou-se por utilizar um repositório de objetos de aprendizagem local, configurado em um servidor próprio, que é acessível pelos Objetos Inteligentes de Aprendizagem no sistema multiagente, e que contém os recursos instrucionais citados no final da seção 5.5.2 (cedidos pela Academia Judicial do TJSC).

Conforme mencionado em capítulos anteriores, o modelo SCORM possibilita a descrição de fluxos de sequenciamento e navegação (SN), com base em regras que permitem a configuração prévia de seqüências de

exibição do conteúdo de um objeto de aprendizagem, de acordo com os resultados das interações do aluno. Entretanto, destaca-se que esta funcionalidade não foi utilizada na implementação do modelo para efetuar a seleção dinâmica de conteúdo do objeto de aprendizagem.

O motivo de se desconsiderar esta alternativa se deu pela dificuldade em se encontrar materiais educacionais que possuam estes elementos de metadados definidos, de modo que esta funcionalidade não se mostra acessível na prática.

A razão para estes elementos (assim como outros itens de metadados) não serem comumente utilizados pelos projetistas de recursos instrucionais, ou até mesmo serem configurados incorretamente, se deve ao fato de que realizar estas configurações costuma ser uma tarefa tediosa e complexa (DIAZ *et al.*, 2015). Sendo assim, procurou-se utilizar elementos de metadados que costumam estar presentes em objetos de aprendizagem reais de instituições de ensino.

## 5.7 DIFICULDADES ENCONTRADAS

É oportuno tecer algumas considerações quanto às dificuldades encontradas durante o desenvolvimento deste trabalho. Estas dificuldades podem ser agrupadas quanto à modelagem e à implementação do sistema multiagente, bem como quanto à tecnologia de objetos de aprendizagem utilizada.

Inicialmente, constatou-se que a modelagem e a implementação de sistemas multiagente não é uma tarefa trivial, exigindo do projetista e do desenvolvedor maiores preocupações e habilidades, em comparação com as metodologias de desenvolvimento de softwares tradicionais (como a abordagem OO).

Em se tratando de metodologia para o desenvolvimento de sistemas multiagente, pode ser feita uma crítica ao Prometheus. Este método deixa a desejar na modelagem de planos e crenças, no que se refere ao modelo individual do agente.

Quanto à implementação do sistema multiagente, duas críticas podem ser feitas à plataforma Jadex. A primeira diz respeito à complexidade de entendimento da estrutura do framework, em um primeiro momento, para aquele desenvolvedor que não esteja familiarizado com a abordagem de agentes.

A segunda crítica ao Jadex refere-se ao fato de esta plataforma não seguir a linha da representação das crenças dos agentes cognitivos através da lógica descritiva. Desta forma, há uma limitação quanto ao

detalhamento de informações nas crenças dos agentes, bem como na definição de pré-condições de planos.

Outra dificuldade resultante desta situação é a comparação com outras plataformas para o desenvolvimento de agentes BDI (como o Jason). A maioria das referências bibliográficas identificadas que relatam o desenvolvimento de sistemas multiagente baseado no modelo BDI utilizam plataformas onde as crenças são representadas através da lógica descritiva.

Por fim, podem ser citadas dificuldades em se desenvolver a integração do ambiente de agentes com o modelo SCORM e obter dados da interação através deste padrão de objetos de aprendizagem. O *player* SCORM presente no LMS Moodle não prevê pontos de integração com outros sistemas, logo foi necessário desenvolver um *player* SCORM próprio e integrá-lo ao ambiente Moodle, através do *plug-in* ILOIR.

Entretanto, a principal dificuldade foi encontrar uma documentação detalhada e tutoriais que descrevessem como implementar um *player* SCORM, e como obter as informações trocadas entre o LMS e o recurso digital (pacote SCORM). As funções JavaScript que compõem a estrutura do *Run-Time* do padrão apresentam relativa complexidade, assim foi necessário realizar um estudo do código presente no Moodle para entender melhor o fluxo de execução do SCORM.

## 5.8 CONTRIBUIÇÕES OBTIDAS

Esta seção descreve as contribuições obtidas com o desenvolvimento deste trabalho. As principais contribuições à área da Informática na Educação são o favorecimento do reuso de objetos de aprendizagem e a adaptação do conteúdo, de forma aderente ao aluno.

O modelo proposto mostrou-se adequado, segundo as validações realizadas através das simulações descritas na seção 5.5 (“Avaliação do Protótipo”) do capítulo atual. O protótipo implementado com o desenvolvimento de um *plug-in* para o LMS Moodle foi outra importante contribuição obtida.

Com relação ao campo da IA, identifica-se uma contribuição na área mais prática do desenvolvimento de aplicações baseadas em agentes, com a modelagem segundo o método Prometheus e o uso da plataforma Jadex na implementação do sistema.

Com relação à modelagem, pode-se afirmar que o emprego da metodologia Prometheus auxiliou a tarefa de confecção do modelo de

agentes, principalmente pela modelagem inicial através da identificação dos objetivos (abstração), e da ferramenta PDT.

Quanto à implementação do sistema multiagente, destaca-se que a utilização do Jadex facilitou a implementação da solução, especialmente pela estrutura da versão mais atual da plataforma, baseada na abordagem OO. Além disso, o uso do Jadex no desenvolvimento do protótipo também contribuiu para a disseminação desta ferramenta.

Desta forma, pode-se afirmar que foi obtido um bom resultado quanto ao uso da tecnologia de agentes como abordagem para a seleção dinâmica de objetos de aprendizagem.

## 5.9 DISCUSSÕES SOBRE A INTEGRAÇÃO ENTRE O MODELO PROPOSTO E ONTOLOGIAS

Esta seção visa realizar algumas discussões referentes a questões levantadas durante o texto desta dissertação sobre a integração do modelo proposto com ontologias, com o intuito de justificar algumas decisões tomadas, baseado em situações aprendidas e conhecimentos obtidos durante o desenvolvimento desta pesquisa.

No decorrer desta dissertação, foram mencionados os benefícios de se integrar ontologias ao modelo ILOMAS, principalmente no que se refere a busca semântica por objetos de aprendizagem, os quais podem ser selecionados dinamicamente durante a experiência de aprendizagem de um estudante.

Porém, é importante deixar claro que não foi desenvolvida uma integração entre os agentes do protótipo implementado e a ontologia que representa o Modelo de Dados, por razões práticas previamente mencionadas. Entretanto, cabe ressaltar que isso não impede que sejam analisados detalhes relacionados a estas questões de representação ontológica no agente OIA (Objeto Inteligente de Aprendizagem) do modelo teórico.

Segundo identificado em trabalhos que procuram propor evoluções ao estado da arte na área de sistemas multiagente, a tendência para a integração entre agentes e ontologias é que os conceitos e elementos da ontologia sejam internalizados no agente.

Desta forma, torna-se possível a representação de informações semânticas nos agentes, resultando em consultas mais expressivas na base de crenças, e no refinamento do processo de revisão de crenças. Um exemplo desta tendência é a extensão da plataforma Jason para programar

agentes com raciocínio ontológico, denominada JASDL (KLAPISCAK e BORDINI, 2009; SANTOS *et al.*, 2009).

Com a inclusão desta característica no ILOAgent do modelo proposto, obtém-se uma maior autonomia do agente com a representação do modelo de dados internamente em sua base de crenças. Assim, passa a ser possível o projeto de agentes que utilizem diferentes critérios para recomendar os objetos de aprendizagem (ex.: alguns tipos de agentes mais benevolentes com o erro dos alunos, outros menos tolerantes a falhas de aprendizagem, ou ainda, agentes que ofereçam ajuda mais rapidamente, enquanto outros que evitem ao máximo intervir na experiência de aprendizagem, etc.).

Por outro lado, em alguns casos, incluir informações de ontologias internamente no agente não é algo interessante, devido a diminuição de autonomia que o agente passará a ter, pois a entidade ficará muito dependente destas ontologias para deliberar e realizar suas ações. Além disso, o processo de revisão de crenças, e de raciocínio precisará incluir etapas semânticas, resultando em possíveis problemas de performance (KLAPISCAK e BORDINI, 2009).

Desta forma, quanto à integração do modelo teórico de agentes proposto com ontologias, é importante separar uma ontologia que represente informações que sejam de competência do agente OIA de outras que sirvam apenas para consulta deste agente.

Um exemplo são ontologias públicas de domínio, que representam conceitos de um determinado domínio educacional. Os conceitos associados ao assunto “derivadas”, por exemplo, dificilmente se modificam (salvo se vierem a ser publicados novas descobertas e propostas neste campo). Assim, a ontologia deste domínio pode ser reaproveitada por vários tipos de agentes.

No contexto desta dissertação, pode-se vislumbrar o uso de uma ontologia de domínio para os agentes que representam um objeto inteligente de aprendizagem (OIA). Quando um OIA estiver monitorando uma sessão de aprendizagem, e vier a deliberar que um aluno está com dificuldade no assunto associado ao objeto de aprendizagem, este agente pode consultar a ontologia do domínio educacional associado, no exemplo “derivadas”, e identificar através desta ontologia que o estudante possivelmente possui deficiência no conceito de “funções do segundo grau” (conceito relacionado), e desta forma precisa estudar sobre esse assunto pré-requisito.

Entretanto, podem existir agentes OIA que possuam diferentes estratégias educacionais para auxiliar o aluno no estudo de “funções do segundo grau”, inclusive com ações diferentes em circunstâncias

diferentes para adaptar o conteúdo educacional (ex.: períodos do dia distintos, histórico do aluno, etc.).

Com este cenário, como o entendimento sobre um domínio específico não é *expertise* do agente OIA, pode-se fazer uso de um raciocinador externo (ex.: Jena e Pellet) como artefato acessado pelo agente (DIBLEY *et al.*, 2011) para consultar as ontologias de domínio, permitindo buscar e recomendar objetos de aprendizagem de assuntos relacionados ao OA atual, tais como o assunto “química dos sólidos” e o conceito relacionado “tabela periódica dos elementos”.

Em contrapartida, faz sentido incluir no modelo teórico a ontologia do modelo de dados (LMS, OA e Interação) como parte integrante do agente (crenças), devido a esta ontologia representar elementos que são utilizados nas decisões tomadas pelo agente para encaminhar o processo de ensino, ou seja, para ofertar novos objetos de aprendizagem.

Esta ontologia da modelagem conceitual (modelo de dados) permitirá o mapeamento entre os elementos conceituais do modelo teórico com os elementos dos padrões de objetos de aprendizagem específicos, possibilitando que sejam feitas mudanças de padrões (bem como alterações em elementos dos padrões existentes) sem alterar a estrutura do agente (modelo mental), mantendo o funcionamento do sistema.

Logo, para integrar-se ao protótipo implementado, a ontologia do modelo de dados deve definir também os elementos dos padrões IEEE-LOM e SCORM e LOM (modelos específicos de metadados e de integração de objetos, respectivamente), juntamente com suas respectivas faixas (*range*) de valores. Como resultado, as crenças do agente (modelo mental) seriam basicamente uma cópia da estrutura do padrão SCORM e LOM, pois estes são os únicos padrões utilizados no protótipo.

A definição da estrutura de informação para o processo de decisão do OIA através de uma ontologia permitirá que a modificação na arquitetura do agente seja mínima, por exemplo, caso não se queira mais usar o modelo com o padrão SCORM e sim utilizar objetos compatíveis com a xAPI (HRUSKA *et al.*, 2015), como uma evolução ao protótipo.

O emprego de um novo padrão poderia ser feito através do mapeamento de sinônimos de elementos SCORM e IEEE-LOM na ontologia do modelo de dados, para os elementos do novo padrão (conceitos equivalentes).

Por outro lado, caso todas estas informações estivessem definidas apenas internamente ao agente, em suas crenças, seria necessário

reimplementar totalmente o agente para tornar possível a integração com a xAPI. Por isso, acredita-se que a estratégia proposta é mais vantajosa.

Para concluir essas discussões relacionadas ao uso de ontologias integradas ao ILOMAS, cabe frisar a evolução ao modelo descrita anteriormente nesta seção, quanto à possibilidade de se representar as informações ontológicas internamente em um agente, ou de apenas consultar uma ontologia externa, sem que sejam feitas representações dos elementos desta ontologia externa no agente.

O modelo mental do OIA (ILOAgent) descreve quais informações são utilizadas para modificar o sequenciamento de objetos de aprendizagem, com base em padrões de objetos. Desta forma, é oportuno que se busque na ontologia do modelo de dados qual é o padrão utilizado, quais são os elementos existentes para poder ser feita a adaptação do conteúdo, bem como seus possíveis valores (faixas).

Além disso, o ILOAgent tem como *expertise* tratar o sequenciamento de objetos, de acordo com o desempenho do aluno. Assim, pode ser usada a metáfora do agente OIA como sendo o modelo de um pedagogo, o qual conhece elementos usados no processo de ensino relacionados à oferta de objetos de aprendizagem segundo padrões de referência.

Desse modo, pode-se perceber que o modelo é independente de padrão de objetos de aprendizagem, e essa independência é tratada na ontologia do modelo de dados. O conhecimento pedagógico é de competência do OIA, justificando a necessidade de a ontologia do modelo de dados estar integrada ao agente, e ter seus elementos representados como crenças do ILOAgent.

Neste caso, não é necessário consultar raciocinador externo para decidir se o OIA deve ofertar ou não um novo OA. Porém o ILOAgent não é especialista em domínio, por isso o que for relacionado a domínio é considerado externo ao agente, justificando-se o uso de raciocinador sobre uma (ou mais) ontologia externa de domínio (cujos elementos não sejam representados nas crenças do OIA), com o intuito de se identificar objetos de aprendizagem semanticamente relacionados.



## 6 CONSIDERAÇÕES FINAIS

Neste capítulo final, primeiramente será feita uma revisão da estrutura deste documento, mencionando brevemente aquilo que foi exposto no decorrer do trabalho, em cada respectivo capítulo. Finalmente, serão sugeridas evoluções ao modelo proposto, com a indicação de possíveis trabalhos futuros.

### 6.1 CONCLUSÃO

Esta dissertação buscou desenvolver um modelo que proporcione experiências de ensino com maior adaptação do conteúdo e favorecimento do reuso, em comparação com outros trabalhos correlatos, e com a finalidade de se responder a seguinte questão de pesquisa:

*Como desenvolver sistemas de ensino on-line adaptativos e reutilizáveis de maneira dinâmica, com base no desempenho do aluno?*

Conforme descrito na seção 1.3 do Capítulo 1 deste documento, o objetivo geral da pesquisa era:

*Propor um modelo de seleção dinâmica de Objetos de Aprendizagem (OA), de forma adaptativa, baseado na abordagem multiagente, e que favoreça o reuso de OA.*

Este objetivo genérico foi alcançado através da elaboração do modelo ILOMAS, o qual foi proposto e descrito em detalhes nos capítulos 4 e 5 desta dissertação. Contudo, até se chegar à elaboração, ao desenvolvimento e à implementação do modelo, foram realizadas pesquisas e estudos prévios, conforme relatado nos capítulos iniciais deste documento.

O Capítulo 1 contextualizou o problema de pesquisa, relatou os objetivos que se desejava atingir com o desenvolvimento deste trabalho, e listou as contribuições esperadas. Os critérios de qualidade de softwares educacionais elencados no capítulo introdutório, a adaptatividade e a reusabilidade, foram elementos-chave que guiaram as decisões tomadas durante a pesquisa e os caminhos trilhados.

Seguindo a estrutura da dissertação, no Capítulo 2, foram detalhadas as teorias relacionadas aos elementos, conceitos e abstrações

utilizados neste estudo, e que serviram de apoio à elaboração do modelo teórico proposto.

Em seguida, buscou-se por pesquisas correlatas ao tema deste estudo, existentes na literatura relacionada ao campo da Informática na Educação e da Inteligência Artificial (ou Computacional) aplicada a Educação. Os trabalhos selecionados foram comparados no Capítulo 3, resultando na identificação de questões passíveis de evolução.

Baseado no cenário encontrado, e em trabalhos anteriores do grupo de pesquisa IATE, foi proposto um novo modelo baseado em agentes cognitivos, que estende a abordagem de Objetos Inteligentes de Aprendizagem, para permitir a adaptação de uma experiência educacional resultante da interação de um aluno com um objeto de aprendizagem, conforme o desempenho deste aluno, e que seja realizada em um sistema de gerenciamento de aprendizagem.

Assim, buscou-se atingir o primeiro objetivo específico descrito no capítulo introdutório desta dissertação:

- *Desenvolver um modelo teórico que estenda a abordagem de objetos inteligentes de aprendizagem.*

A partir dos elementos existentes nos padrões de referência de objetos de aprendizagem e das características que o modelo deve apresentar (capítulos 1, 2 e 3), foi elaborado um modelo conceitual de dados, que envolve informações do LMS, dos objetos de aprendizagem e da interação do aluno com o objeto. Em seguida, foi proposto um modelo de agentes BDI, de tal forma que o comportamento destas entidades cognitivas é guiado por objetivos que desejam atingir (raciocínio prático).

Os agentes cognitivos se utilizam dos dados supracitados para deliberarem sobre a necessidade (ou não) de apresentar novos objetos de aprendizagem para os estudantes de um curso on-line, de acordo com o desempenho demonstrado na sessão de aprendizagem.

Estas entidades compõem um sistema multiagente denominado ILOMAS, que foi projetado conforme a metodologia AOSE Prometheus. O sistema proposto é capaz de se comunicar com elementos de um LMS e de objetos de aprendizagem segundo padrões de referência (de metadados e de integração), possibilitando que a etapa de adaptação de

conteúdo educacional seja feita de forma integrada com ambientes virtuais de aprendizagem largamente difundidos em todo o mundo.

Desta forma, outros dois objetivos específicos listados na introdução foram cumpridos:

- *Propor um modelo conceitual de dados que serão utilizados pelos agentes cognitivos.*
- *Especificar o modelo da sociedade dos agentes que irão habitar no ambiente multiagente através de metodologias de desenvolvimento orientado a agentes (AOSE).*

Com o intuito de instanciar o modelo teórico proposto foi implementado um protótipo com o objetivo de verificar minimamente que este modelo é viável. Este protótipo foi descrito em detalhes no Capítulo 5 desta dissertação.

No desenvolvimento do referido protótipo, foram realizadas limitações de escopo e consideradas algumas questões de simplificação de implementação, tais como a utilização de um subconjunto dos elementos do modelo de dados. Estas premissas foram estabelecidas para possibilitar que a pesquisa também apresentasse um caráter prático, com o desenvolvimento de uma aplicação.

A solução implementada permite que objetos de aprendizagem compatíveis com o padrão de empacotamento e sequenciamento SCORM sejam selecionados dinamicamente em cursos oferecidos no LMS Moodle, conforme o desempenho do aluno no curso e durante a interação com o objeto.

O SMA, implementado com base na plataforma Jadex BDI V3, torna possível o acesso das funcionalidades dos agentes como serviços, através da camada *web*, a qual integra o ambiente ILOMAS (parte inteligente da aplicação) e a camada cliente (*browser* do estudante).

A implementação da comunicação entre ILOMAS e a API SCORM possibilita que os dados da interação do aluno com o objeto sejam utilizados no processo de deliberação dos agentes, tais como: o resultado de um questionário, o percentual de telas avançadas e o tempo decorrido na sessão de aprendizagem.

O histórico das interações anteriores (dentro da mesma sessão de aprendizagem) também é utilizado pelos agentes para decidir se um novo objeto de aprendizagem precisa ser ofertado ao estudante, com o

propósito de potencializar a capacidade de entendimento do aluno sobre um determinado conceito.

Após o detalhamento de como o sistema foi projetado e implementado, foi descrito o artefato (*plug-in*) ILOIR, criado para integrar o ambiente ILOMAS ao LMS Moodle.

Em seguida, foram descritos os testes de avaliação do modelo teórico proposto, do ponto de vista computacional, os quais foram realizados por meio de simulações representando o acesso de estudantes com diferentes níveis de desempenho em um curso de Direito Previdenciário, através do *plug-in* ILOIR.

Os experimentos efetuados foram comparados com o intuito de se verificar as características de adaptatividade e reusabilidade elencadas no capítulo inicial desta dissertação. Além de favorecerem tais características de adaptação do conteúdo educacional, através da seleção dinâmica de objetos de aprendizagem, e de reuso de materiais instrucionais, os testes realizados indicaram a viabilidade computacional do modelo proposto.

Além disso, o capítulo 5 também descreveu as tecnologias utilizadas na implementação do modelo, as limitações do protótipo, as dificuldades encontradas, as contribuições obtidas, bem como discussões sobre o uso de ontologias integradas ao modelo ILOMAS.

Todas estas atividades realizadas e descritas no Capítulo 5, serviram para que se pudesse alcançar os últimos três objetivos específicos elencados na introdução deste documento:

- *Projetar a integração do sistema multiagente elaborado com o padrão IEEE-LOM, o padrão SCORM e o LMS Moodle.*
- *Avaliar o modelo proposto com base no protótipo desenvolvido, através da simulação de cursos.*

## 6.2 TRABALHOS FUTUROS

Conclui-se esta dissertação com a descrição de possíveis evoluções ao modelo ILOMAS, o qual foi apresentado neste trabalho. Os avanços elencados podem agrupados quanto ao tipo, sendo: (1) do ponto de vista das características e funcionalidades do modelo de seleção dinâmica de

objetos de aprendizagem; (2) no âmbito da tecnologia de agentes; e por fim, (3) do ponto de vista educacional.

Desta forma, destaca-se que futuramente o modelo de seleção dinâmica de OA pode ser estendido para incluir as seguintes características:

- Utilizar informações de estilos de aprendizagem (DEL MORAL *et al.*, 2013), características de afetividade (JAQUES e VICARI, 2005) ou acessibilidade (LUQUE *et al.*, 2014; MELO, 2014) para realizar a adaptação do conteúdo.
- Integrar-se a um sistema de recomendação para indexação e recuperação de objetos de aprendizagem, como o descrito em Campos *et al.* (2013), expandindo a lista de objetos de aprendizagem relativos ao tema definido pelo professor, e possibilitando também o aprimoramento da tomada de decisão dos agentes para a escolha de objetos que serão ofertados.
- Enriquecer o processo de seleção de novos objetos de aprendizagem, levando-se em consideração: a estrutura de competências que os alunos devem desenvolver, a hierarquia de pré-requisitos de cada tema, etc. (DE-MARCOS *et al.*, 2007).
- Integrar-se a repositórios externos (e públicos) de recursos instrucionais (ex.: MERLOT e ARIADNE).
- Utilizar o componente de Sequenciamento e Navegação do SCORM (SN, 2009) para agregar outros dados ao modelo da interação do aluno com o OA.
- Integrar o sistema multiagente a outros ambientes virtuais de aprendizagem (LMS), como por exemplo: Blackboard (2015) e Claroline (2015).
- Utilizar outros padrões de objetos de aprendizagem, como IMS *Common Cartridge* (DA NÓBREGA *et al.*, 2010) ou TinCan (DE

NIES *et al.*, 2015), que é considerado a evolução do SCORM (PAPAZOGLAKIS, 2013).

- Utilizar outros padrões de metadados de objetos de aprendizagem, como Dublin Core ou OBAA (GLUZ e VICARI, 2011).
- Permitir a comunicação do sistema com outras plataformas de agentes, facilitando a interoperabilidade entre ambientes educacionais distintos (SANTOS *et al.*, 2006).
- Adquirir informações de dados educacionais na *web* (SANTOS *et al.*, 2009).
- Possibilitar a busca semântica por objetos de aprendizagem, através da qualificação e expansão de elementos dos metadados (como nível educacional ou estilo de aprendizagem), associando estes elementos a conceitos (e indivíduos) de ontologias, e outras abordagens da *Web Semântica*, inclusive com a realização de inferências sobre as ontologias (CAMPOS, 2013; PRIMO, 2013; SALEENA e SRIVATSA, 2014).

Com relação a trabalhos futuros que resultem em contribuições para área de IA, em especial ao desenvolvimento de agentes, podem ser listados:

- Realizar estudo comparativo da implementação do modelo de agentes proposto, através de uma plataforma que utilize lógica descritiva na representação das crenças (como o Jason) em relação ao Jadex (representa as crenças através da tecnologia OO). O objetivo será verificar as vantagens e desvantagens existentes em cada uma das alternativas, principalmente identificando o que não foi possível ser feito no protótipo implementado e que poderia ser feito, ao se representar as crenças através de lógica.
- Realizar estudo comparativo da modelagem proposta, através de outras metodologias de desenvolvimento de agentes (tais como: O-MaSE, Gaia, Tropos, etc.) em relação ao Prometheus. O intuito será analisar as vantagens e desvantagens existentes em cada um dos métodos, e especialmente, verificar se a modelagem dos agentes através de outra metodologia traria benefícios.
- Realizar estudo referente à modelagem de agentes híbridos que possuam uma base de conhecimento mista (com e sem o uso de

lógica descritiva), prevendo um cenário de aplicação para esta arquitetura híbrida de agentes.

- Realizar análise comparativa dos trabalhos relacionados (apresentados no capítulo 3), com relação às formas de representação das crenças, às formas de deliberação, e às formas de implementação dos planos, do ponto de vista técnico.
- Realizar estudo comparativo entre o modelo proposto nesta dissertação e todos os trabalhos relacionados (Capítulo 3) baseados em SMA (mesmo que o propósito do trabalho relacionado não seja a adaptação). O intuito desta análise será demonstrar em quais situações é indicado o desenvolvimento de um sistema como este proposto nesta dissertação.

Finalmente, do ponto de vista educacional, trabalhos futuros envolvem a avaliação do modelo proposto com o intuito de verificar a sua efetividade didática. Desta forma, devem ser realizadas validações com outros tipos de cenários educacionais em instituições distintas, e com alunos de faixas etárias diversas, em situações reais de ensino-aprendizagem.

Trabalhos futuros que avaliem o sistema com a participação de seres humanos (e não apenas simulações) irão possibilitar que sejam verificadas questões relacionadas ao processo de ensino-aprendizagem, tais como:

- O quanto a abordagem dos Objetos Inteligentes de Aprendizagem (OIA) impacta efetivamente na experiência educacional.
- Como os alunos aceitam esta solução.
- Como os estudantes descrevem a experiência.
- Quais as consequências (positivas e negativas) de se inserir a ferramenta ILOIR em uma prática educacional.





## REFERÊNCIAS

- ABBAD, G. **Educação a distância: o estado da arte e o futuro necessário**. Revista do Serviço Público, Vol. 58, Nº 3, p. 351-374, 2007.
- ADAMUSIAK, T.; BURDETT, T.; KURBATOVA, N.; VAN DER VELDE, K.; ABEYGUNAWARDENA, N.; ANTONAKAKI, D.; KAPUSHESKY, M.; PARKINSON, H.; SWERTZ, M. **OntoCAT – Simple ontology search and integration in Java, R and REST/JavaScript**. BMC bioinformatics, Vol. 12(1), 218. 2011. Disponível em: <<http://www.biomedcentral.com/1471-2105/12/218>>. Acesso em: 10 nov. 2015.
- ADL. **SCORM 2004**. Sharable Content Object Reference Model. Advanced Distributed Learning, 2004. Disponível em: <<http://www.adlnet.org/scorm>>. Acesso em: 10 nov. 2015.
- ADL. **SCORM Run-Time Environment (RTE)**, SCORM 2004 4th Edition, Version 1.1, Advanced Distributed Learning, 2009.
- ADL. **SCORM Sequencing and Navigation (SN)**, SCORM 2004 4th Edition, Version 1.1, Advanced Distributed Learning, 2009.
- ADL. **SCORM Users Guide for Instructional Designers**, SCORM 2004 4th Edition, Version 8, Advanced Distributed Learning, 2011.
- ADL. **SCORM Users Guide for Programmers**, SCORM 2004 4th Edition, Version 10, Advanced Distributed Learning, 2011.
- ADL. **Training and Learning Architecture (TLA)**, 2012. Disponível em: <<http://adlnet.gov/ta>>. Acesso em: 10 nov. 2015.
- AICC. **AICC Document Archive**. Aviation Industry CBT (Computer-Based Training) Committee. 2014. Disponível em: <<https://github.com/ADL-AICC/AICC-Document-Archive>> Acesso em: 10 nov. 2015.
- ALLISON, C.; MILLER, A.; OLIVER, I.; MICHAELSON, R.; TIROPANIS, T. **The Web in education**. Computer Networks 56, Elsevier, p. 3811-3824, 2012.
- ALENCAR, M.; NETTO, J. **Improving Cooperation in Virtual Learning Environments Using Multi-Agent Systems and AIML**. 41st ASEE/IEEE Frontiers in Education Conference, Session F4C, IEEE, p. 1-6, 2011.

- AMADEUS. **Amadeus LMS**. Software Público Brasileiro. Disponível em: <<http://www.softwarepublico.gov.br/dotlrn/clubs/amadeus/>>. Acesso em: 10 nov. 2015.
- AMORIM JR., J.; GELAIM, T.; SILVEIRA, R. **Dynamic e-Learning Content Selection with BDI Agents**, In: J. Bajo *et al.* (Eds.): PAAMS 2015 Workshops, CCIS 524, Springer, p. 299–308, 2015.
- AMORIM JR., J.; SILVEIRA, R. **Dynamic Selection of Learning Objects Based on SCORM Communication**, In: F. Pereira *et al.* (Eds.) EPIA 2015, LNAI 9273, Springer, pp. 658-663, 2015.
- AMORIM JR., J.; SILVEIRA, R. **Seleção Dinâmica de Objetos de Aprendizagem Baseados no Padrão SCORM no Ambiente Moodle**. Anais do XXVI SBIE, p. 967-976, 2015.
- ANITHA, D.; DEISY, C. **A novel approach for selection of Learning Objects for personalised delivery of elearning content**. Computer Science & Information Technology, p. 413-420, 2013.
- ARDITO, C.; COSTABILE, M.; MARSICO, M.; LANZILOTTI, R.; LEVIALDI, S.; ROSELLI, T.; ROSSANO, V. **An approach to usability evaluation of e-learning applications**. Universal Access in the Information Society, Vol. 4 (3), Springer, p. 270-283, 2006.
- ARIADNE. **ARIADNE Foundation**. Global Learning Objects Brokering Exchange (GLOBE) Alliance. Disponível em: <<http://www.ariadne-eu.org>>. Acesso em: 10 nov. 2015.
- ARTICULATE. **Articulate Software**. Articulate Global, Inc. Disponível em: <<https://www.articulate.com/>>. Acesso em: 10 nov. 2015.
- BACHARI, E.; ABELWAHED, E.; ADNANI, M. **E-Learning personalization based on dynamic learners' preference**. International Journal of Computer Science & Information Technology (IJCSIT), Vol 3, No 3. IJCSIT, 2011.
- BALTHASAR, G.; SUDEIKAT, J.; RENZ, W. **On the decentralized coordination of herding activities: a Jadex-based solution**. Annals of Mathematics and Artificial Intelligence, Vol. 59, Nº 3-4. Springer, p. 411-426, 2010.
- BATISTA, S.; BARCELOS, G.; ELENA, C. **Avaliar é Preciso: o caso de softwares educacionais para Matemática no Ensino Médio**. Trabalho apresentado no I WorkComp Sul, Florianópolis, 2004.

- BAVARESCO, N.; SILVEIRA, R. **Proposta de uma arquitetura para construção de Objetos Inteligentes de Aprendizagem baseados em agentes BDI**. Anais do XX SBIE, 2009.
- BECHHOFFER, S.; VOLZ, R.; LORD, P. **Cooking the Semantic Web with the OWL API**. In: The Semantic Web-ISWC 2003. Springer. p. 659-675, 2003.
- BINAS, A.; IOERGER, T. **Multi-Agent Belief Reasoning in a First-Order Logic Back-Chainer**. Technical Report. 2004.
- BIOE. **Banco Internacional de Objetos Educacionais**. Ministério da Educação – Brasil. Disponível em: <<http://objetoseducacionais2.mec.gov.br/>>. Acesso em: 10 nov. 2015.
- BITTENCOURT, G. **Inteligência Artificial: ferramentas e teorias**. 3ª Edição Revisada. Série Didática. Editora da UFSC. 2006.
- BLACKBOARD. **Blackboard LMS**. Grupo A. Disponível em: <<http://www.blackboard.com/>>. Acesso em: 10 nov. 2015.
- BOISSIER, O.; BORDINI, R.; HÜBNER, J.; RICCI, A.; SANTI, A. **Multi-agent oriented programming with JaCaMo**. Science of Computer Programming, Vol. 78, Nº 6, p. 747-761, 2013.
- BOYCE, S.; PAHL, C. **Developing domain ontologies for course content**. Journal of Educational Technology & Society, Vol. 10, Nº 3, p. 275-288, 2007.
- BOLZAN, W.; GIRAFFA, L. **Estudo comparativo sobre sistemas tutores inteligentes multiagentes Web**. FACIN-PUCRS, Porto Alegre, 2002.
- BONSON, Jéssica Pauli de Castro. **Aplicação de Agentes & Artefatos para acessar a Web Semântica: Desenvolvimento de uma ferramenta de autoria de metadados de Objetos de Aprendizagem**. Trabalho de Conclusão do Curso de Graduação em Ciência da Computação. UFSC. Florianópolis, 2013.
- BOTTENTUIT JUNIOR, J.; COUTINHO, C. **Educação On-line: Conceitos, metodologias, ferramentas e aplicações**. Editora CRV, p. 7, 2012.
- BRANDÃO, Anarosa Alves Franco. **Um método para estruturação e análise de modelos de sistemas multiagentes baseado em ontologias**. Tese de Doutorado em Informática. PUC-Rio, Rio de

Janeiro, 2005.

BRASE, J.; NEJDL, W. **Ontologies and Metadata for eLearning**. In: Handbook on ontologies. Springer, p. 555-573, 2004.

BRATMAN, M.; **Intention, plans, and practical reason**. Harvard University, Cambridge, 1987.

BREMGARTNER, V.; NETTO, J.; MENEZES, C. **Utilizando Agentes e Ontologia de Modelo de Aluno Aberto para Prover Adaptação de Conteúdos Construtivistas em Ambientes Virtuais de Aprendizagem**. Anais do XXV SBIE, 2014.

BREMGARTNER, V.; NETTO, J.; MENEZES, C. **Explorando Arquiteturas Pedagógicas Recomendadas por meio de Agentes e Ontologia de Modelo do Aluno em Ambientes Virtuais de Aprendizagem**. Anais do XXVI SBIE, 2015.

BROWN, E.; CRISTEA, A.; STEWART, C.; BRAILSFORD, T. **Patterns in Authoring of Adaptive Educational Hypermedia: A Taxonomy of Learning Styles**. In: Education Technology & Society, 8 (3), IFETS, p. 77-90, 2005.

BUSS, N.; HÜLSE, W. **Desafios na Gestão da Educação a Distância do Poder Judiciário de Santa Catarina**. In: Coleção Judiciário do Futuro: Gestão Organizacional e Administração de Recursos Humanos. Novas Tecnologias e Eficiência no Judiciário. Vol. 1, PJSC e UFSC. Academia Judicial. p. 195-218, 2015. Disponível em: <[http://acadjud.tjsc.jus.br/c/document\\_library/get\\_file?uuid=012b3177-d3a9-49d5-80c2-7a94c4ddcfbe&groupId=10157](http://acadjud.tjsc.jus.br/c/document_library/get_file?uuid=012b3177-d3a9-49d5-80c2-7a94c4ddcfbe&groupId=10157)>. Acesso em: 10 nov. 2015.

CAMPOS, Diogo de. **Representação de Dados Semânticos em Agentes BDI**. Dissertação de Mestrado em Ciência da Computação. UFSC. Florianópolis, 2014.

CAMPOS, Ronaldo Lima Rocha. **Modelo de Busca Inteligente e Recomendação de Objetos de Aprendizagem em Repositórios Heterogêneos**. Dissertação de Mestrado em Ciência da Computação. UFSC. Florianópolis, 2013.

CAMPOS, R.; COMARELLA, R.; SILVEIRA, R. **Multiagent Based Recommendation System Model for Indexing and Retrieving Learning Objects**. In: Highlights on PAAMS, CCIS, Vol. 365, Springer, p. 328-339, 2013.

- CANO, C. **Os recursos da informática e os contextos de ensino e aprendizagem.** In: SANCHO, J. Para uma tecnologia educacional. Porto Alegre: ArtMed, p. 169, 1998.
- CAPTIVATE. **Adobe Captivate.** Adobe Systems Incorporated. Disponível em: <<http://www.adobe.com/br/products/captivate.html>>. Acesso em: 10 nov. 2015.
- CAREO. **Campus Alberta Repository of Educational Objects.** CAREO Project. Disponível em: <<http://www.careo.org/>>. Acesso em: 10 nov. 2015.
- CARRERA, A.; IGLESIAS, C.; GARCÍA-ALGARRA, J.; KOLARÍK, D. **A real-life application of multi-agent systems for fault diagnosis in the provision of an Internet business service.** Journal of Network and Computer Applications, v. 37, Elsevier, p. 146-154, 2014.
- CARVALHO, V.; DORÇA, F.; CATTELAN, R.; ARAÚJO, R. **Uma Abordagem para Recomendação Automática e Dinâmica de Objetos de Aprendizagem Baseada em Estilos de Aprendizagem.** Anais do XXV SBIE, 2014.
- CASALI, A.; DECO, C.; ROMANO, A.; TOMÉ, G. **An assistant for loading learning object metadata: An ontology based approach.** Interdisciplinary Journal of E-Learning and Learning Objects. Vol. 9, p. 77-87, 2013.
- CESTA. **Coletânea de Entidades de Suporte ao uso de Tecnologia na Aprendizagem.** CINTED, UFRGS. Disponível em: <<http://cesta.cinted.ufrgs.br>>. Acesso em: 10 nov. 2015.
- CHEN, C.-M. **Intelligent web-based learning system with personalized learning path guidance.** Computers & Education (51), Elsevier, p. 787-814. 2008.
- CHIKH, Azeddine. **A general model of learning design objects.** Journal of King Saud University-Computer and Information Sciences, Vol. 26, Nº 1, Elsevier, p. 29-40, 2014.
- CHURCHILL, D. **Towards a useful classification of learning objects.** Educational Technology Research and Development, Vol. 55, Nº 5, Springer, p. 479-497, 2007.
- CLAROLINE. **Claroline Connect.** Consortium Claroline. Disponível em: <<http://www.claroline.net/>>. Acesso em: 10 nov. 2015.

- CONNECT. **Adobe Connect**. Adobe Systems Incorporated. Disponível em: <<http://www.adobe.com/products/adobeconnect.html>>. Acesso em: 10 nov. 2015.
- DA NÓBREGA, G.; DE SANTIAGO, R.; VAHLDICK, A. **Estudo sobre Padrões de Objetos de Aprendizagem para Compartilhamento na Rede LOP2P**. Anais do XXI SBIE, 2010.
- DA SILVA, E.; CAFÉ, L.; CATAPAN, A. **Os objetos educacionais, os metadados e os repositórios na sociedade da informação**. Ciência da Informação, Vol. 39, Nº. 3, p. 93-104, 2010.
- DAG, F.; ERKAN, K. **A Personalized Learning System Based on Semantic Web Technologies**. In: Intelligent Tutoring Systems in E-Learning Environments: Design, Implementation and Evaluation. IGI Global. p. 258-285, 2010.
- DALMON, D.; BRANDÃO, L.; BRANDÃO, A.; ISOTANI, S. **A domain engineering for interactive learning modules**. Journal of Research and Practice in Information Technology, Vol. 44, Nº 3, p. 309-326, 2012.
- DALMON, D.; SEIJI, I.; BRANDÃO, A.; BRANDÃO, L. **Work in progress – Enhancing Interactive Geometry Systems with intelligent tutoring features**. In: Frontiers in Education Conference (FIE). IEEE, p. F3E-1-F3E-2, 2011.
- DAM, H.; WINIKOFF, M. **Towards a next-generation AOSE methodology**. Science of Computer Programming, Vol. 78, Nº 6, p. 684-694, 2013.
- DE NIES, T.; SALLIAU, F.; VERBORGH, R.; MANNENS, E.; DE WALLE, R. **TinCan2PROV: Exposing Interoperable Provenance of Learning Processes through Experience API Logs**. Proceedings of the Linked Learning Workshop, International World Wide Web Conference Committee (IW3C2), p. 689-694, 2015.
- DEL MORAL, M.; CERNEA, A.; VILLALUSTRE, L. **Connectivist learning objects and learning styles**. Interdisciplinary Journal of E-Learning and Learning Objects, Vol. 9, p. 105-124, 2013.
- DESPOTOVIC-ZRAKIC, M.; MARKOVIC, A.; BOGDANOVIC, Z.; BARAC, D.; KRKO, S. **Providing Adaptivity in Moodle LMS Courses**. Educational Technology & Society, 15 (1), IFETS, p. 326-338, 2012.

- DETONI, D.; ARAUJO, R.; CECHINEL, C. **Predição de Reprovação de Alunos de Educação a Distância Utilizando Contagem de Interações**. Anais do XXV SBIE, 2014.
- DE-MARCOS, L.; PAGES, C.; MARTINEZ, J.; GUTIERREZ, J. **Competency-Based Learning Object Sequencing Using Particle Swarms**. In: 19th IEEE International Conference on Tools with Artificial Intelligence - ICTAI, Vol. 2, IEEE, p. 111-116, 2007.
- DIAS, C.; KEMCZINSKI, A.; LUCENA, S.; FERLIN, J.; HOUNSELL, M. **Padrões abertos: aplicabilidade em Objetos de Aprendizagem (OAs)**. Anais do XX SBIE, 2009.
- DIAZ, A.; FERNANDEZ, A.; CHIARADIA, M. **WikiLOR: A Collaborative Approach for Improving Learning Object Metadata**. Anais do X LACLO, 2015.
- DIBLEY, M; LI, H.; MILES, J.; REZGUI, Y. **The application of intelligent agency in a software model for buildings**. In: W. Tizani (Ed.), Proceedings of the International Conference on Computing in Civil and Building Engineering. Nottingham University Press, p. 205-212, 2010.
- DIBLEY, M.; LI, H.; MILES, J.; REZGUI, Y. **Towards intelligent agent based software for building related decision support**. Advanced Engineering Informatics, Vol. 25, N° 2. Elsevier, p. 311-329, 2011.
- DO PRADO, R.; BRANDÃO, L.; BRANDÃO, A. **iPlan: a model of lesson plans to help teaching process in Web learning environment**. In: Frontiers in Education Conference (FIE). IEEE, p. 1-7, 2014.
- DOUGIAMAS, M.; TAYLOR, P. **Moodle: Using learning communities to create an open source course management system**. In: World conference on educational multimedia, hypermedia and telecommunications. p. 171-178. 2003.
- DUBLIN CORE. **Dublin Core Metadata**. The Dublin Core Metadata Initiative. Disponível em: <<http://www.dublincore.org/>>. Acesso em: 10 nov. 2015.
- DUTRA, Renato Luís de Souza. **Encapsulamento e Utilização de Objetos de Aprendizagem Abertos SCORM para Ensinar a Avaliação Formativa**. Tese de Doutorado em Ciência da Computação. UFRGS. Porto Alegre, 2008.

- EXELEARNING. **El nuevo eXeLearning**. Instituto de Tecnologías Educativas – Gobierno de España. Disponível em: <<http://www.exelearning.net/>>. Acesso em: 10 nov. 2015.
- FASSBINDER, A.; DELAMARO, M.; BARBOSA, E. **Construção e Uso de MOOCs: Uma Revisão Sistemática**. Anais do XXV SBIE, 2014.
- FERREIRA L.; BARBOSA J.; GLUZ J. **Um modelo de Recomendação Ubíqua de Conteúdo para Grupos de Aprendizizes**. Anais do 2º CBIE, 2013, p. 679 – 706, 2013.
- FILATRO, A. **Design Instrucional Contextualizado – Educação e tecnologia**. 3ª Edição. Editora Senac São Paulo, 2010.
- FRADE, R.; NETO, F.; LIMA, R. W.; LIMA, R. M.; SILVA, L.; SOUZA, R. **Um Ambiente Virtual 3D Multiagente com Recomendação Personalizada de Objetos de Aprendizagem**. Anais do XXV SBIE, 2014.
- FRANTZ, C.; NOWOSTAWSKI, M.; PURVIS, M. **Augmenting android with AOSE principles for enhanced functionality reuse in mobile applications**. In: Advanced Agent Technology. Springer, p. 187-211, 2012.
- GAMMA, E.; HELM, R.; JOHNSON, R.; VLISSIDES, J. **Design Patterns: Elements of Reusable Object-Oriented Software**. Addison-Wesley, 1995.
- GASCUEÑA, J.; FERNANDEZ-CABALLERO, A.; GONZALEZ, P. **Domain Ontology for Personalized E-Learning in Educational Systems**. In: ICALT. p. 456-458, 2006.
- GHEBGHOUB, O.; ABEL, M.-H.; MOULIN, C. **Learning object indexing tool based on a LOM ontology**. In: Eighth IEEE International Conference on Advanced Learning Technologies, ICALT'08. IEEE. p. 576-578, 2008.
- GIUFFRA, P.; SILVEIRA, R. **A multi-agent system model to integrate Virtual Learning Environments and Intelligent Tutoring Systems**. International Journal of Interactive Multimedia and Artificial Intelligence, Vol. 2, No. 1, IJIMAI, p. 51-58, 2013.
- GLADUN, A.; ROGUSHINA, J.; GARCI, F.; MARTÍNEZ-BÉJAR, R.; FERNÁNDEZ-BREIS, J. **An application of intelligent techniques and semantic web technologies in e-learning environments**. In:



- Expert Systems with Applications, Vol. 36, Nº 2, Elsevier. p. 1922-1931, 2009.
- GLUZ, J.; VICARI, R. **Uma Ontologia OWL para Metadados IEEE-LOM, Dublin-Core e OBAA**. Anais do XXII SBIE, 2011.
- GOMES, Eduardo Rodrigues. **Objetos Inteligentes de Aprendizagem: uma abordagem baseada em agentes para objetos de aprendizagem**. Dissertação de Mestrado em Ciência da Computação. UFRGS. Porto Alegre, 2005.
- GONZALEZ-BARBONE, V.; ANIDO-RIFON, L. **From SCORM to Common Cartridge: A step forward**. Computers & Education, Vol. 54, Nº 1, Elsevier, p. 88-102, 2010.
- GUENAGA, M.; MECHACA, I.; ROMERO, S.; EGUÍLUZ, A. **A Tool to Evaluate the Level of Inclusion of Digital Learning Objects**, Procedia Computer Science, Vol. 14, Elsevier, p. 148-154, 2012.
- HINDRIKS, K. **Programming Rational Agents in GOAL**. In: Multi-Agent Programming: Languages, Tools and Applications. Cap. 4. Springer, p. 119-157, 2009.
- HOLMES, D.; STOCKING, R. **Augmenting agent knowledge bases with owl ontologies**. In: Aerospace conference. IEEE, p. 1-15, 2009.
- HORRIDGE, M.; BECHHOFER, S. **The OWL API: A Java API for OWL ontologies**. Semantic Web Journal, Vol. 2, Nº 1, p. 11-21, 2011.
- HSU, I.-C. **Intelligent discovery for learning objects using semantic web technologies**. Journal of Educational Technology & Society, Vol. 15, Nº 1, JSTOR, p. 298-312, 2012.
- HSU, I.-C. **Extensible access control mark-up language integrated with Semantic Web Technologies**. Information Sciences, Vol. 238, Elsevier, p. 33-51, 2013.
- HÜBNER, J. F.; BORDINI, R. H.; VIEIRA, R. **Introdução ao Desenvolvimento de Sistemas Multiagentes com Jason**. [S.I.], 2004. Disponível em: <<http://user.das.ufsc.br/~jomi/pubs/2004/Hubner-eriPR2004.pdf>>. Acesso em: 10 nov. 2015.
- HÜBNER, J.; SICHMAN, J. **Organização de sistemas multiagentes**. III Jornada de Mini-Cursos de Inteligência Artificial (JAIA'03), Vol. 8, p. 247-296, 2003.

- IEEE. **IEEE Std 1484.11.1**. IEEE Standard for Learning Technology — Data Model for Content to Learning Management System Communication. IEEE Computer Society, 2005.
- IEEE. **IEEE Std 1484.12.1**. IEEE Standard for Learning Object Metadata. IEEE Computer Society, 2002.
- IMS CP. **IMS Content Packaging**. IMS Global Learning Consortium. Disponível em: <<http://www.imsglobal.org/>>. Acesso em: 10 nov. 2015.
- IMS GLC. **IMS Common Cartridge Specification – Version 1.3**, IMS Global Learning Consortium, 2013. Disponível em: <<http://www.imsglobal.org/cc>>. Acesso em: 10 nov. 2015.
- IMS LTI. **IMS Learning Tools Interoperability**. IMS Global Learning Consortium. Disponível em: <<http://www.imsglobal.org/>>. Acesso em: 10 nov. 2015.
- ISO/IEC 25010:2011. **Systems and software Quality Requirements and Evaluation (SQuARE)**. System and software quality models, International Organization for Standardization, 2011.
- Jadex. **Jadex BDI Agent System**. Active Components. Disponível em: <<http://www.activecomponents.org/>>. Acesso em: 10 nov. 2015.
- JAIN, P.; HITZLER, P.; SHETH, A.; VERMA, K.; YEH, P. **Ontology alignment for linked open data**. In: The Semantic Web–ISWC 2010. Springer, p. 402-417, 2010.
- JAQUES, Patrícia Augustin. **Using an Animated Pedagogical Agent to Interact Affectively with the Student**. Tese de Doutorado em Ciência da Computação. UFRGS. Porto Alegre, 2004.
- JAQUES, P.; VICARI, R. **Estado da arte em ambientes inteligentes de aprendizagem que consideram a afetividade do aluno**. Revista Informática na Educação: Teoria & Prática, Vol. 8, Nº 1, p. 15-38, 2005.
- JORUM. **Jorum – The UK's largest repository for discovering and sharing Open Educational Resources for HE, FE and Skills**. JISC. Disponível em: <<http://www.jorum.ac.uk/>>. Acesso em: 10 nov. 2015.
- JURADO, F.; REDONDO, M. **Learning tools interoperability for enhancing a distributed personal learning environment with support for programming assignments**. International Symposium

- on Computers in Education (SIIE), IEEE, p. 87-92, 2014.
- KADHIM, M.; ALAM, M.; KAUR, H. **A Multi-intelligent Agent Architecture for Knowledge Extraction: Novel Approaches for Automatic Production Rules Extraction.** International Journal of Multimedia and Ubiquitous Engineering, Vol. 9, N° 2, p. 95-114, 2014.
- KARAGIANNIDIS, C.; SAMPSON, D. **Adaptation rules relating learning styles research and learning objects meta-data.** In: Workshop on Individual Differences in Adaptive Hypermedia. 3rd International Conference on Adaptive Hypermedia and Adaptive Web-based Systems (AH2004), 2004.
- KLAPISCAK, T.; BORDINI, R. **JASDL: A practical programming approach combining agent and semantic web technologies.** In: Declarative Agent Languages and Technologies VI. Springer, p. 91-110, 2009.
- KOMLENOV, Z.; BUDIMAC, Z.; IVANOVIC, M. **Introducing Adaptivity Features to a Regular Learning Management System to Support Creation of Advanced eLessons.** Informatics in Education, Vol. 9, No. 1, Institute of Mathematics and Informatics, p. 63-80, 2010.
- KOSTOLANYOVA, K. **Adaptation of Teaching Process Based on a Students Individual Learning Needs.** Journal on Efficiency and Responsibility in Education and Science, 2011.
- KRAVARI, K.; BASSILIADES, N. **A Survey of Agent Platforms.** Journal of Artificial Societies and Social Simulation, Vol. 18, N° 1, 2015. Disponível em: <<http://jasss.soc.surrey.ac.uk/18/1/11.html>>. Acesso em: 10 nov. 2015.
- KUMAR, S.; GANKOTIYA, A.; DUTTA, K. **A comparative study of moodle with other e-learning systems.** In: 3rd International Conference on Electronics Computer Technology (ICECT). IEEE, p. 414-418. 2011.
- KURILOVAS, E.; ZILINSKIENE, I.; DAGIENE, V. **Recommending suitable scenarios according to learners' preferences: An improved swarm based approach.** Computers in Human Behavior 30, Elsevier, p. 550-557, 2014.
- LEE, M.-C.; TSAI, K.; WANG, T. **A practical ontology query**

- expansion algorithm for semantic-aware learning objects retrieval.** Computers & Education, Vol. 50, Nº 4, Elsevier, p. 1240-1257, 2008.
- LEUNG, S.; Virwaney, S.; Lin, F.; Armstrong, A.; Dubbelboer, A. **TSI-Enhanced Pedagogical Agents to Engage Learners in Virtual Worlds.** International Journal of Distance Education Technologies, 2013.
- LUQUE, L., BRANDÃO, L.; TORI, R.; BRANDÃO, A. **Are You Seeing This? What Is Available and How Can We Include Blind Students in Virtual UML Learning Activities.** Anais do XXV SBIE. p. 204-213, 2014.
- MAHKAMEH, Y.; BAHREININEJAD, A. **A context-aware adaptive learning system using agents.** Expert Systems with Applications 38, Elsevier, p. 3280-3286, 2011.
- MATEUS, Gustavo Pereira. **Um Framework Baseado em Sistema Multiagentes e Lógica Fuzzy para Acompanhamento de Alunos em um Ambiente Virtual de Ensino-Aprendizagem.** Dissertação de Mestrado em Ciência da Computação. UFSC. Florianópolis, 2010.
- MATEUS, G.; WILGES, B.; NASSAR, S.; SILVEIRA, R. **Uma ferramenta de análise do desempenho de estudantes baseada em SMA e lógica Fuzzy.** Anais do XX SBIE, 2009.
- MCGREAL, R. **A typology of learning object repositories.** In: Handbook on information technologies for education and training. Springer, p. 5-28, 2008.
- MCGREAL, R. **Online Education Using Learning Objects.** Psychology Press, 2004.
- MELO, A. **Acessibilidade e Inclusão Digital em Contexto Educacional.** Anais da 3ª Jornada de Atualização em Informática na Educação (JAIE 2014), p. 1-41, 2014.
- MERLOT. **Multimedia Educational Resource for Learning and Online Teaching.** California State University. Disponível em: <<http://www.merlot.org/merlot/index.htm>>. Acesso em: 10 nov. 2015.
- MIKIC-FONTE, F.; BURGUILLO-RIAL, J.; LLAMAS-NISTAL, M.; FERNANDÉZ-HERMIDA, D. **A BDI-based intelligent tutoring module for the e-learning platform INES.** In: Frontiers in Education

- Conference (FIE). IEEE, p. T1J-1-T1J-6, 2010.
- MOODLE. **Modular Object-Oriented Dynamic Learning Environment**. Moodle Community. Disponível em: <<http://www.moodle.org/>>. Acesso em: 10 nov. 2015.
- MORRISON, M. **Head First JavaScript**. O'Reilly, 2008.
- MOURA, F.; FERNANDES, M. **Uma proposta de um modelo computacional que usa PSO para a escolha de Objetos de Aprendizagem baseado na Espiral de Kolb e nas Inteligências Múltiplas**. Anais do XXIII SBIE, 2012.
- NASCIMENTO, M.; BRANDÃO, L.; BRANDÃO, A. **A model to support a learning object repository for web-based courses**. In: 2013 IEEE Frontiers in Education Conference (FIE), Vol. 1, IEEE, p. 548-552. 2013.
- NASH, S. **Learning Objects, Learning Object Repositories, and Learning Theory: Preliminary Best Practices for Online Courses**. Interdisciplinary Journal of Knowledge and Learning Objects, Vol. 1, Nº 1, p. 217-228, 2005.
- NDLR. **National Digital Learning Resources**. NDLR Community. Disponível em: <<http://www.ndlr.ie/>>. Acesso em: 10 nov. 2015.
- NESSAH, D.; KAZAR, O. **An improved semantic information searching scheme based multi-agent system and an innovative similarity measure**. International Journal of Metadata, Semantics and Ontologies, Vol. 8, Nº 4, p. 282-297, 2013.
- NILSSON, M.; PALMÉR, M.; BRASE, J. **The LOM RDF binding – principles and implementation**. In 3rd Annual Ariadne Conference, 2003.
- OHLBACH, H. **Java2OWL – A System for Synchronising Java and OWL Version 1.1**. University of Munich, 2012.
- PADGHAM, L.; WINIKOFF, M. **Developing Intelligent Agent Systems - A practical guide**. John Wiley & Sons Ltd., 2004.
- PALOMINO, Cecília Estela Giuffra. **Modelo de Sistema Tutorial Inteligente para Ambientes Virtuais de Aprendizagem baseado em Agentes**. Dissertação de Mestrado em Ciência da Computação. UFSC. Florianópolis, 2013.
- PAPAZOGLAKIS, P. **The Past, Present and Future of SCORM**.

- Economy Informatics, Vol. 13 (1), INFOREC Association, p. 16-26, 2013.
- PARRISH, P. **The Trouble with Learning Objects**. Educational Technology Research and Development, Vol. 52, Nº 1, Springer, p. 49-67, 2004.
- PEREIRA, C.; CAMPOS, F.; STRÖELE, V.; DAVID, J.; BRAGA, R. **Extração de Características de Perfil e de Contexto em Redes Sociais para Recomendação de Recursos Educacionais**. Anais do XXV SBIE, 2014.
- PEREIRA, Luiz Antônio M.; PORTO, Fábio A. Machado; MELO, Rubens Nascimento. **Objetos de Aprendizado Reutilizáveis (RLOs): Conceitos, Padronização, Uso e Armazenamento**. Monografia Departamento de Informática, PUC-Rio, Rio de Janeiro 2003.
- PEREIRA, R.; LENTZ, C.; CATAPAN, A.; QUARTIERO, E.; GOMES, N.; CERNY, R. **Estudo de Softwares Educacionais**. EAD/UFSC/CED/CFM, Florianópolis, 2005.
- PETER, S.; BACON, E. **Adaptable, personalised e-learning incorporating learning styles**. Campus-Wide Information Systems, Vol. 27, Nº 2, Emerald, p. 91-100, 2010.
- POKAHR, A.; BRAUBACH, L. **The active components approach for distributed systems development**. International Journal of Parallel, Emergent and Distributed Systems, Vol. 28 (4), p. 321-369, 2013.
- POKAHR, A.; BRAUBACH, L.; HAUBECK, C.; LADIGES, J. **Programming BDI Agents with Pure Java**. Multiagent System Technologies, LNCS, Vol. 8732, Springer, p. 216-233, 2014.
- POKAHR, A.; BRAUBACH, L.; JANDER, K. **Unifying agent and component concepts**. In: Multiagent System Technologies. Springer, p. 100-112, 2010.
- POZZEBON, E.; FRIGO, L.; BITTENCOURT, G. **Inteligência artificial na educação universitária: quais as contribuições?** Revista do Centro de Ciências da Economia e Informática da Universidade da Região da Campanha Urcamp, Editora da URCAMP - EDIURCAMP, v. 8, n. 13, p. 34-41, 2004.
- PRIMO, Tiago Thompsen. **Método de Representação de Conhecimento Baseado em Ontologias para Apoiar Sistemas de**

- Recomendação Educacionais.** Tese de Doutorado em Ciência da Computação. UFRGS. Porto Alegre, 2013.
- PROMETHEUS DESIGN TOOL. **PDT Eclipse Plug-in.** Royal Melbourne Institute of Technology University. Disponível em: <<https://code.google.com/p/pdt-plugin/>>. Acesso em: 10 nov. 2015.
- RAMOS, E.; MENDONÇA, I. **O fundamental na avaliação da qualidade do software educacional.** Anais do II SBIE, SBC, Porto Alegre, p. 122-131, 1991.
- RANKINE, L.; STEVENSON, L.; MALFROY, J.; ASHFORD-ROWE, K. **Benchmarking across universities: A framework for LMS analysis.** In: Ascilite. p. 815-819. 2009.
- REINISCH, C; KOFLER, M.; KASTNER, W. **ThinkHome: A smart home as digital ecosystem.** In: Digital Ecosystems and Technologies (DEST), 2010 4th IEEE International Conference on. IEEE, p. 256-261, 2010.
- REIS, M.; PALADINI, E.; KHATOR, S.; NASSAR, S. **An Intelligent Tutorial System for Statistical Quality Control.** In: (CATE 2001) Fourth IASTED International Conference Computers and Advanced Technology in Education. 2001.
- RIOS, J. **Ontologias: alternativa para a representação do conhecimento explícito organizacional.** Anais do VI CINFOM – Encontro Nacional de Ciência da Informação, 2005.
- RITROVATO, P. **A methodology and an authoring tool for creating Complex Learning Objects to support interactive storytelling.** Computers in Human Behavior 31, Elsevier, p. 620–637, 2014.
- RIVED. **Rede Interativa Virtual de Educação.** SEED/MEC. Disponível em: <<http://rived.mec.gov.br/>>. Acesso em: 10 nov. 2015.
- ROBAA. **Repositório Digital OBAA.** Padrão de Metadados de Objetos de Aprendizagem. Disponível em: <<http://repositorio.portalobaa.org>>. Acesso em: 10 nov. 2015.
- ROCHA, A.; CAMPOS; G. **Avaliação da qualidade de software educacional.** In: Em Aberto, ano 12, nº 57, Brasília, p. 32-44, 1993. Disponível em: <<http://www.emaberto.inep.gov.br/index.php/emaberto/article/viewFile/845/757>>. Acesso em: 03 nov. 2015.

- RODRIGUES, A.; KONRATH, M.; TAROUÇO, L.; MEZZARI, A. **Autoria e empacotamento de conteúdos**. Novas Tecnologias na Educação, Vol. 7, Nº 3, CINTED-UFRGS, p. 1-11, 2009.
- RODRIGUES, J.; BRANDÃO, L.; NASCIMENTO, M.; RODRIGUES, P.; BRANDÃO, A.; GIROIRE, H.; AUZENDE, O. **iQuiz: integrated assessment environment to improve Moodle Quiz**. In: Frontiers in Education Conference, IEEE, p. 293-295, 2013.
- RUSSELL, S.; NORVIG, P. **Inteligência Artificial**. Elsevier (Campus), 2013.
- SABITHA, S.; MEHROTRA, D.; BANSAL, A. **A Data Mining Approach to Improve Re-Accessibility and Delivery of Learning Knowledge Objects**. Interdisciplinary Journal of E-Learning and Learning Objects, Vol. 10, p. 247-268, 2014.
- SALAZAR OSPINA, Oscar Mauricio. **Modelo de Sistema Multi-Agente ubicuo, adaptativo y sensible al contexto para ofrecer recomendaciones personalizadas de recursos educativos basado en ontologías**. Dissertação de Mestrado em Engenharia de Sistemas. Universidad Nacional de Colombia, Medellín. 2014.
- SALEENA, B.; SRIVATSA, S. **Designing a novel cross ontology measure to extract resources from knowledge base for an adaptive e-learning system**. International Journal of Metadata, Semantics and Ontologies, Vol. 9, Nº 3, p. 193-203, 2014.
- SANTOS, E.; BOFF, E.; VICARI, R. **Semantic Web Technologies Applied to Interoperability on an Educational Portal**. Intelligent Tutoring Systems, LNCS, Vol. 4053, Springer, p. 308-317, 2006.
- SANTOS, E.; VICARI, R.; COELHO, H. **Knowledge Acquisition and Intelligent Agency on the Web of Data**. In: New Trends in Artificial Intelligence. 14th Portuguese Conference on Artificial Intelligence - EPIA, p. 413-424, 2009.
- SANTOS, G.; JORGE, J. **Interoperable Intelligent Tutoring Systems as Open Educational Resources**. IEEE Transactions On Learning Technologies, Vol. 6, Nº 3, IEEE CS & ES, p. 271-282, 2013.
- SANTOS, P.; ALVES, R. **Metadados e Web Semântica para estruturação da Web 2.0 e Web 3.0**, DataGramZero - Revista de Ciência da Informação, Vol. 10, Nº 6. 2009. Disponível em: <[http://www.dgz.org.br/dez09/Art\\_04.htm](http://www.dgz.org.br/dez09/Art_04.htm)>. Acesso em: 10 nov.



2015.

- SANTOS, R.; LUZ, B.; MARTINS, V.; GUIMARÃES, M. **eTutor: Um Ambiente de Aprendizagem Interativo**. Anais do XXV SBIE, 2014.
- SCHONS, C.; RIBEIRO, A.; BATTISTI, P. **Educação a Distância: Web 2.0 na Construção do Conhecimento Coletivo**. 2008. Disponível em: <<http://repositorio.ufsc.br/handle/123456789/61430>>. Acesso em: 10 nov. 2015.
- SCORM 2004 – **Content Development Guide**, Version 1.04, The e-Learning Consortium, Japan, 2006.
- SILVA, E.; MENEZES, E. **Metodologia da pesquisa e elaboração de dissertação**. UFSC, 4ª Edição, 2005.
- SILVA, Júlia Marques Carvalho da. **Desenvolvimento de um Framework para Objetos Inteligentes de Aprendizagem Aderente a um Modelo de Referência para Construção de Conteúdos de Aprendizagem**. Dissertação de Mestrado em Ciência da Computação. UFSC. Florianópolis, 2007.
- SILVA, J.; BAVARESCO, N.; SILVEIRA, R. **Projeto e desenvolvimento de um Sistema Multi-agentes para Objetos Inteligentes de Aprendizagem baseado no padrão SCORM**. Revista Brasileira de Informática na Educação, Vol. 16, Nº 01, 2008.
- SILVA, T.; LIMA, R.; MESQUITA, H.; MARQUES, C.; COSTA, R.; ROCHA, S.; MEDEIROS, R. **OBA-MC: um modelo de Objeto de Aprendizagem centrado no processo de ensino-aprendizagem utilizando o padrão SCORM**. Anais do XXIII SBIE, 2012.
- SILVEIRA, Ricardo Azambuja. **Modelagem orientada a agentes aplicada a ambientes inteligentes distribuídos de ensino: JADE: Java Agent framework for Distance learning Environments**. Tese de Doutorado em Ciência da Computação, UFRGS. Porto Alegre, 2001.
- SILVEIRA, R.; GOMES, E.; VICARI, R. **Intelligent Learning Objects: An Agent-Based Approach of Learning Objects**. Information and Communication Technologies and Real-Life Learning, 2005.
- SILVEIRA, R.; GOMES, E.; VICARI, R. **Intelligent Learning Objects: An Agent-Based Approach of Learning Objects**. In: IFIP – International Federation For Information Processing, Vol. 182, Springer, p. 103-110, 2006.

- SKILLSOFT. **Skillsoft Elearning Content & Platforms**. Skillsoft Ireland Limited. Disponível em: <<http://www.skillsoft.com/products/default.asp>>. Acesso em: 10 nov. 2015.
- SOUZA, A.; ALVES, L. **Objetos digitais de aprendizagem e SCORM. Integração no Moodle**. In: L. Alves *et al.* (Eds.), Moodle: Estratégias Pedagógicas e Estudos de Caso, 2ª Edição, ALVES EDUNEB. 2012.
- STEVENSON, G.; DOBSON, S. **Sapphire: Generating Java runtime artefacts from OWL ontologies**. In: Advanced Information Systems Engineering Workshops. Springer. p. 425-436, 2011.
- SUN, H.; THANGARAJAH, J.; PADGHAM, L. **Eclipse-based prometheus design tool**. In: Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems. Vol. 1. International Foundation for Autonomous Agents and Multiagent Systems, p. 1769-1770, 2010.
- SUN, P.; TSAI, R.; FINGER, G.; CHEN, Y.; YEH, D. **What drives a successful e-Learning? An empirical investigation of the critical factors influencing learner satisfaction**. In: Computers & Education, Vol. 50. Nº 4, Elsevier, p. 1183-1202, 2008.
- SURESH, R.; DINAKARAN, K.; AMULYA, R. **Automating ontologies for e-learning**. International Journal of Metadata, Semantics and Ontologies, Vol. 9, Nº 3, p. 227-232, 2014.
- TANGARIFE, T. **A acessibilidade nos websites governamentais: um estudo de caso no site da Eletrobrás**. Dissertação de Mestrado em Artes e Design. PUC-Rio, Rio de Janeiro, p. 110-116, 2007.
- TINCAN. TinCan API, Rustici Software, LLC. Disponível em: <<https://tincanapi.com>>. Acesso em: 10 nov. 2015.
- TOMAIUOLO, M., TURCI, P., BERGENTI, F., & POGGI, A. **An ontology support for semantic aware agents**. In: Agent-Oriented Information Systems III. Vol. 3529. Springer. p. 140-153. 2006.
- UEZ, Daniela Maria. Método Prometheus AEOLus. 2014. Disponível em: <[http://www.uez.com.br/aeolus/docs/aeolus\\_11112014.pdf](http://www.uez.com.br/aeolus/docs/aeolus_11112014.pdf)>. Acesso em: 10 nov. 2015.
- UEZ, D.; HÜBNER, J. **Environments and organizations in multi-agent systems: From modelling to code**. In: Engineering Multi-Agent Systems. Springer. p. 181-203, 2014.

- VALDERRAMA, R.; CRUZ, A.; SALAZAR, A.; RAMÍREZ, J. **Personalized knowledge management in environments of web-based education.** International Journal of Information Technology & Decision Making, Vol. 12, Nº 2. World Scientific, p. 277-307, 2013.
- VAN SETERS, J.; OSSEVOORT, M.; TRAMPER, J.; GOEDHART, M. **The influence of student characteristics on the use of adaptive e-learning material,** Computers & Education, Vol. 58, Nº 3, Elsevier. p. 942-952, 2012.
- VESIN, B.; KLASNJA-MILICEVIC, A.; IVANOVIC, M.; BUDIMAC, Z. **Applying recommender systems and adaptive hypermedia for e-learning personalization.** Computing and Informatics, Vol. 32, Institute of Informatics, p. 629-659, 2013.
- VIAN, Jonas. **Sistema Multiagente para a Indexação e Recuperação Aplicado a Objetos de Aprendizagem.** Dissertação de Mestrado em Ciência da Computação. UFSC. Florianópolis, 2010.
- VICARI, R.; BEZ, M.; SILVA, J.; RIBEIRO, A.; GLUZ, J.; PASSERINO, L.; SANTOS, E.; PRIMO, T.; ROSSI, L.; BORDIGNON, A.; BEHAR, P.; FILHO, R.; ROESLER, V. **Proposta Brasileira de Metadados para Objetos de Aprendizagem Baseados em Agentes (OBAA),** RENOTE – Revista Novas Tecnologias na Educação, Vol. 8, Nº 2, CINTED – UFRGS, Porto Alegre, 2010.
- VICARI, R.; RIBEIRO, A.; SILVA, J.; SANTOS, E.; PRIMO, T.; BEZ, M. **Brazilian Proposal for Agent-Based Learning Objects Metadata Standard - OBAA,** In: Metadata and Semantic Research, CCIS, Vol. 108, Springer, p. 300-311, 2010.
- VICKERS, S. **IMS Learning Tools Interoperability. A Briefing Paper.** Centre for Educational Technology, Interoperability and Standards. JISC CETIS. 2012. Disponível em: <<http://publications.cetis.org.uk/wp-content/uploads/2012/05/LTI-Briefing-Paper.pdf>>. Acesso em: 10 nov. 2015.
- VIEIRA, P.; COSTA, A.; RAABE, A. **SCORM X Common Cartridge: Um Estudo Comparativo.** Anais do Computer on the Beach, p. 61-70, 2012.
- WANG, H.-C.; HSU, C.-W. **Teaching-Material Design Center: An ontology-based system for customizing reusable e-materials.** Computers & Education, Vol. 46, Nº 4, Elsevier, p. 458-470, 2006.

- WANG, S. **Ontology of learning objects repository for pedagogical knowledge sharing**. Interdisciplinary Journal of E-Learning and Learning Objects, Vol. 4, Nº 1, p. 1-12, 2008.
- WAZLAWICK, R. **Metodologia de Pesquisa para Ciência da Computação**. Elsevier, 2009.
- WEAVER, D.; SPRATT, C.; NAIR, C. **Academic and student use of a learning management system: Implications for quality**. Australasian Journal of Educational Technology, Vol. 24, Nº 1, 2008.
- WEBER, A.; SANTOS, E. **Educação Online em Tempos de Mobilidade e Aprendizagem Ubíqua: Desafios para as Práticas Pedagógicas na Cibercultura**. Revista EDaPECI, São Cristóvão (SE), v.13. n. 2, p. 168-183, 2013.
- WILEY, D. **Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy**. In: D. A. Wiley (Ed.), The instructional use of learning objects, 2000 [Online]. Disponível em: <<http://reusability.org/read/chapters/wiley.doc>>. Acessado em: 10 nov. 2015.
- WILGES, B.; MATEUS, G.; NASSAR, S.; BASTOS, R. **Integration of BDI agent with fuzzy logic in a virtual learning environment**. Latin America Transactions, IEEE (Revista IEEE America Latina), Vol. 10, Nº 1. IEEE, p. 1370-1376, 2012.
- WILSON, S.; CURRIER, S. **What is IMS Content Packaging?** CETIS Briefing Paper, CETIS, 2002.
- WINTER, J.; COTTON, D.; GAVIN, J.; YORKE, J. **Effective e-learning? Multi-tasking, distractions and boundary management by graduate students in an online environment**. Research in Learning Technology, Vol. 18, Nº 1, ALT-J, p. 71-83, 2010.
- WOOLDRIDGE, M. **An Introduction to MultiAgent Systems**. John Wiley & Sons, 2nd Edition, 2009.

## APÊNDICE A – Publicações

Com o intuito de divulgar os resultados obtidos durante o desenvolvimento da pesquisa, foram produzidos três artigos científicos, os quais foram submetidos, aceitos e publicados em eventos diferentes.

Inicialmente foi produzido um artigo relatando a fase inicial da pesquisa (AMORIM JR. *et al.*, 2015), o qual foi submetido ao *workshop* MASLE, do evento PAAMS 2015. Segundo a avaliação *Qualis* da CAPES/MEC, o PAAMS tem classificação EB4.

Os resultados preliminares da segunda fase da pesquisa foram divulgados através de um artigo resumido (AMORIM JR. e SILVEIRA, 2015) aceito no evento EPIA 2015, o qual tem classificação EB2 no *Qualis* da CAPES/MEC.

Por fim, foi publicado um artigo completo no XXVI SBIE, com a descrição da integração do modelo proposto com o LMS Moodle (AMORIM JR. e SILVEIRA, 2015). Destaca-se que o SBIE é o principal evento de Informática na Educação no Brasil, possuindo classificação EB2 no *Qualis* da CAPES/MEC. A 26ª edição do SBIE fez parte do CBIE & LACLO 2015.