

UNIVERSIDADE FEDERAL DE SANTA CATARINA

Jonas Vian

SISTEMA MULTIAGENTE PARA INDEXAÇÃO E  
RECUPERAÇÃO APLICADO A OBJETOS DE APRENDIZAGEM

Florianópolis  
2010



JONAS VIAN

SISTEMA MULTIAGENTE PARA INDEXAÇÃO E  
RECUPERAÇÃO APLICADO A OBJETOS DE APRENDIZAGEM

Dissertação submetida à Universidade  
Federal de Santa Catarina como parte dos  
requisitos para a obtenção do grau de  
Mestre em Ciência da Computação.

Orientador: Dr. Ricardo Azambuja Silveira

Florianópolis  
2010

Catálogo na fonte pela Biblioteca Universitária  
da  
Universidade Federal de Santa Catarina

V614s Vian, Jonas

Sistema multiagente para indexação e recuperação aplicado a objetos de aprendizagem [dissertação] / Jonas Vian ; orientador, Ricardo Azambuja Silveira. - Florianópolis, SC, 2010.

87 p.: il., grafs., tabs.

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

Inclui referências

1. Ciência da computação. 2. Ontologias (Sistema de recuperação da informação). I. Silveira, Ricardo Azambuja. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Ciência da Computação. III. Título.

CDU 681

Jonas Vian

SISTEMA MULTIAGENTE PARA INDEXAÇÃO E  
RECUPERAÇÃO APLICADO A OBJETOS DE APRENDIZAGEM

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, 25 de agosto de 2010.

---

Prof. Mario Antonio Ribeiro Dantas, Dr.  
Coordenador do Curso

**Banca Examinadora:**

---

Prof<sup>a</sup>. Rosa Maria Vicari, Dra.  
Universidade Federal do Rio Grande do Sul

---

Prof. Renato Fileto, Dr.  
Universidade Federal de Santa Catarina

---

Prof. Roberto Willrich, Dr.  
Universidade Federal de Santa Catarina



## **AGRADECIMENTOS**

Mais uma etapa vencida na vida acadêmica e digna de muitos agradecimentos a todos que apoiaram, participaram e colaboraram com este feito.

Meu agradecimento especial ao Prof. Ricardo pela oportunidade e orientação durante mais esta etapa de minha vida.

A todos os professores com quem convivi no mestrado, na graduação e também nos anos iniciais de meus estudos agradeço pela amizade e conhecimento transmitido. Especialmente aos professores que fizeram parte da banca.

A minha família agradeço de coração pelo apoio e incentivo. Obrigado Pai e Mãe.

Aos amigos de republica, de laboratório e de trabalho agradeço pelo apoio e companheirismo.

Enfim agradeço de coração a todos que participaram e apoiaram na conquista deste título.





*“Há mais pessoas que desistem do que  
pessoas que fracassam.”*

Henry Ford



## RESUMO

VIAN, Jonas. **Sistema multiagente para indexação e recuperação aplicado a objetos de aprendizagem**. Florianópolis, 2010. Dissertação (Mestrado em Ciência da Computação) – Universidade Federal de Santa Catarina, Florianópolis, 2010.

Em *E-learning* é necessária a produção de conteúdos de aprendizagem para disponibilização através da web, tornando assim o processo de ensino e aprendizagem flexível. Porém, a produção de conteúdos para *E-learning* possui alto custo e, devido a isso, é importante a reutilização. Para prover reutilização de conteúdos de aprendizagem, surgiu o conceito de Objetos de Aprendizagem, onde um conteúdo de aprendizagem é devidamente descrito por metadados e armazenado em um repositório. Contudo, a heterogeneidade de padrões e especificações na área, dificulta a reutilização e compartilhamento de Objetos de Aprendizagem. Assim, a recuperação de Objetos de Aprendizagem, para reuso de forma compartilhada é um problema dinâmico, devido ao grande volume de pesquisas, e necessita de sistemas que permitam interoperabilidade entre repositórios heterogêneos e distribuídos, onde se espera boa precisão.

O problema de recuperação de Objetos de Aprendizagem é um exemplo da crescente demanda por sistemas mais complexos e inteligentes. Esta demanda possibilitou o surgimento do paradigma de desenvolvimento de sistemas orientado a agentes. Este paradigma permite a exploração de habilidades sociais em agentes de software e tem o objetivo de buscar soluções mais inteligentes para problemas computacionais complexos. Este trabalho propõe um sistema multiagente para recuperação e indexação de Objetos de Aprendizagem, armazenados em repositórios distintos e descritos com diferentes padrões de metadados. Com objetivo de melhorar a precisão e cobertura na recuperação de Objetos de Aprendizagem, são utilizadas técnicas de recuperação de informação e ontologias, em apoio ao sistema multiagente.

**Palavras-chave:** Objetos de Aprendizagem, Sistemas Multiagente, Ontologias, Interoperabilidade, Recuperação de Informação.

## ABSTRACT

For e-learning is necessary to produce learning content for delivery on the web, to make the teaching and learning process more flexible. However, production of learning content for e-learning has high cost and because of this it is important to promote the reuse.

Aiming to provide the reuse of learning content arose the concept of learning objects. Learning object consists of a learning content properly described by metadata and stored in a learning object repository. However, the heterogeneity of standards and specifications in the area of learning objects, make difficult to reuse and share learning objects. Thus the recovery of learning objects for reuse and sharing is considered a dynamic problem due to the large volume of research and requiring systems to enable interoperability among heterogeneous and distributed repositories and where it expect good accuracy.

The problem of learning objects retrieval is an example of complex and intelligent systems. This need has enabled the emergence of the agents-oriented programming paradigm. This paradigm allows the exploration of social skills in software agents and aims to get more intelligent solutions to complex computational problems.

This work proposes a multiagent system for indexing and retrieval learning objects stored in different repositories and described with different metadata standards. Aiming to improve accuracy and coverage in the recovery of learning objects are used information retrieval techniques and ontologies together the multiagent system.

**Keywords:** Learning Objects, Multiagent Systems, Ontologies, Interoperability, Information Retrieval.

## LISTA DE ILUSTRAÇÕES

Figura 1:	Granularidade de objetos de aprendizagem .....	26
Figura 2:	Elementos do padrão de metadados LOM .....	29
Figura 3:	Agente.....	39
Figura 4:	FIPA Request Interaction Protocol .....	46
Figura 5:	Ciclo de vida de agentes, definido pela FIPA.....	48
Figura 6:	Interface gráfica do Jade – RMA .....	49
Figura 7:	Metamodelo O-MaSE .....	52
Figura 8:	Estrutura do conjunto de métodos O-MaSE .....	53
Figura 9:	Esquema que representa a arquitetura do sistema proposto .....	63
Figura 10:	Diagrama de agentes.....	64
Figura 11:	Fragmento XML do índice gerado pelo agente indexador .....	65
Figura 12:	Diagrama de protocolo.....	67
Figura 13:	Interface de Busca.....	69
Figura 14:	Interface de Resultado .....	70
Figura 15:	Ontologia para o domínio de Segurança da Informação.....	71
Figura 16:	Resultado comparativo .....	76

## LISTA DE TABELAS

Tabela 1:	Parâmetros para mensagens FIPA ACL .....	42
Tabela 2:	Atos performativos da linguagem FIPA ACL .....	43
Tabela 3:	Protocolos de comunicação definidos pela FIPA .....	44
Tabela 4:	Exemplo de heterogeneidade entre elementos dos padrões de metadados .....	61
Tabela 5:	Resultado quantitativo das consultas realizadas no repositório LUME .....	73
Tabela 6:	Resultado quantitativo das consultas realizadas no repositório CESTA .....	74
Tabela 7:	Resultado quantitativo das consultas realizadas através do protótipo implementado no trabalho .....	75

## LISTA DE ABREVIATURAS E SIGLAS

ACL	<i>Agent Communication Language</i>
ADL	<i>Advanced Distributed Learning</i>
AICC	<i>Aviation Industry CBT (Computer-Based Training) Committee</i>
AMD	<i>Advanced Micro Devices</i>
AMS	<i>Agent Manager System</i>
API	<i>Application Programming Interface</i>
AOSE	<i>Agent Oriented Software Engineering</i>
AUML	<i>Agent-based Unified Modeling Language</i>
BIOE	<i>Banco Internacional de Objetos Educacionais</i>
BDI	<i>Belief, Desire, Intention</i>
CESTA	<i>Coletânea de Entidades de Suporte ao uso de Tecnologia na Aprendizagem</i>
CINTED	<i>Centro Interdisciplinar de Novas Tecnologias na Educação</i>
CORDRA	<i>Content Object Repository Discovery and Registration / Resolution Architecture</i>
DC	<i>Dublin Core</i>
DCMI	<i>Dublin Core Metadata Initiative</i>
DF	<i>Directory Facilitator</i>
EAD	<i>Educação a Distância</i>
EUA	<i>Estados Unidos da América</i>
FEB	<i>Federação Educa Brasil</i>
FEDORA	<i>Flexible Extensible Digital Object Repository Architecture</i>
FIPA	<i>Foundation for Intelligent Physical Agents</i>
HP	<i>Hewlett-Packard</i>
GB	<i>Gigabyte</i>
GHz	<i>Gigahertz</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IA	<i>Inteligência Artificial</i>
IIOIP	<i>Internet Inter-Orb Protocol</i>
IMTP	<i>Internal Message Transport Protocol</i>
IEEE	<i>Institute of Electrical and Electronics Engineers</i>
IMS	<i>Instructional Management Systems</i>
IMS LRM	<i>IMS Learning Resource Metadata</i>
ISO	<i>International Organization for Standardization</i>
JADE	<i>Java Agent Development Framework</i>
JDOM	<i>Java Document Object Model</i>
JLDAP	<i>Java Lightweight Directory Access Protocol</i>

JSP	<i>Java Server Page</i>
KQML	<i>Knowledge Query Manipulation Language</i>
LMS	<i>Learning Management System</i>
LO	<i>Learning Objects</i>
LOM	<i>Learning Object Metadata</i>
LOR	<i>Learning Object Repository</i>
LTSC	<i>Learning Technology Standards Committee</i>
LDAP	<i>Lightweight Directory Access Protocol</i>
MAS	<i>Multiagent System</i>
MaSE	<i>Multiagent Software Engineering</i>
MEC	<i>Ministério da Educação e Cultura</i>
MERLOT	<i>Multimedia Educational Resource for Learning and Online Teaching</i>
MIT	<i>Massachusetts Institute of Technology</i>
MTP	<i>Message Transport Protocol</i>
OA(s)	<i>Objeto(s) de Aprendizagem</i>
OAI-PMH	<i>Open Archives Initiative Protocol for Metadata Harvesting</i>
OBAA	<i>Agent Based Learning Object</i>
O-MaSE	<i>Organization Multi-agent Software Engineering</i>
OWL	<i>Web Ontology Language</i>
PPGIE	<i>Programa de Pós Graduação em Informática na Educação</i>
RDF	<i>Resource Description Framework</i>
REST	<i>Representational State Transfer</i>
RI	<i>Recuperação de Informação</i>
RIVED	<i>Rede Interativa Virtual de Educação</i>
RMA	<i>Remote Management Agent</i>
RNP	<i>Rede Nacional de Ensino e Pesquisa</i>
SCORM	<i>Sharable Content Object Reference Model</i>
SEED	<i>Secretaria de Educação a Distância</i>
SMA	<i>Sistema Multiagente</i>
SOAP	<i>Simple Object Access Protocol</i>
SPARQL	<i>SPARQL Protocol and RDF Query Language</i>
TCP/IP	<i>Transmission Control Protocol / Internet Protocol</i>
TF/IDF	<i>Term Frequency / Inverse Document Frequency</i>
TIC	<i>Tecnologia de informação e comunicação</i>
UFRGS	<i>Universidade Federal do Rio Grande do Sul</i>
UK	<i>United Kingdom</i>
URL	<i>Uniform Resource Locator</i>
XML	<i>Extensible Markup Language</i>
XMLRPC	<i>XML Remote Procedure Call</i>
W3C	<i>World Wide Web Consortium</i>



## SUMÁRIO

1	<b>INTRODUÇÃO</b> .....	<b>19</b>
1.1	PROBLEMATIZAÇÃO .....	20
1.2	OBJETIVOS .....	21
1.2.1	<b>Objetivo Geral</b> .....	<b>21</b>
1.2.2	<b>Objetivos Específicos</b> .....	<b>21</b>
1.3	METODOLOGIA .....	22
1.4	ESTRUTURA DO TRABALHO .....	23
2	<b>OBJETOS DE APRENDIZAGEM</b> .....	<b>25</b>
2.1	PADRÕES DE METADADOS .....	27
2.1.1	<b>LOM IEEE</b> .....	<b>28</b>
2.1.2	<b>Dublin Core</b> .....	<b>29</b>
2.1.3	<b>IMS LRM</b> .....	<b>30</b>
2.1.4	<b>ISO MRL</b> .....	<b>30</b>
2.1.5	<b>AICC</b> .....	<b>30</b>
2.1.6	<b>OBAA</b> .....	<b>31</b>
2.2	REPOSITÓRIOS PARA OBJETOS DE APRENDIZAGEM .....	31
2.2.1	<b>CESTA</b> .....	<b>32</b>
2.2.2	<b>LUME</b> .....	<b>32</b>
2.2.3	<b>BIOE</b> .....	<b>32</b>
2.2.4	<b>RIVED</b> .....	<b>33</b>
2.2.5	<b>MERLOT</b> .....	<b>33</b>
2.2.6	<b>FEB</b> .....	<b>34</b>
2.3	TECNOLOGIAS PARA CRIAÇÃO DE REPOSITÓRIOS .....	34
2.3.1	<b>Sistema de Banco de Dados</b> .....	<b>35</b>
2.3.2	<b>Servidor LDAP</b> .....	<b>35</b>
2.3.3	<b>XML</b> .....	<b>35</b>
2.3.4	<b>DSpace</b> .....	<b>36</b>
2.3.5	<b>Fedora</b> .....	<b>36</b>
3	<b>AGENTES</b> .....	<b>39</b>
3.1	SISTEMAS MULTI-AGENTES (SMAS) .....	40
3.1.1	<b>Comunicação entre Agentes</b> .....	<b>41</b>
3.2	PADRÃO FIPA.....	41
3.2.1	<b>Linguagem FIPA ACL</b> .....	<b>42</b>
3.2.1.1	Atos performativos FIPA .....	43
3.2.1.2	Protocolos de Interação FIPA .....	44
3.3	AMBIENTES PARA DESENVOLVIMENTO DE SISTEMAS MULTIAGENTES .....	46

3.3.1	<b>JADE</b> .....	<b>47</b>
3.4	MODELAGEM E ESPECIFICAÇÃO DE SMA .....	50
3.4.1	<b>Engenharia de Software Orientada a Agentes</b> .....	<b>50</b>
3.4.1.1	Organization Multi-agent Software Engineering.....	51
4	<b>RECUPERAÇÃO DE INFORMAÇÃO</b> .....	<b>55</b>
4.1	INDEXAÇÃO.....	56
4.1.1	<b>Técnicas para auxiliar a indexação</b> .....	<b>57</b>
4.1.2	<b>Método para determinação de pesos</b> .....	<b>57</b>
4.2	WEB SEMÂNTICA .....	58
4.2.1	<b>Ontologias</b> .....	<b>59</b>
5	<b>SISTEMA PROPOSTO</b> .....	<b>61</b>
5.1	ARQUITETURA.....	61
5.2	SISTEMA MULTIAGENTE.....	63
5.2.1	<b>Agente Indexador</b> .....	<b>64</b>
5.2.2	<b>Agente Recuperador</b> .....	<b>66</b>
5.2.3	<b>Protocolo</b> .....	<b>67</b>
5.2.4	<b>Ciclo de Vida dos agentes no SMA</b> .....	<b>68</b>
5.3	DESENVOLVIMENTO DO PROTÓTIPO.....	68
6	<b>VALIDAÇÃO DO PROTÓTIPO</b> .....	<b>71</b>
6.1	REPOSITÓRIOS INDEXADOS .....	72
6.2	TESTES .....	73
6.3	RESULTADOS .....	76
7	<b>CONSIDERAÇÕES FINAIS</b> .....	<b>79</b>
7.1	TRABALHOS FUTUROS .....	80
8	<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	<b>81</b>

# 1 INTRODUÇÃO

A produção de conteúdos de aprendizagem, para utilização em processos de ensino e treinamento, demanda esforço e investimento por parte de educadores e instituições de ensino, principalmente na produção de conteúdos para utilização em educação a distância – EAD, onde a informação deve estar totalmente explícita no conteúdo (DOWNES, 2001). Com o advento da internet o conceito de educação a distância tornou-se mais expressivo, incluindo também o conceito de *e-learning*. *E-learning* é como ficou conhecido o processo de ensino e treinamento apoiado por tecnologias de informação e comunicação (TIC), e que proporciona flexibilidade e interatividade ao processo.

Devido ao custo envolvido na produção de conteúdos de aprendizagem para *e-learning*, existe forte interesse na possibilidade de reutilização de conteúdos já produzidos. Por isso, a possibilidade de facilitar a reutilização de conteúdos de aprendizagem tem despertado o interesse de grupos de pesquisas, organizações e instituições de ensino ao redor do mundo.

Com objetivo de permitir a organização e reutilização de conteúdos de aprendizagem, surge o conceito de objetos de aprendizagem (*Learning Object* - LO), (MCGREAL, 2004). Pesquisas em torno deste conceito originaram diversos padrões e especificações que norteiam a produção e o armazenamento de OAs. Porém não é possível destacar, no estado da arte atual, um padrão que seja amplamente aceito, dado que cada padrão ou especificação tem suas particularidades, isso justifica a heterogeneidade existente nas tecnologias utilizadas para a produção e o armazenamento de OAs, bem como na definição da granularidade dos objetos. Esta heterogeneidade dificulta a interoperabilidade entre aplicações que produzem e que armazenam OAs e as que tentam recuperá-los, dificultando assim a reutilização de OAs já produzidos. O problema de reutilização de OAs também é dificultado pela diversidade e pela heterogeneidade dos repositórios de objetos de aprendizagem (*Learning Object Repository* – LOR) existentes na web, criados com objetivo de armazenar OAs descritos por metadados (DOWNES, 2001).

Após esta análise percebe-se que o cenário de recuperação de OAs é heterogêneo, devido aos diferentes padrões e especificações existentes para a produção e armazenamento; distribuído, dado à existência de inúmeros repositórios espalhados pela web e é também dinâmico devido ao volume de pesquisas na área e a amplitude do campo de utilização deste conceito. Além disso, percebe-se que a falta

de ferramentas de busca eficientes e especializadas em OAs não permite uma ampla reutilização de objetos já produzidos.

Neste tipo de cenário, se justifica a utilização da tecnologia de sistemas multiagente – SMA, para construção de ferramentas inteligentes de busca, que facilitem a recuperação de OAs existentes e disponíveis em repositórios, proporcionando a reutilização dos mesmos em ambientes de ensino e aprendizagem. A tecnologia de SMA é baseada no desenvolvimento de componentes de software que possuam capacidade de adaptação ao ambiente, de agir de forma autônoma, de cooperar e de se comunicar, a fim de atingir um objetivo comum (WOOLDRIDGE, 2002).

De acordo com este cenário, o presente trabalho de pesquisa visa propor um modelo de busca inteligente, capaz de indexar e recuperar OAs, independente do padrão de metadados utilizado na sua descrição, localizados em repositórios distintos, utilizando a tecnologia de sistemas multiagente e a ideia de ontologias, com intuito de proporcionar maior reutilização de OAs.

## 1.1 PROBLEMATIZAÇÃO

A partir da análise do estado da arte da tecnologia de OAs identificou-se a sua expressividade e também as limitações existentes para atingir o objetivo da reutilização. As limitações ocorrem principalmente pela falta de padronização consolidada, visto que é um conceito relativamente novo e também a falta de ferramentas que facilitem a recuperação de OAs.

As ferramentas de recuperação de informações baseadas apenas na busca sintática possuem limitações, pois consideram a informação apenas em sua forma textual, desconsiderando o significado da mesma. Com isso não são eficientes na recuperação de OAs, visto a necessidade de maior precisão. Na web a informação não está catalogada como nos repositórios de OAs, dificultando assim a utilização de aspectos semânticos por ferramentas de recuperação de informação. Já na recuperação de OAs onde existe uma descrição mais formal do conteúdo através de metadados e também onde cada objeto tem sua própria finalidade, a utilização de aspectos semânticos da informação proporciona recursos mais precisos para a recuperação.

Frente à análise dos processos de elaboração e armazenamento de OAs, percebeu-se a possibilidade de propor um modelo de busca e recuperação de OAs que permita a recuperação de forma federada conforme especificado por (CORDRA, 2006) e (FEB, 2010),

recuperando objetos em repositórios distintos e heterogêneos, descritos com diferentes padrões de metadados, além de considerar aspectos semânticos de conteúdos de aprendizagem.

Este modelo pretende minimizar o problema de reutilização e compartilhamento de OAs, causado pela heterogeneidade de padrões de metadados, pela diferença de tecnologia e forma de armazenamento dos repositórios que estão distribuídos na web.

## 1.2 OBJETIVOS

### 1.2.1 Objetivo Geral

Propor um modelo de indexação e busca de OAs hospedados em repositórios heterogêneos e distribuídos, descritos com diferentes padrões de metadados, baseado no paradigma de agentes inteligentes e no emprego de sistemas ontológicos para descrição do domínio na semântica dos mecanismos de busca.

### 1.2.2 Objetivos Específicos

- Analisar os padrões de metadados existentes, utilizados para descrição de OAs.
- Analisar as principais arquiteturas utilizadas na implementação de repositórios de OAs.
- Desenvolver um modelo de busca baseado em resolução semântica obtida a partir de bases ontológicas de descrição de domínios de conteúdo de aprendizagem.
- Desenvolver modelo de agentes capazes de indexar OAs independente da forma de armazenamento destes e do padrão de metadados utilizado na descrição.
- Desenvolver a arquitetura dos agentes capazes de recuperar OAs relacionados a temas solicitados por usuários, utilizando ontologias para envolver aspectos semânticos relacionados à solicitação.
- Implementar o protótipo de uma solução para o problema de recuperação de OAs, baseada no paradigma de agentes que funcione na web.
- Validar a proposta realizando comparações entre a ferramenta de busca nativa de repositórios de OAs existentes e o protótipo desenvolvido.

### 1.3 METODOLOGIA

A metodologia do trabalho envolve inicialmente a revisão bibliográfica dos conceitos envolvidos na pesquisa e levantamento do estado da arte, com a finalidade de adquirir o conhecimento necessário para compreensão do problema abordado e a proposta de solução. A revisão bibliográfica abrange os conceitos inerentes aos OAs e a sua aplicação em e-learning. Com esta revisão foi possível compreender o conceito e os padrões de especificação que norteiam a utilização de OAs. Após a familiarização com os conceitos relacionados aos OAs e e-learning, que fortaleceram as características do problema abordado pela pesquisa, realizou-se a revisão bibliográfica inerente a tecnologia de sistemas multiagente, sobre a qual já existia um embasamento e que a colocava como preferível na busca pela solução do problema envolvido. Esta revisão permitiu reforçar os conceitos relacionados aos sistemas multiagente e também sobre os objetivos e a aplicabilidade desta tecnologia na solução de problemas, tendo foco na comparação com tecnologias que possuam características similares, com objetivo de justificar a escolha da tecnologia de sistemas multiagente.

Durante o estudo percebeu-se que somente a tecnologia de sistemas multiagente não é suficiente para resolver todos os problemas envolvidos na busca e recuperação de OAs em repositórios heterogêneos. A partir disso, foi investigada a utilização de técnicas de Web Semântica e Recuperação de Informação, que em combinação com a tecnologia de sistemas multiagente, possa resolver a dificuldade de recuperação de OAs de forma distribuída, em ambiente heterogêneo e utilizando aspectos semânticos dos OAs, com o objetivo de proporcionar maior reutilização dos mesmos, em ambientes de ensino e aprendizagem suportados por tecnologia. Esta investigação resulta no domínio da tecnologia de representação de ontologias cuja utilização tem o objetivo de contextualizar um domínio e os conceitos de indexação e atribuição de ponderação à informação utilizada na indexação, para melhorar a precisão e a cobertura da recuperação, melhorando também o desempenho de ferramentas de busca.

A partir do domínio do conhecimento envolvido, foi analisada a perspectiva das ferramentas e das metodologias que permitam a proposição de um modelo de busca e da implementação de um protótipo capaz de demonstrar a validade deste modelo. Esta busca resultou em uma metodologia de modelagem para sistemas multiagente e um *framework* para desenvolvimento de sistemas multiagente que segue padrões determinados para a tecnologia. O *framework* prevê a

implementação das características básicas de um sistema multiagente com as funcionalidades necessárias para a implementação do modelo proposto.

A metodologia escolhida permitiu a criação do modelo do sistema multiagente proposto e a utilização do *framework* permitiu a construção de um protótipo do sistema, utilizado para validar a aplicabilidade da proposta no cenário de recuperação de OAs.

Para validação da proposta, são utilizados para teste do protótipo implementado, OAs registrados em repositórios reais, já consolidados e contendo bom numero de objetos. Estes repositórios foram disponibilizados para a realização deste trabalho, através de parceria com grupo de pesquisa de outra universidade. Com a realização dos testes comparativos entre a recuperação proporcionada pelo protótipo do sistema proposto e as ferramentas de recuperação nativas dos repositórios envolvidos na validação, conseguiu-se os resultados que validam a proposta e estão descritos neste trabalho.

#### 1.4 ESTRUTURA DO TRABALHO

O trabalho esta dividido em sete capítulos: 1 Introdução, 2 Objetos de Aprendizagem, 3 Agentes, 4 Recuperação de Informações e Web Semântica, 5 Sistema proposto, 6 Validação da proposta e 7 Conclusões, sendo que na introdução são apresentadas as motivações e justificativas, bem como os objetivos desta dissertação.

Nos capítulos 2, 3 e 4 são apresentados os conceitos necessários para o desenvolvimento do trabalho de pesquisa. No capítulo 2 são apresentados o conceito e os objetivos de OAs, bem como os principais padrões de metadados utilizados para descrição de OAs, com as particularidades de cada um. Também são descritos alguns dos repositórios existentes e suas características juntamente com a apresentação de algumas das tecnologias empregadas na criação de repositórios. No capítulo 3 são apresentados os conceitos inerentes as tecnologias de agentes e sistemas multiagente como: padrões, ambientes de desenvolvimento e metodologias de modelagem relacionados a estas tecnologias. No capítulo 4 são apresentados os conceitos relacionados a tecnologias e técnicas utilizadas para Recuperação de Informação de forma automatizada, bem como os conceitos da Web Semântica que permitem otimizar a Recuperação de Informação.

O sistema proposto é descrito no capítulo 5 onde são apresentadas as tecnologias utilizadas no desenvolvimento bem como o

funcionamento da solução proposta, que une os conceitos descritos nos capítulos anteriores.

No capítulo 6 é apresentada a aplicação do protótipo desenvolvido em um cenário real, com objetivo de validar a solução proposta para o problema de pesquisa, bem como os resultados obtidos. No capítulo 7 são apresentadas as considerações finais sobre os resultados, dificuldades e sucessos ocorridos durante as etapas da pesquisa.



## 2 OBJETOS DE APRENDIZAGEM

Objetos de aprendizagem são recursos educacionais que podem ser empregados no processo de aprendizagem suportado por tecnologia (MCGREAL, 2004). A IEEE (IEEE-LTSC, 2005) define OAs como: “todo material digital ou não que possa ser utilizado, reutilizado e referenciado no processo de ensino aprendido apoiado por tecnologia”.

Segundo (MCGREAL, 2004) podem-se destacar algumas características que OAs devem possuir:

- **Acessibilidade:** devem estar disponível para acesso em qualquer local.
- **Interoperabilidade:** podem ser desenvolvidos em locais diferentes, utilizando plataformas diferentes.
- **Adaptabilidade:** devem permitir a adaptação para atender necessidades específicas.
- **Reusabilidade:** podem ser utilizados em várias aplicações
- **Durabilidade:** a utilização deve ser permitida, mesmo quando ocorram mudanças tecnológicas, sem a necessidade de modificações.
- **Recuperabilidade:** devem ser localizados e recuperados facilmente a partir de busca por termos, e quando e onde for necessário.
- **Avaliabilidade:** efetividade pedagógica, preço e usabilidade podem ser avaliados.
- **Intercambialidade:** um objeto pode ser facilmente substituído por outro.

OAs podem ter como base: textos, animações, apresentações, imagens, softwares, etc. e para serem localizados e reutilizados é preciso que sejam descritos por uma especificação de metadados (dados sobre os dados), (DOWNES, 2001). O conteúdo de aprendizagem devidamente descrito por metadados, forma o OA, que deve ser armazenado em um repositório e pode ser combinado com outros OAs para construção de objetos maiores como lições e cursos (NASH, 2005).

O repositório de OAs tem a finalidade de armazenar e organizar os OAs permitindo que estes sejam localizados e recuperados.

O conceito de OAs esta intimamente ligado ao conceito de programação orientada a objeto (DOWNES, 2001), cuja ideia central é a definição de entidades como objetos, e estes uma vez definidos podem

ser instanciados e utilizados no desenvolvimento de software conforme a necessidade.

A propriedade fundamental para permitir a reutilização de OAs é a granularidade, que determina a possibilidade de combinação entre os objetos (WILEY, 2000).

Não existe uma definição de qual seria a granularidade perfeita de OAs, mas defende-se que OAs devem possuir um grau interno de contextualização que possibilite a combinação entre si (WILEY, 2000). A figura 1 exemplifica a granularidade de OAs e como estes podem ser combinados formando estruturas mais complexas.

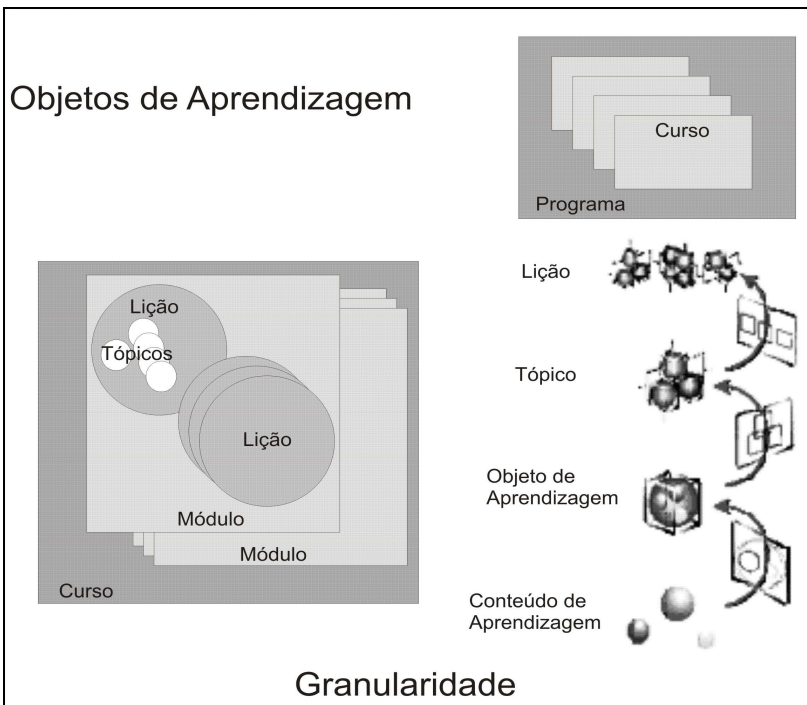


Figura 1: Granularidade de objetos de aprendizagem

Fonte: Adaptado de (MCGREAL, 2004)

Outro componente importante relacionado ao conceito de OA é o sistema gerenciador de aprendizagem (LMS – Learning Management System), cujo objetivo é gerenciar o processo de aprendizagem e uma das funcionalidades é possibilitar a utilização de OAs no contexto de aprendizagem em que esta sendo empregado.

Durante os últimos anos, grupos de pesquisas de organizações renomadas têm pesquisado e proposto padrões e especificações para OAs. De acordo com (CURRIER, 2008) é possível elencar algumas organizações que vêm realizando pesquisas, com objetivo de desenvolver: conjuntos de metadados para descrição de OAs, orientações para organização e armazenamento de metadados de OAs em estrutura XML (Extensible Markup Language) e modelos de referencia para construção de OAs considerando aspectos tecnológicos, como SCORM (Sharable Content Object Reference Model). A seguir são destacadas organizações que possuem significativo esforço empregado nas áreas citas:

- Ariadne Foundation, (ARIADNE, 2006)
- AICC (Aviation Industry CBT (Computer-Based Training) Committee), (AICC, 2010)
- Dublin Core Metadata Initiative, (DCMI, 2010)
- IEEE-LTSC (Institute of Electrical and Electronics Engineers - Learning Technology Standards Committee), (IEEE-LTSC, 2005)
- IMS (Instructional Management Systems) Global Learning Consortium, (IMS, 2010)
- ISO (International Organization for Standardization), (ISO, 2010)
- ADL (Advanced Distributed Learning), (ADL, 2010)
- OBAA-UFRGS, (OBAA, 2010)

Além da definição de especificações e padrões muitas instituições, principalmente universidades tem trabalhado e pesquisado em torno do desenvolvimento de tecnologias e padrões para criação de repositórios, juntamente com a criação de OAs para utilização institucional que são armazenados nos próprios repositórios desenvolvidos.

A seguir são expostos mais detalhes sobre estes padrões e especificações para OAs.

## 2.1 PADRÕES DE METADADOS

Metadados (dados sobre dados) no âmbito de OAs é o conjunto mínimo de elementos necessários para que um OA possa ser gerenciado, localizado e avaliado (IEEE-LTSC, 2005), e tem o objetivo de contextualizar um conteúdo de aprendizagem. Um padrão de metadados define as informações que permitem um OA ser encontrado e reutilizado, também define a obrigatoriedade e o domínio dos elementos

de metadados do padrão, porém a qualidade da descrição dos elementos para um OA não é controlada pelo padrão de metadados. A quantidade e a organização dos elementos definidos por um padrão de metadados variam de acordo com a finalidade do padrão, desta forma existe a possibilidade de um mesmo conteúdo de aprendizagem ser descrito por mais de um padrão de metadados.

A descrição dos metadados de um OA, dado um padrão, não necessariamente precisa estar fisicamente junto ao conteúdo e podem ser armazenados de diferentes formas. A descrição de metadados pode ser realizada diretamente pelo autor do OA e também existe a possibilidade do autor utilizar ferramentas de autoria disponível no mercado, que fazem a geração automática dos metadados. A seguir são detalhados alguns dos principais padrões de metadados.

### 2.1.1 LOM IEEE

LOM é um padrão definido pela IEEE, como IEEE 1484.12.1-2002 *Draft Standard for Learning Object Metadata* (IEEE-LTSC, 2005), e especifica a sintaxe e a semântica dos elementos definidos pelo padrão, para a descrição adequada de um conteúdo de aprendizagem.

O LOM é uma iniciativa voltada a descrição de conteúdos de aprendizagem e teve seu desenvolvimento iniciado a partir de uma proposta de padrão de metadados, desenvolvida pelo IMS Global Learning Consortium em parceria com Ariadne Foundation e submetida a IEEE, sendo aprovada em 2002, como padrão LOM pelo grupo IEEE-LTSC (IEEE-LTSC, 2005).

O padrão consiste de 76 elementos divididos em 9 categorias: *general, life cycle, meta-metadata, educational, technical, educational, rights, relation, annotation, e classification*. Na figura 2 é possível visualizar a distribuição dos elementos em cada categoria.

O padrão LOM é utilizado como base em outras iniciativas como UK LOM Core e CanCore, sendo que também é recomendado como padrão de metadados para OAs criados de acordo com o modelo SCORM (JISC-CETIS, 2008; CURRIER, 2008).

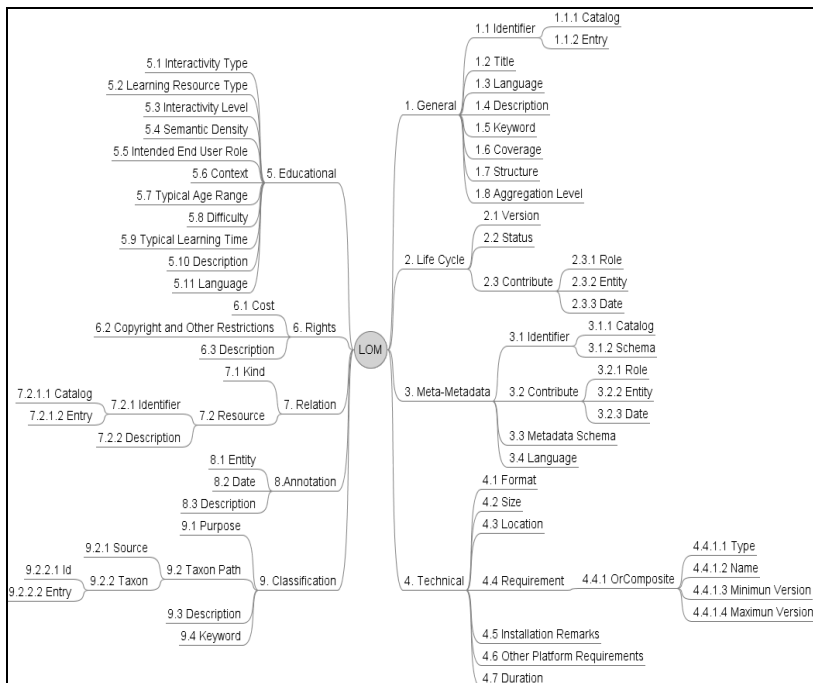


Figura 2: Elementos do padrão de metadados LOM

Fonte: Adaptado de (BARKER, 2005).

### 2.1.2 Dublin Core

Padrão desenvolvido e mantido por Dublin Core Metadata Initiative (DCMI), que é considerada uma das primeiras iniciativas envolvidas na especificação de metadados, utilizados para descrição de conteúdos em geral, na web. DCMI é uma organização aberta e engajada no desenvolvimento de padrões de metadados interoperáveis e de propósito geral. O DCMI mantém comunidades, grupos de trabalhos e promove eventos com objetivo de tornar amplamente aceita a utilização de padrões de metadados (DCMI, 2010).

O padrão possui 15 elementos que são: *contributor*, *coverage*, *creator*, *date*, *description*, *format*, *identifier*, *language*, *publisher*, *relation*, *rights*, *source*, *subject*, *title*, *type*.

O DCMI através do grupo de trabalho DC-Education Application Profile e em parceria com o grupo de trabalho da IEEE que mantém o padrão LOM, vem estudando o desenvolvimento de metadados voltados para descrição de conteúdos de aprendizagem (DCMI, 2010).

### 2.1.3 IMS LRM

Padrão de metadados antecessor do LOM, desenvolvido por iniciativa do projeto IMS. Projeto esse iniciado pelo EDUCOM, um consórcio sem fins lucrativos, formado por instituições de ensino superior dos Estados Unidos da América e seus parceiros, que tinha objetivo de desenvolver padrões abertos e de mercado para aprendizagem online, incluindo a especificação de metadados para conteúdos de aprendizagem (IMS, 2010).

O trabalho conjunto entre o projeto IMS e o projeto europeu Ariadne resultou no envio da proposta baseada no IMS LRM a IEEE resultando no desenvolvimento do padrão LOM (IMS, 2010).

Após a aprovação do padrão LOM pela IEEE, o padrão IMS LRM foi descontinuado pela IMS, que continuou trabalhando em orientações e boas práticas de utilização do padrão LOM.

### 2.1.4 ISO MRL

Padrão sendo desenvolvido por iniciativa da ISO através do comitê JTC1/SC36 (ISO, 2010), que tem trabalhado a partir do padrão LOM com objetivo de melhorá-lo, adicionando a este capacidade de trabalhar com múltiplas línguas e aspectos de acessibilidade (CURRIER, 2008). As pesquisas desenvolvidas pelo comitê buscam o desenvolvimento de um padrão que seja aceito internacionalmente e que seja compatível com padrões já existentes como LOM e Dublin Core.

O objetivo do padrão é definir regras para descrição de metadados que facilitem a busca, aquisição, avaliação e utilização de conteúdos de aprendizagem. Por alunos, professores ou processos automatizados (ISO, 2010).

O padrão consiste de duas partes:

- **Framework:** define as regras para descrição dos elementos de metadados, como formatos e vocabulários.
- **Conjunto de elementos:** define um conjunto de 72 elementos que compõe o padrão e são utilizados para descrever um conteúdo de aprendizagem.

### 2.1.5 AICC

Padrão desenvolvido pela AICC (AICC, 2010), uma associação internacional dedicada a auxiliar a comunidade de treinamento em

aviação, usufruir das tecnologias existentes. Esta organização tem objetivo de desenvolver normas, orientações e boas práticas para treinamento na área de aviação, considerando as tecnologias de treinamento relacionadas. A AICC mantém com acesso público as suas especificações.

O Padrão desenvolvido pela AICC para descrever conteúdos de aprendizagem toma como base o padrão LOM e adiciona extensões necessárias, dado o domínio em que é empregado (AICC, 2010).

### 2.1.6 OBAA

Especificação desenvolvida na Universidade Federal do Rio Grande do Sul (UFRGS), como parte do projeto OBAA e com apoio do Ministério da Educação e Cultura (MEC). O projeto tem objetivo de desenvolver a especificação para OAs interoperáveis entre diferentes plataformas tecnológicas, como Internet, TV digital e tecnologias móveis. Para atender o objetivo aborda a construção de OAs juntamente com a tecnologia de sistemas multiagente (OBAA, 2010, VICARI, 2010).

O OBAA tem como base o padrão LOM, assumindo extensões que permitem descrever aspectos referentes às diferentes tecnologias onde o conteúdo de aprendizagem pode ser utilizado. Também contém elementos para descrever os OAs com foco em acessibilidade e especificidades do sistema educacional brasileiro (BEZ, et al., 2009).

## 2.2 REPOSITÓRIOS PARA OBJETOS DE APRENDIZAGEM

Repositório de objetos de aprendizagem tem a finalidade de armazenar, organizar e permitir que OAs sejam encontrados e utilizados. O OA após descrito por metadados é armazenado em um repositório, que deve prover a interface para submissão, avaliação e recuperação do objeto (DOWNES, 2001; NASH, 2005).

A recuperação é feita através de ferramentas de busca, sendo que os objetos recuperados de um repositório podem ser utilizados diretamente por alunos, por professores na autoria de cursos ou pelo LMS.

A seguir são apresentados alguns repositórios estudados neste trabalho, juntamente com suas características.

### 2.2.1 CESTA

O projeto CESTA - Coletânea de Entidades de Suporte ao uso de Tecnologia na Aprendizagem foi idealizado com objetivo de sistematizar e organizar o registro dos objetos educacionais, desenvolvidos pela equipe do Programa de Pós-Graduação de Informática na Educação (PPGIE) e do CINTED - Centro Interdisciplinar de Novas Tecnologias na Educação da UFRGS, para utilização em cursos de capacitação e pós-graduação na modalidade de Educação a Distância (CESTA, 2010).

A descrição dos OAs no repositório CESTA é feita seguindo a padronização IEEE-P1484, que se refere ao padrão de metadados LOM.

A arquitetura do repositório é montada através de servidor LDAP (Lightweight Directory Access Protocol). LDAP é um protocolo aberto de acesso on-line a diretórios.

O repositório contém aproximadamente 420 OAs, em variados domínios (CESTA, 2010).

### 2.2.2 LUME

Lume é o repositório digital da Universidade Federal do Rio Grande do Sul, e reúne os documentos digitais gerados na universidade, visando sua preservação e divulgação. Tem o objetivo de reunir, preservar, divulgar e garantir o acesso confiável e permanente aos documentos acadêmicos, científicos, artísticos e administrativos gerados no âmbito da universidade, bem como às suas coleções históricas, e a outros documentos de relevância para a instituição (LUME, 2010).

A arquitetura do repositório Lume é baseada no software DSpace, software este utilizado para construção de repositórios digitais que permite o armazenamento e a disponibilização de documentos digitais.

O Lume possui aproximadamente 23000 documentos armazenados e organizados de acordo com o domínio abrangido pelos mesmos.

### 2.2.3 BIOE

O Banco Internacional de Objetos Educacionais (BIOE) é um repositório criado em 2008 com iniciativa do Ministério da Educação e tem o propósito de manter e compartilhar recursos educacionais digitais de livre acesso, em diferentes formatos e adequados para a realidade da comunidade educacional brasileira.



Possui aproximadamente 10000 OAs e assim como o Lume utiliza o software DSpace como base para o repositório e o padrão de metadados Dublin Core para descrição dos objetos armazenados (BIOE, 2008).

#### 2.2.4 RIVED

O RIVED (Rede Interativa Virtual de Educação) é um programa que tem por objetivo a produção de conteúdos pedagógicos digitais, na forma de OAs, bem como o armazenamento destes. Inicialmente foi uma iniciativa governamental através da Secretaria de Educação a Distância (SEED) e atualmente é mantido por universidades (RIVED, 2010).

O RIVED não disponibiliza interface para recuperação dos OAs, além do seu próprio sistema de busca disponível no site. Isto dificulta o acesso ao repositório por outros sistemas de busca. Também não disponibiliza informações referentes ao padrão de metadados utilizados na descrição dos objetos e a quantidade de OAs armazenados no repositório.

#### 2.2.5 MERLOT

O repositório MERLOT (Multimedia Educational Resource for Learning and Online Teaching) é uma iniciativa da universidade estadual da Califórnia-EUA, direcionado principalmente para faculdades e estudantes de ensino superior, sendo de acesso livre e aberto. Possui aproximadamente 25000 OAs, (MERLOT, 2010).

O repositório armazena além dos próprios conteúdos de aprendizagem, comentários e experiências sobre a utilização destes, juntamente com exercícios relacionados, que tem o objetivo de melhorar a experiência de ensino dos OAs. O MERLOT provê ferramentas para a construção de conteúdos de aprendizagem e permite a organização de OAs em coleções pessoais, o que possibilita a organização de cursos dentro do próprio repositório, utilizando os objetos existentes. Os OAs incluídos no repositório, bem como as coleções pessoais podem ser classificados em 11 tipos diferentes de conteúdo, definidos pelo repositório (MERLOT, 2010).

Assim como o RIVED, o MERLOT também não disponibiliza interface para recuperação dos OAs, além do seu próprio sistema de busca disponível no site e também não disponibiliza informações

referentes ao padrão de metadados utilizados na descrição dos objetos armazenados no repositório.

### 2.2.6 FEB

Projeto desenvolvido na UFRGS com objetivo de criar uma estrutura para federação de repositórios já existentes, onde a busca de OAs armazenados em diferentes repositórios seja realizada através de um ponto único. Esta estrutura permite o cadastro e administração dos repositórios pertencentes a federação e além disso disponibiliza a ferramenta de busca para recuperação dos objetos pertencentes aos repositórios envolvidos (FEB, 2010).

A Federação Educa Brasil (FEB) é implementada sobre uma arquitetura de diretórios LDAP e está em conformidade com a especificação CORDRA (CORDRA, 2006). A base de funcionamento consiste em traduzir a descrição de metadados dos OAs armazenados em repositórios existentes, e descritos com o padrão de metadados nativo de cada repositório, para o padrão OBAA.

O projeto é apoiado pela Rede Nacional de Ensino e Pesquisa (RNP) e está vinculado ao projeto OBAA (OBAA, 2010). O resultado do projeto tem objetivo de tornar-se uma ferramenta de recuperação de OAs, para utilização no âmbito educacional brasileiro.

## 2.3 TECNOLOGIAS PARA CRIAÇÃO DE REPOSITÓRIOS

Como visto anteriormente, um OA é composto por um conteúdo que possua informação utilizável para aprendizagem, descrito por um padrão de metadados que permita este conteúdo ser localizado e recuperado. A descrição de metadados de um OA necessita ser armazenada de forma organizada e que permita acesso fácil e rápido, para que assim os metadados atendam seu objetivo.

A seguir são apresentadas algumas tecnologias que podem ser utilizadas para criação de repositórios de objetos de aprendizagem. A escolha de uma tecnologia de armazenamento de metadados, em detrimento a outra é feita basicamente pelo volume de informação que será armazenada e a forma de acesso aos metadados que se pretende utilizar. Para exemplificar também são descritos alguns sistemas existentes, que permitem a criação de repositório de OAs e são referencia no cenário atual de repositórios digitais. Em (JISC-CETIS, 2006) e (MARTINS, et al., 2008) são listados vários sistemas que

permitem o armazenamento, manutenção e recuperação de conteúdos digitais. Estes sistemas podem ser comerciais ou de acesso livre.

### 2.3.1 Sistema de Banco de Dados

Tecnologia que pode ser utilizada na criação de repositórios, porém não é específica para isto, demandando assim esforço de programação sobre o sistema de Banco de Dados, para que possa atender as necessidades de um repositório de OAs.

Consiste de softwares que permitem o gerenciamento de informação, através da definição da estrutura de armazenamento e mecanismo de manipulação de informações (SILBERSCHATZ, et al., 1999). Possuem a vantagem de serem robustos e confiáveis no armazenamento de grande volume de informações, além de ser uma tecnologia amplamente utilizada e com grande ferramental de apoio a utilização. Desvantagem é a necessidade de conhecimentos especializados para instalação, configuração e manutenção.

### 2.3.2 Servidor LDAP

LDAP é um protocolo aberto de acesso a diretório que funciona sobre o protocolo TCP/IP. Foi concebido como alternativa ao modelo X.500, sendo uma simplificação deste (CARTER, 2003).

Um serviço LDAP consiste em uma árvore de nós, onde cada nó possui um conjunto de atributos com seus respectivos valores e é capaz de organizar e controlar o acesso a dados armazenados em uma hierarquia de diretórios.

Para armazenamento de OAs em um servidor LDAP, cada nó da árvore é considerado um OA e os atributos do nó correspondem aos elementos de metadados, de acordo com o padrão utilizado para o repositório. Neste caso o conteúdo de aprendizagem que compõe o objeto, não está armazenado na estrutura do LDAP.

### 2.3.3 XML

Extensible Markup Language (XML) é um linguagem de marcação idealizada e recomendada pela World Wide Web Consortium (W3C). É utilizada para geração de arquivos que possuem tanto estrutura física como lógica e que permite o armazenamento de informação de forma flexível, através de *tags* de marcação (BRAY, et al., 2008).

Fisicamente um documento XML é composto por entidades que são identificadas através de *tags* de marcação, sendo que uma entidade pode conter subentidades, de forma que um documento XML sempre possui uma *tag* raiz. Já logicamente o documento possui declarações, elementos, comentários, referências e instruções de processamento que são identificadas através das *tags* utilizadas para marcação (BRAY, et al., 2008). Dado um documento XML é possível, através de mecanismos da própria linguagem, definir se o documento esta bem formado ou não, bem como tratar restrições ao layout deste.

Documentos XML facilitam a recuperação e a interpretação de informação, e também flexibilizam a forma de armazenamento para metadados de OAs, através da definição de *tags* que correspondem aos elementos do padrão de metadados. Por outro lado, dependendo do volume de informação e a frequência de acesso, a utilização de documentos XML pode diminuir a performance do sistema que os manipula.

#### 2.3.4 DSpace

DSpace é um sistema cliente-servidor, utilizado para criação de repositórios digitais e permite o armazenamento, indexação, manutenção e recuperação de conteúdos em formato digital. DSpace é um sistema de acesso livre e código aberto desenvolvido pelo MIT e pela HP (Hewlett Packard) e amplamente utilizado para armazenamento de materiais digitais, no âmbito de instituições de ensino (DSpace, 2010). O DSpace é compatível com protocolo OAI-PMH (Open Archives Initiative Protocol for Metadata Harvesting), utilizado para disponibilização dos metadados referente a OAs (OAI, 2010).

OAI-PMH é um protocolo desenvolvido pela OAI e utilizado para recuperação de metadados, os quais descrevem os registros armazenados em um repositório e também disponibilizá-los em formato XML via requisições HTTP. O protocolo suporta a recuperação de metadados através do padrão Dublin Core, mas pode ser estendido para elementos de outros padrões de metadados (OAI, 2010). A utilização deste protocolo facilita a recuperação de OAs, bem como o desenvolvimento de ferramentas de busca.

#### 2.3.5 Fedora

Fedora (Flexible Extensible Digital Object Repository Architecture) é um sistema que permite a criação de repositórios, de

propósito geral, bem como disponibilização destes na forma de serviços. Esta tecnologia foi desenvolvida numa parceria entre a universidade da Virgínia e a universidade de Cornell.

O objetivo do projeto Fedora é fornecer software aberto, para criação de repositório que possam servir a diversos tipos de sistemas de gestão de informação. Para atender o objetivo, toda funcionalidade do Fedora é disponibilizada através de serviços web, com interfaces REST (Representational State Transfer) e SOAP (Simple Object Access Protocol) (FEDORA, 2010). Fedora também é compatível com protocolo OAI-PMH para disponibilização de metadados.

A arquitetura do Fedora prove a possibilidade de realacionamento entre objetos armazenados em repositórios distribuídos. Também possibilita o armazenamento de referência a objetos armazenados em outros sistemas.

O Fedora define um modelo de objeto digital que permite a criação de objetos a partir de qualquer tipo de conteúdo digital, inclusive OAs. Sendo que este modelo oferece as vantagens de abstração, flexibilidade, generalização, agregação e expansão (FEDORA, 2010).



### 3 AGENTES

Inteligência Artificial (IA) é a área da ciência da computação que visa estudar e propor técnicas de solução, para problemas computacionais complexos, utilizando como base a experiência e o comportamento humano. Com o objetivo de criar mecanismos para construção de sistemas mais inteligentes, que possam solucionar problemas de forma parecida com a que humanos fariam, a IA tem apresentado várias técnicas, entre as quais podem ser destacadas: sistemas especialistas, redes neurais, algoritmos genéticos e agentes inteligentes, sendo esta ultima o foco deste trabalho.

Um agente, segundo (WOOLDRIDGE, 2002) é um sistema computacional que está situado em um ambiente, e que é capaz de realizar ações autônomas neste ambiente de forma a atingir os objetivos projetados. Já (BRADSHAW, 1997) define agente como sendo uma entidade de software, que funciona de forma contínua e autônoma em um ambiente em particular, geralmente habitado por outros agentes, e que é capaz de intervir em seu ambiente de forma flexível e inteligente, sem requerer intervenção ou orientação humana constante. De um modo ideal, funcionando continuamente por longos períodos de tempo.

Agentes são capazes de perceber o ambiente onde habitam através de sensores e a partir disso tomar decisões a serem realizadas no ambiente através de atuadores, como exemplifica a figura 3. Isto dá aos agentes o poder de racionalidade, que é desenvolvida a partir da interpretação das percepções do ambiente, raciocínio sobre seus objetivos e execução de ações que venham a maximizar a sua medida de desempenho (RUSSEL, et al., 2004).

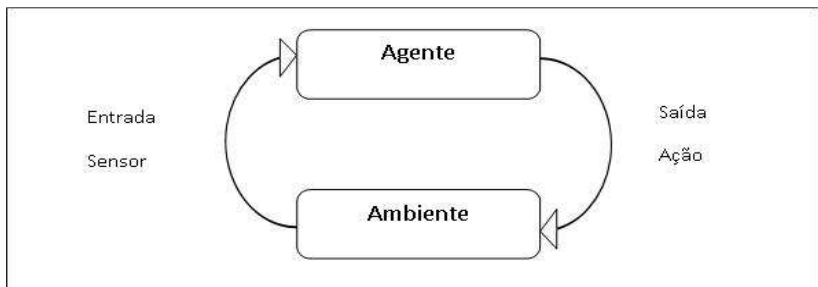


Figura 3: Agente

Fonte: Adaptado de (WOOLDRIDGE, 2002).

Um agente pode também possuir capacidade de aprender com as experiências, de cooperar, de negociar e de se comunicar com outros agentes, além de mobilidade. Segundo (WOOLDRIDGE, 2002), para um agente ser considerado inteligente deve possuir as capacidades de perceber e reagir a mudanças no ambiente onde habita, de agir por iniciativa própria e ter habilidades sociais que permitam a interação com outros agentes (inclusive humanos), a fim de satisfazer seus objetivos.

Quanto à arquitetura um agente pode ser:

- **Reativo:** agentes que funcionam em modo estímulo-resposta, sem a existência de memória dos estados e acontecimentos passados.
- **Cognitivo:** agentes que possuem estado mental, permitindo que o raciocínio do agente leve em conta as informações já vivenciadas no passado. Nesta arquitetura tem destaque os agentes BDI (belief-desire-intention) que possuem como estado mental uma base de Crenças, Desejos e Intenções que é atualizada durante o ciclo de vida do agente e utilizada para a tomada de decisões e ações.
- **Híbrido:** arquitetura que mistura características reativas e cognitivas.

Outro fator que influencia diretamente no comportamento de um agente é o tipo do ambiente onde habita, sendo que os ambientes possíveis podem ser (RUSSEL, et al., 2004):

- Completamente observável x Parcialmente observável
- Determinístico x Estocástico
- Episódico x Sequencial
- Estático x Dinâmico
- Discreto x Contínuo
- Agente único x Multiagente

### 3.1 SISTEMAS MULTI-AGENTES (SMAS)

É crescente a demanda por softwares que atendam as necessidades: de autonomia, de adaptação a ambientes dinâmicos e interação entre aplicações diferentes. Com objetivo de permitir o desenvolvimento destes requisitos e prover soluções otimizadas, para problemas computacionais surgiu a tecnologia de sistemas multiagente.

Um sistema multiagente segundo (WOOLDRIDGE, 2002) é uma rede fracamente acoplada de solucionadores de problemas que trabalham em conjunto, para resolver problemas que vão além da capacidade individual de cada um. Estes solucionadores de problemas



são softwares essencialmente autônomos, distribuídos e muitas vezes heterogêneos em sua natureza.

Sistemas multiagente é uma subárea da inteligência artificial distribuída que visa suprir deficiências da inteligência artificial clássica, defendendo que a inteligência surge não apenas em cada indivíduo de uma sociedade, mas sim da interação entre os indivíduos (JENNINGS, 1996). Agentes em um sistema multiagente devem desempenhar as seguintes habilidades: cooperação, resolução de conflitos, negociação, comprometimento, interação e comunicação (WOOLDRIDGE, 2002). Sendo a comunicação uma característica chave para a tecnologia de sistemas multiagente.

### 3.1.1 Comunicação entre Agentes

Para que haja comunicação é fundamental a existência de um meio de comunicação, que no caso de agentes permita a troca de mensagens entre os agentes, utilizando uma linguagem comum denominada ACL (Agent Communication Language). As linguagens desenvolvidas para comunicação de agentes têm sido diretamente influenciadas, pela teoria dos atos de fala. Elas permitem além da transmissão de informação, expressar a intenção da informação contida em uma mensagem. Uma ACL tem o objetivo de definir a sintaxe e a semântica para a comunicação de agentes em um sistema multiagente, independentemente da tecnologia utilizada na implementação dos agentes.

O mecanismo de comunicação é imprescindível para que agentes de software, em um sistema multiagente, possam exercer suas habilidades de comunicação, negociação e coordenação. Entre as linguagens desenvolvidas para comunicação de agentes podem ser destacadas FIPA ACL (Foundation for Intelligent Physical Agents, Agent Communication Language) e KQML (Knowledge Query and Manipulation Language), (WOOLDRIDGE, 2002).

## 3.2 PADRÃO FIPA

FIPA é uma organização ligada a IEEE Computer Society, que promove especificações e padrões para tecnologia de agentes e sistemas multiagente, bem como a interoperabilidade da tecnologia de agentes com outras tecnologias (FIPA, 2010). FIPA foi aceita em 2005, como comitê de padrões IEEE. As especificações FIPA representam um conjunto de padrões designados a promover o desenvolvimento de

agentes. As especificações têm um ciclo de vida que possui os seguintes estados: preliminar, experimental, padrão, abandonado ou obsoleto.

FIPA é mantida por organizações internacionais que colaboram abertamente e têm o interesse de aprimorar e divulgar a tecnologia de sistemas multiagente, sendo as especificações desenvolvidas em seu âmbito, de livre acesso.

### 3.2.1 Linguagem FIPA ACL

Para existir comunicação, entre agentes deve existir um entendimento semântico bem definido e comum aos participantes do diálogo, ou seja, é difícil falar em comunicação e interoperabilidade sem existir padrões. Neste sentido a FIPA desenvolveu a especificação de uma linguagem para comunicação de agentes chamada FIPA ACL, que visa padronizar a comunicação entre agentes.

A estrutura de mensagens definida pela linguagem é descrita no documento “*FIPA ACL Message Structure Specification - SC00061G (2002)*”, sendo que esta estrutura define os parâmetros que são obrigatórios e opcionais para composição de uma mensagem FIPA ACL, bem como a finalidade de cada parâmetro. A tabela 1 apresenta os parâmetros definidos no SC00061G e a categoria a que cada parâmetro pertence (FIPA, 2010).

A comunicação através da linguagem FIPA ACL baseia-se na composição e troca de mensagens entre os agentes em um ambiente, independente das particularidades de cada agente, como plataforma operacional e tecnologia de rede.

<b>Parâmetro</b>	<b>Categoria de Parâmetros</b>
Performative	Tipo de ato comunicativo
Sender	Participantes da comunicação
Receiver	
Reply-to	
Content	Conteúdo da mensagem
Language	Descrição do Conteúdo
Encode	
Ontology	
Protocol	Controle da Conversação
Conversation-id	
Reply-with	
In-reply-to	

<b>Parâmetro</b>	<b>Categoria de Parâmetros</b>
Reply-by	

Tabela 1: Parâmetros para mensagens FIPA ACL

Fonte: Adaptado de (FIPA, 2010) – SC00061G

### 3.2.1.1 Atos performativos FIPA

Os atos performativos da linguagem FIPA ACL, têm como base os atos performativos da linguagem humana e são fundamentais na comunicação de agentes. É através do ato performativo, informado como parâmetro da mensagem, que o agente interpreta o objetivo de uma mensagem.

Os atos performativos definidos pela linguagem FIPA ACL são descritos no documento “*FIPA Communicative Act Library Specification - SC00037J (2002)*”. A tabela 2 apresenta os atos performativos da linguagem FIPA ACL.

<b>Ato Performativo</b>	<b>Finalidade</b>
Accept Proposal	Indicar a aceitação de proposta recebida previamente para execução de ação.
Agree	Indicar a concordância com a execução de ação
Cancel	Indicar o cancelamento da execução de ação
Call for Proposal	Efetuar consulta de proposta para execução de ação
Confirm	Indicar que determinada proposição é verdadeira
Disconfirm	Indicar que determinada proposição é falsa
Failure	Indicar falha na execução de ação
Inform	Informar que uma proposição é verdadeira
Inform If	Informar que uma proposição pode ser verdadeira ou não.
Inform Ref	Informar ao receptor um objeto através do descritor deste
Not Understood	Informar o receptor que não entendeu uma ação solicitada
Propagate	Indicar que a mensagem deve ser propagada pelos receptores
Propose	Informar uma proposta para execução de ação
Proxy	Informar o receptor que deve repassar a mensagem

<b>Ato Performativo</b>	<b>Finalidade</b>
	a outros agentes que se encaixem na descrição da mensagem
Query If	Perguntar se uma proposição é verdadeira ou não
Query Ref	Perguntar se um objeto é válido
Refuse	Indicar a recusa em executar uma ação
Reject Proposal	Informar a rejeição a proposta de execução de ação
Request	Solicitar a execução de ação
Request When	Solicitar a execução de ação quando determinada condição for verdadeira
Request Whenever	Solicitar a execução de ação sempre que determinada condição for verdadeira.
Subscribe	Solicitar a informação do valor de dada referencia e que seja informado sempre que haja alteração no objeto referenciado

Tabela 2: Atos performativos da linguagem FIPA ACL

Fonte: Adaptado de (FIPA, 2010) – SC00037J

### 3.2.1.2 Protocolos de Interação FIPA

A FIPA também define um conjunto de protocolos para comunicação de agentes. Um protocolo consiste em uma sequência pré-definida de mensagens, com seus atos performativos, que devem ser trocadas entre agentes em uma interação, para atender um objetivo (FIPA, 2010). A tabela 3 apresenta os protocolos especificados pela FIPA e que estão em estado de padrão ou experimental.

<b>Especificação</b>	<b>Protocolo</b>
SC00026	FIPA Request Interaction Protocol Specification
SC00027	FIPA Query Interaction Protocol Specification
SC00028	FIPA Request When Interaction Protocol Specification
SC00029	FIPA Contract Net Interaction Protocol Specification
SC00030	FIPA Iterated Contract Net Interaction Protocol Specification
XC00031	FIPA English Auction Interaction Protocol Specification
XC00032	FIPA Dutch Auction Interaction Protocol

<b>Especificação</b>	<b>Protocolo</b>
	Specification
SC00033	FIPA Brokering Interaction Protocol Specification
SC00034	FIPA Recruiting Interaction Protocol Specification
SC00035	FIPA Subscribe Interaction Protocol Specification
SC00036	FIPA Propose Interaction Protocol Specification

Tabela 3: Protocolos de comunicação definidos pela FIPA

Fonte: Adaptado de (FIPA, 2010)

No desenvolvimento deste trabalho é utilizado como base o protocolo “*FIPA Request Interaction Protocol Specification- SC00026*”, que é apresentado a seguir.

### ***FIPA Request Interaction Protocol Specification***

O protocolo de interação de requisição é utilizado para que um agente solicite, através de mensagem com ato performativo *request*, que outro agente execute uma ação. Sendo que o receptor da mensagem pode recusar-se a executar a solicitação respondendo com ato *refuse* ou aceitar a requisição, através do ato *agree* e executar a ação. Após a execução da ação o receptor responde a requisição informando falha na execução, através do ato *failure* ou o resultado através do ato *inform*, sendo que o resultado pode ser apenas o indicativo que a ação foi executada ou o resultado produzido pela execução da ação. A figura 4 apresenta o diagrama de sequência do protocolo, demonstrando a troca de mensagens possíveis, entre dois agentes.

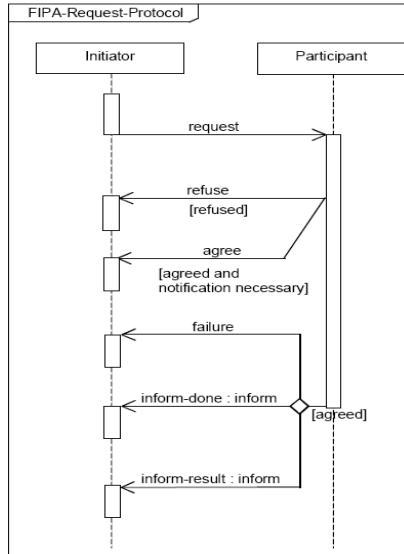


Figura 4: FIPA Request Interaction Protocol  
 Fonte: (FIPA, 2010) – SC00026H

### 3.3 AMBIENTES PARA DESENVOLVIMENTO DE SISTEMAS MULTIAGENTES

O desenvolvimento de sistemas, utilizando o paradigma de agentes é possível a partir qualquer linguagem de programação. Porém visto que existem *frameworks*, desenvolvidos em diferentes tecnologias e aprimorados ao longo do tempo, com objetivo de prover a arquitetura para desenvolvimento de sistemas multiagente é altamente recomendada à utilização destes *frameworks*. Com isso o foco do desenvolvimento fica com a solução do problema e não com a preparação do ambiente de execução dos agentes e a infra-estrutura de comunicação, que já são definidos pelos *frameworks*.

Ambientes para desenvolvimento de software orientado a agentes facilitam o trabalho do desenvolvedor, visto que implementam as camadas básicas de um sistema multiagente, as quais permitem o gerenciamento do ambiente habitado pelos agentes e a criação dos próprios agentes, além do meio de comunicação entre os agentes. Entre os ambientes existentes podem ser destacados: JATLite, Zeus, Jade, Aglets, Fipa-OS, Jack, Jason (SILVA, 2007; BORDINI, et al., 2007). Entre estes *frameworks* tem destaque o JADE, visto que é um

*framework* que segue as especificações definidas pela FIPA e possui equipe evolvida em pesquisas e melhoramentos do mesmo.

### 3.3.1 JADE

Jade (Java Agent Development Framework) é um *framework* que simplifica o desenvolvimento de sistemas multiagente, implementado em linguagem Java, de acesso livre e baseado nas especificações FIPA (JADE, 2010). Jade consiste em um ambiente para execução de agentes, que permite a criação de agentes e o gerenciamento de suas ações, através de ferramentas gráficas.

Cada instância do ambiente de execução do Jade é chamada de contêiner (*Container*), onde agentes são registrados e gerenciados. Um conjunto de contêineres forma uma plataforma, onde sempre existe um contêiner principal, ao qual podem estar registrados outros contêineres. O contêiner principal contém os agentes MAS (*Agent Manager System*) e DF (*Directory Facilitator*), responsáveis pelo gerenciamento e o registros de agentes na plataforma. É através do registro de um agente no contêiner, que os serviços providos por este ficam disponíveis e podem ser localizados por outros agentes (BELLIFEMINE, et al., 2007).

O ambiente de execução do Jade disponibiliza a infra-estrutura para troca de mensagens entre os agentes de acordo com a especificação FIPA ACL, sendo que já implementa alguns protocolos deste padrão. Também permite o trabalho com contêineres e agentes remotos, e a utilização de ontologias na comunicação entre agentes.

No Jade, cada agente registrado possui uma fila de mensagens, onde o agente pode gerenciar as mensagens a ele endereçadas e originadas do processo de comunicação com outros agentes, durante o seu ciclo de vida. O ciclo de vida de um agente no ambiente de execução Jade é definido pelo diagrama de estados da figura 5.

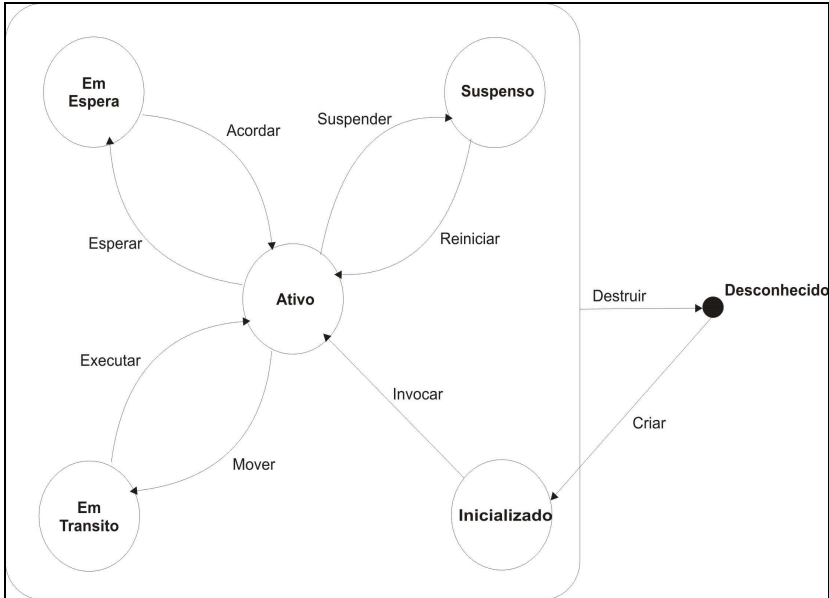


Figura 5: Ciclo de vida de agentes, definido pela FIPA

Fonte: Adaptado de (JADE, 2010).

O *framework* Jade é composto por um conjunto de bibliotecas que são descritas a seguir (BELLIFEMINE, et al., 2007):

- **jade.core**: implementação do *kernel* do sistema. Refere-se as funcionalidades do ambiente de execução Jade, com classes que permitem a criação de agentes e seus comportamentos.
- **jade.content**: implementação de suporte a ontologias.
- **jade.domain**: implementação dos agentes MAS e DF, de acordo com especificações FIPA.
- **jade.gui**: coleção de componentes Java úteis para criação de interface gráfica para agentes Jade.
- **jade.imtp**: implementação do protocolo IMTP (Internal Message Transport Protocol).
- **jade.mtp**: implementação do protocolo MTP (Message Transport Protocol), sobre os protocolos HTTP e IIOP.
- **jade.lang.acl**: implementação do suporte a linguagem FIPA ACL.
- **jade.proto**: implementação de alguns protocolos de interação de propósito geral, inclusive especificados pelo padrão FIPA.
- **jade.util**: conjunto de classes de utilidades gerais.
- **jade.tools**: implementação das ferramentas gráficas de Jade.



- ***jade.wrapper***: permite que o Jade seja utilizado como biblioteca em aplicações Java externas.

Sistemas multiagente geralmente são complexos, e para facilitar o desenvolvimento, o Jade disponibiliza através da biblioteca *jade.tools*, um conjunto de ferramentas gráficas que permitem a administração e o monitoramento do SMA. A figura 6 apresenta a interface gráfica do Jade também chamada RMA (*Remote Monitoring Agent*). A partir do RMA é possível administrar a plataforma de agentes e acessar as ferramentas descritas logo abaixo (BELLIFEMINE, et al., 2007).

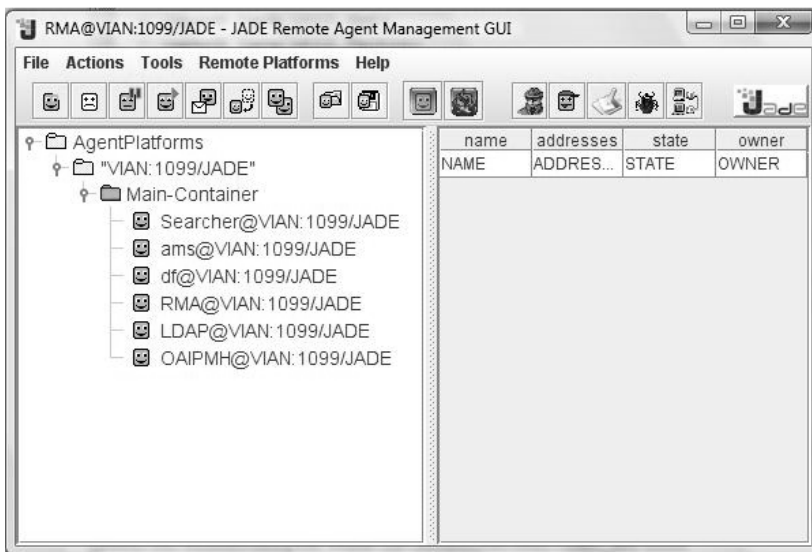


Figura 6: Interface gráfica do Jade - RMA (Remote Monitoring Agent)  
Fonte: Adaptado de (JADE, 2010).

Ferramentas disponibilizadas pelo Jade para administração e monitoramento de sistemas multiagente:

- ***Sniffer***: permite o monitoramento de mensagens trocadas entre agentes, sendo possível visualizar o conteúdo das mensagens.
- ***Dummy***: permite simular o envio e recebimento de mensagens FIPA ACL, bem como a listagem de mensagens trocadas.
- ***Introspector***: permite o monitoramento das atividades realizadas por um agente durante seu ciclo de vida.
- ***Log Manager***: disponibiliza informações de log do SMA, como um todo.

### 3.4 MODELAGEM E ESPECIFICAÇÃO DE SMA

Efetuar a especificação e a modelagem do sistema que se pretende construir é extremamente necessário, para desenvolvimento de software de qualidade. Estas atividades visam diminuir custos, tempo de desenvolvimento e mitigar os riscos envolvidos no desenvolvimento, através da utilização de métodos, processos, tecnologias e princípios. A Engenharia de Software é área que desenvolve metodologias e ferramentas que permitem a especificação e modelagem de sistemas (REZENDE, 2005).

Para o desenvolvimento de sistemas multiagente, também existe a necessidade de especificação e modelagem, porém as metodologias e ferramentas utilizadas no desenvolvimento orientado a objetos, por exemplo, não atendem todas as especificidades de um SMA. Com isso, surgiu a área de pesquisa de Engenharia de Software Orientada a Agentes - AOSE (*Agent Oriented Software Engineering*).

#### 3.4.1 Engenharia de Software Orientada a Agentes

A engenharia de software orientada a agentes tem objetivo de desenvolver metodologias e ferramentas, que permitam a modelagem de sistemas multiagente, considerando que este tipo de sistema baseia-se no uso de agentes para resolver problemas computacionais complexos, dinâmicos e distribuídos. Existem várias metodologias desenvolvidas para modelagem de sistemas multiagente, mas não existe uma que seja amplamente aceita, devido às variadas abordagens empregadas pelas mesmas (OJEDA, et al., 2007)

Entre as metodologias existentes estão AUML, Gaia, MaSE, Tropos, Prometheus (SILVA, 2007; BERGENTI, et al., 2004), as quais se baseiam em diferentes abordagens e definem modelos de processo estáticos. Já o *framework* O-MaSE, possibilita a customização do processo de modelagem e permite a integração das características relevantes das metodologias existentes, com objetivo de se tornar amplamente aceito (O-MASE, 2010). A seguir são apresentados mais detalhes sobre *framework* O-MaSE, que teve sua origem a partir da metodologia MaSE (O-MASE, 2010).

### 3.4.1.1 Organization Multi-agent Software Engineering

O-MaSE é um *framework* que permite a definição do processo de engenharia, para o desenvolvimento de sistemas orientados a agentes, de forma flexível e customizada. Permitindo que o desenvolvedor defina a metodologia de desenvolvimento, de acordo com a necessidade imposta pelo sistema a ser desenvolvido. Para atender este objetivo o O-MaSE é definido através de um metamodelo, um conjunto de métodos e um conjunto de orientação (OJEDA, et al., 2007).

O metamodelo define os conceitos de análise, projeto e implementação, bem como as restrições entre eles. O conjunto de métodos determina como os produtos de análise e projeto podem ser criados e utilizados, no *framework* e as orientações definem como os métodos podem ser combinados, para definir uma instância da metodologia O-MaSE (DELOACH, et al., 2007).

A metamodelo possui uma abordagem organizacional, onde a organização é composta por cinco entidades: Objetivos, Papéis, Agentes, Domínio e Políticas, conforme apresentado pela figura 7, (OJEDA, et al., 2007).

A entidade Agente possui atrelada a si os conceitos de: Capacidades, Planos e Ações. Outros conceitos relevantes do metamodelo são as Organizações de Agentes e os Protocolos.

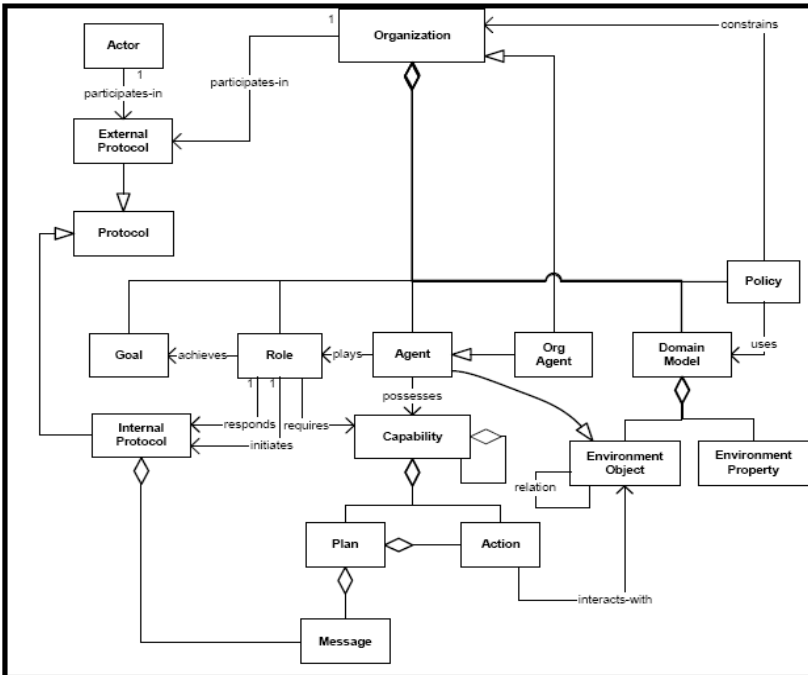


Figura 7: Metamodelo O-MaSE

Fonte: (DELOACH, et al., 2007)

O conjunto de métodos definidos pelo *framework*, apresentados através da figura 8, demonstra a forma de modelagem interativa que o O-MaSE permite, através do relacionamento entre fases, unidades de trabalho, produtos de trabalho, produtores e linguagens, utilizadas em cada interação. As unidades de trabalho estão divididas em atividades, tarefas e técnicas (DELOACH, et al., 2007).

As orientações determinam como o conjunto de métodos pode ser combinado, para realização das tarefas. Estas determinações são definidas em nível de pré-condições e pós-condições, sobre as entradas e saídas de uma tarefa (DELOACH, et al., 2007).

Unidades de Trabalho						
Fases	Atividades	Tarefas	Técnicas	Produtos de Trabalho	Produtores	Linguagem
Inicialização	Engenharia de Requisitos	Modelo de Objetivos	Decomposição AND/OR	Arvore de Objetivos AND/OR	Modelador de Objetivos	Linguagem Natural para documentos de texto UML para modelos específicos Agente UML
		Refinamento de Objetivos	Análise de relações entre Objetivos	Arvore de Objetivos refinados		
	Análise	Modelo de Interfaces Organizacionais	Modelagem Organizacional	Modelo Organizacional	Modelador Organizacional	
		Modelo de Papéis	Modelagem de Papéis	Modelo de Papéis	Modelador de Papéis	
		Definição de Papéis	Descrição de Papéis	Documento de Definição de Papéis		
		Modelo de Domínio	Notações UML Tradicionais	Modelo de Domínio	Especialista de Domínio	
Construção	Projeto	Modelo de Agentes	Modelagem de Agentes	Modelo de Classes de Agentes	Modelador de Agentes	Notação específica O-MaSE
		Modelo de Protocolos	Modelagem de Protocolos	Modelo de Protocolos	Modelador de Protocolos	Linguagem Formal para especificação de
		Modelo de Planos	Especificação de Planos	Modelo de Planos de Agente	Modelador de Planos	
		Modelo de Políticas	Especificação de Políticas	Modelo de Políticas	Modelador de Políticas	propriedades do sistema.
		Modelo de Capacidades	Modelagem de Capacidades	Modelo de Capacidades	Modelador de Capacidades	
		Modelo de Ações	Modelagem de ações	Modelo de Ações	Modelador de Ações	
		Modelo de Serviços	Modelagem de Serviços	Modelo de Serviços	Modelador de Serviços	

Figura 8: Estrutura do conjunto de métodos O-MaSE

Fonte: Adaptado de (DELOACH, et al., 2007)

A ferramenta AgentTool III (AGENTTOOL, 2009) é um plugin do ambiente de desenvolvimento Eclipse, desenvolvido em Java e que permite o projetista modelar o comportamento de sistemas multiagente, graficamente. O AgentTool III tem suporte a metodologia O-MaSE e disponibiliza as técnicas, que ajudam o projetista na realização das tarefas, nas fases de análise e projeto definidas pelo conjunto de métodos da metodologia.



## 4 RECUPERAÇÃO DE INFORMAÇÃO

A Recuperação de informação (RI) é definida como a tarefa de encontrar documentos relevantes para necessidades de informação de um usuário (RUSSEL, et al., 2004). RI é uma área da computação que visa explorar técnicas e desenvolver softwares com intuito de facilitar a localização de informação, que satisfaça necessidades de informação, em meio digital (MANNING, et al., 2008). Exemplos bem conhecidos de sistemas de RI são os mecanismos de busca na web, que tem se tornando uma importante fonte de acesso a informação.

Segundo (RUSSEL, et al., 2004) um sistema de RI pode ser estruturado através dos conceitos de:

- **Documentos:** Estruturas como textos, hipertextos, imagens, que possuem informação e são definidos pelo sistema de RI, como sendo o domínio, no qual tem objetivo de recuperar informação.
- **Consulta:** Necessidade de informação do usuário expressada através de palavras-chave, de acordo com a linguagem definida pelo sistema de RI.
- **Resultados:** Subconjunto dos documentos que possuem informação considerada relevante para a necessidade de informação do usuário, expressada na consulta. Ou seja, os documentos que podem ser úteis, para o usuário solicitante da consulta.
- **Apresentação:** Forma de apresentar ao usuário os resultados, considerando uma medida de grau de relevância definida pelo sistema de RI, que determina a relevância de cada documento perante o objetivo da consulta.

A qualidade de sistemas RI é avaliada através dos conceitos estatísticos de precisão e cobertura, onde a precisão consiste na proporção de documentos do resultado que são relevantes a consulta e a cobertura consiste da proporção de documentos relevantes no domínio do sistema de RI, que são contemplados pelo resultado (MANNING, et al., 2008). Porém, visto que uma informação pode ser representada por termos diferentes com significado equivalente, o resultado de um sistema de RI é influenciado pela qualidade das consultas definidas pelos usuários.

Os modelos clássicos de recuperação de informação são: booleano, vetorial e probabilístico, sendo que cada modelo tem suas particularidades (MANNING, et al., 2008).

- **Modelo Booleano:** representação dos termos da consulta como uma expressão booleana e o resultado formatado pelos documentos onde a expressão booleana seja verdadeira.
- **Modelo Vetorial:** representação da consulta como um vetor de termos e também dos documentos do domínio como vetores documento, sendo o resultado definido e organizado pelos documentos que possuam maior similaridade entre o seu vetor documento e o vetor da consulta.
- **Modelo Probabilístico:** resultado é determinado através de técnicas de probabilidade e estatística, onde a ordenação dos documentos no resultado é definida pela probabilidade do documento ser relevante para determinada consulta.

Em um sistema de RI além da utilização de um modelo de recuperação, podem ser implementadas técnicas que tem o objetivo de melhorar a avaliação do sistema. Entre as técnicas podem ser destacadas: expansão de consulta, *feedback* de relevante e personalização (CARDOSO, 2000; MANNING, et al., 2008).

#### 4.1 INDEXAÇÃO

Um dos principais conceitos em recuperação de informação é o índice invertido, que consiste em uma lista de termos onde para cada item da lista, está relacionado um conjunto de documentos, nos quais o termo ocorre e é considerado relevante (MANNING, et al., 2008). A indexação tem objetivo de organizar os termos considerados relevantes em um conjunto de documentos, que correspondem ao domínio de um sistema de RI e indicar para cada termo os documentos onde o termo ocorre. Através do índice uma busca por um determinado termo pode facilmente retornar os documentos, onde o termo se encontra sem a necessidade de percorrer todo o texto do documento. Assim é possível otimizar o tempo de resposta de um sistema de RI.

O processo de indexação consiste em analisar o conjunto de documentos pertencentes ao domínio do sistema de RI, destacando para cada documento os termos relevantes e que caracterizam o documento. Após a identificação dos termos é possível utilizar técnicas para determinar o quão relevante para o documento é cada termo, através da definição de pesos para os termos, estes pesos são utilizados para organização dos resultados de uma consulta. O índice corresponde à lista com os termos que representam o conjunto de documentos, onde



cada termo esta relacionado aos documentos em que ocorre, com a respectiva relevância. O índice é armazenado com o objetivo de ser consultado no momento que o usuário solicitar informação ao sistema de RI, possibilitando indicar os documentos que possuem os termos relacionados pela consulta do usuário.

A seguir são destacadas algumas técnicas utilizadas para efetuar a análise de documentos, com a finalidade de destacar os termos relevantes e que representam a informação do documento. Também é destacado um método para determinar a relevância dos termos de um documento.

#### 4.1.1 Técnicas para auxiliar a indexação

Através de técnicas de mineração de texto é possível definir os termos que identificam a informação contida em um documento. Algumas das técnicas bem difundidas na área de recuperação de informações são (MANNING, et al., 2008; RUSSEL, et al., 2004):

- **Remoção de *Stopwords***: técnica que visa eliminar de um texto, termos sem significado relevante para categorizar a informação nele contida.
- ***Stemming***: técnica que visa extrair apenas o radical dos termos relevantes existentes em um texto. Não considerar as flexões de termos na categorização da informação, contida em um texto, facilita no sentido que diminui o número de termos a serem analisados. Por outro lado caso flexões possuam significado diferente esta técnica causa perda de precisão.
- **Normalização**: técnica utilizada para tornar todos os termos de um texto minúsculos e remover a acentuação com objetivo de facilitar a comparação entre termos. Porém tornar os termos minúsculos pode dificultar a busca por siglas.

#### 4.1.2 Método para determinação de pesos

Um método utilizado para calcular o peso de termos em um documento, chamado *Term Frequency x Inverse Document Frequency* (TF x IDF), considera a frequência de um termo em um documento juntamente com a frequência de documentos, que possuem este termo, em uma coleção de documentos (MANNING, et al., 2008). O método utiliza a fórmula abaixo para determinar os pesos para os termos do documento:

$$W_{dt} = f_{d,t} \times \log(N/f_t)$$

- $W_{dt}$  é o peso do termo para o documento
- $N$  é o número total de documentos na coleção
- $f_t$  é o número de documentos que possuem o termo
- $f_{d,t}$  é o número de ocorrências do termo no documento

Este método atribui altos pesos para termos mais raros, considerando que estes são melhores discriminantes para o documento.

Existem outras formas de determinação de pesos, considerando o objetivo do sistema de RI. Por exemplo, na recuperação de páginas na web, podem ser monitorados os *hiperlinks* e definido maior peso para *hiperlinks* mais acessados.

## 4.2 WEB SEMÂNTICA

Os métodos de recuperação de informação citados anteriormente utilizam fortemente o conceito de palavra-chave, considerando apenas aspectos sintáticos e de ocorrência de termos. Porém esta característica não garante boa precisão, principalmente quando na linguagem dos documentos existirem formações, como: sinônimos, homônimos, meronímia (parte de) e hiperonímia (é um). Estas formações dificultam a recuperação automatizada de informação se considerada apenas a sintaxe dos termos.

A área de pesquisas em web semântica tem proposto e desenvolvido técnicas que auxiliam o tratamento das formações citadas, através do envolvimento semântico entre termos, suportando a recuperação semântica de informação (MANGOLD, 2007; GUHA, 2003). Permitindo assim além da análise sintática à análise semântica dos termos contidos em páginas web e documentos. O objetivo da web semântica é permitir que computadores consigam interpretar e inferir informações a partir de documentos em formato digital, disponíveis na web (BERNERS-LEE, et al., 2001).

Na web semântica existem diferentes técnicas para representação semântica de conceitos como: dicionários, taxonomias, thesaurus e ontologias. Sendo que ontologias possuem o maior poder de expressividade e formalismo, para conceitualização de conceitos relacionados a um domínio (EUZENAT, et al., 2007).

Estas técnicas desenvolvidas no âmbito da web semântica auxiliam no aumento da precisão de sistemas de RI.

#### 4.2.1 Ontologias

Uma ontologia é constituída por entidades que podem ser: classes ou conceitos, instâncias ou indivíduos, relações ou propriedades, tipos de dados e valores (EUZENAT, et al., 2007). Ontologias são elaboradas através de linguagem capaz de atribuir o formalismo necessário e entre as linguagens para construção de ontologias, tem destaque a linguagem OWL (Web Ontology Language), que é uma recomendação da W3C (OWL-W3C, 2009).

Ontologias representam o conhecimento de um domínio através dos relacionamentos entre os conceitos pertencentes ao domínio. Utilizando uma ontologia um sistema computacional pode inferir informações relativas ao domínio tratado pela mesma.

Ontologias podem ser empregadas de diferentes formas na recuperação de informação, por exemplo, auxiliando a interpretação de termos, no momento da mineração de textos, que objetiva encontrar os termos relevantes para identificar um documento. Também podem ser empregadas na expansão de consultas (MANGOLD, 2007), onde os termos da consulta informados pelo usuário, para satisfazer sua necessidade de informação são processados pela ontologia, em busca de classes ou entidades existentes na ontologia e que tem relações com os termos informados. Com isso, ferramentas de busca podem localizar informações que estão identificadas sintaticamente por termos não contidos na consulta solicitada pelo o usuário, mas que permitem a satisfação da sua necessidade.

A técnica de expansão de consultas utilizando ontologias permite, através das relações semânticas definidas em ontologias por exemplo de subclasse, superclasse, disjunção e equivalência, a identificação de formações como sinônimos e homônimos. Dada a complexidade das linguagens naturais, a análise semântica de termos para recuperação de informação tende a melhorar a qualidade dos resultados de um sistema de RI (GUHA, 2003).



## 5 SISTEMA PROPOSTO

A heterogeneidade que cerca a tecnologia de OAs ocorre principalmente de duas formas. A primeira é referente aos padrões de metadados utilizados para descrever OAs, onde cada padrão define a sintaxe e a semântica dos elementos que o compõe, como mostra a tabela 4. A segunda é referente a arquitetura dos repositórios que tem o objetivo de armazenar os OAs, onde cada repositório define a tecnologia de armazenamento e a forma de acesso aos objetos.

O objetivo do sistema proposto é proporcionar a reutilização de OAs, através do desenvolvimento de agentes de software com capacidade de conviver com a heterogeneidade do ambiente, no cenário de recuperação de OAs. O sistema é capaz de indexar e recuperar OAs em diferentes repositórios, de forma transparente para o usuário.

<b>LOM</b>	<b>Dublin Core</b>	<b>IMS LRM</b>	<b>ISO MLR</b>	<b>AICC Metadata</b>
Title	Title	Title	Title	Title
Description	Description	Description	Description	Description
Keyword	Subject	Keyword	Subject	Keyword
Contribute, Role	Contributor, Creator, Publisher	Contribute, Role	Contributor, Role	Contribute, Role
Format	Format	Format	Format	Format

Tabela 4: Exemplo de heterogeneidade entre elementos dos padrões de metadados.

A seguir é detalhada a arquitetura do sistema proposta bem como o modelo de agentes de software desenvolvido para atender os objetivos do trabalho. Também é detalhado o protótipo desenvolvido, que tem objetivo de validar o modelo de recuperação de OAs, que visa permitir a reusabilidade e compartilhamento dos mesmos.

### 5.1 ARQUITETURA

A arquitetura do sistema multiagente proposto, para indexação e recuperação de OAs é apresentada na figura 9, e possui os seguintes componentes:

- **Repositórios (Rep 01, Rep 02, ..., Rep n):** representa o conjunto de repositórios de OAs sobre os quais o sistema efetua a indexação e a recuperação de objetos. Estes repositórios podem conter OAs descritos com diferentes padrões de metadados, possuir estruturas diferentes e estão distribuídos na web. Cada repositório pode ser atualizado constantemente com a inclusão, alteração ou remoção de OAs.
- **Sistema multiagente:** composto por agentes indexadores e agentes recuperadores, que serão detalhados a seguir.
- **Mapeamento de metadados:** consiste em uma estrutura de dados onde é feita a correlação entre os elementos de metadados de cada padrão de metadados, tratado pelo sistema. Esta estrutura auxilia os agentes na extração da informação contida nos elementos de metadados dos OAs, e que é utilizada para indexação dos mesmos.
- **Ontologias de domínio:** conjunto de ontologias utilizadas pelos agentes recuperadores, que permitem a expansão das consultas solicitadas pelos usuários. Permitem agregar a consulta solicitada, informação semântica através das relações de subclasses e superclasses, das propriedades e dos indivíduos existentes nas ontologias e relacionados com os termos informados pelo usuário para a consulta.
- **Serviço de busca:** corresponde ao servidor web que disponibiliza a interface de busca, onde os usuários efetuam as consultas ao sistema, em busca de OAs e onde os resultados da consulta são apresentados após a realização da busca no sistema multiagente. Possibilita ao usuário informar os termos que deseja consultar e a escolha de um domínio de conhecimento no qual os termos informados devem ser buscados. Com isso permite que o agente recuperador utilize a ontologia de domínio correspondente, para realizar a expansão da consulta.

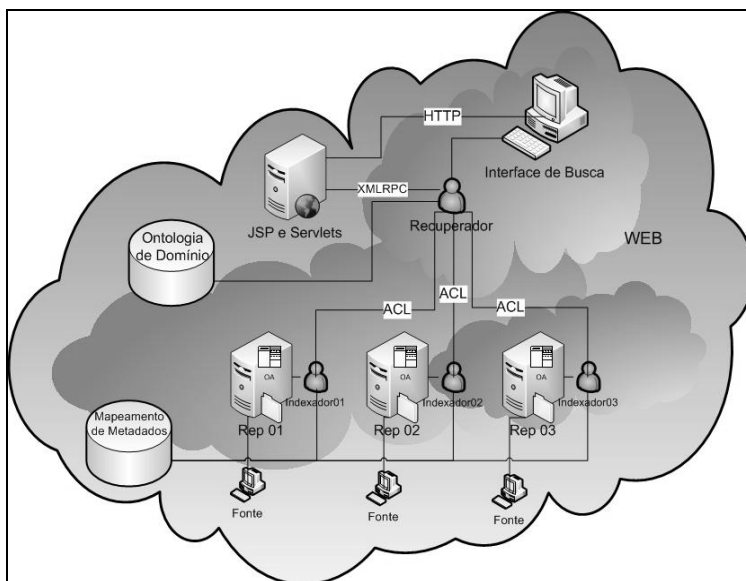


Figura 9: Esquema que representa a arquitetura do sistema proposto

A seguir é detalhada a utilização da arquitetura para o desenvolvimento de um protótipo, onde são relacionadas as tecnologias utilizadas em cada componente da arquitetura e também como é feita a ligação entre os componentes.

## 5.2 SISTEMA MULTIAGENTE

O sistema multiagente foi modelado através da metodologia O-MASE, e os diagramas desenhados com auxílio da ferramenta AgentTool III. A ferramenta permite a construção de uma série de diagramas que dão visibilidade ao sistema multiagente sendo construído, como é o caso do diagrama de agentes mostrado na figura 10. No diagrama de agentes é possível visualizar os dois tipos de agentes que o sistema proposto possui: agente indexador e agente recuperador.

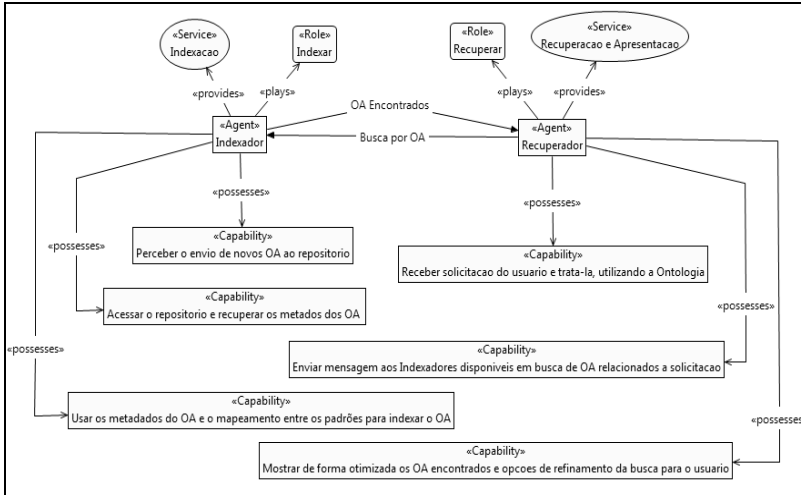


Figura 10: Diagrama de agentes

### 5.2.1 Agente Indexador

O Agente Indexador possui comportamento capaz de recuperar os metadados de um OA localizado em um repositório. O agente deve conhecer a localização do repositório, os dados para acessar o repositório e o padrão de metadados utilizado para descrever os OAs existentes no repositório.

Utilizando a estrutura que mapeia as diferenças entre os padrões de metadados e com a capacidade de acessar no repositório, a descrição de metadados dos OAs, o agente consegue extrair o conteúdo dos elementos de metadados. De posse deste conteúdo o agente é capaz de tratá-lo utilizando técnicas de mineração de texto e calcular os pesos, como visto no capítulo 4, para os termos considerados relevantes para o OA. Após definidos os pesos, o agente efetua a indexação do OA, utilizando a linguagem XML, como mostra a figura 11.

O índice consiste em mapear para um dado termo os objetos que possuem este termo com o respectivo peso de relevância que o termo representa para o objeto.

O agente indexador também possui capacidade de receber de agentes recuperadores, solicitações por OAs, através de mensagens. Para cada solicitação recebida, o agente deve percorrer o seu índice em busca de objetos relevantes. Para isso, o agente identifica os termos recebidos na solicitação junto ao seu índice e responde ao agente recuperador,



através de mensagem contendo os objetos indexados pelos termos que existem na solicitação recebida, juntamente com o conteúdo de metadados que contextualizam cada objeto considerado relevante para a solicitação.

É desejável que o agente indexador possua o índice também em memória para melhorar o desempenho do sistema, porém dado o volume de dados no índice isso não seja possível, mesmo assim é desejável alguma forma de *buffer*, com objetivo de diminuir os acessos a disco. Também é importante o comportamento que permita o agente perceber mudanças no repositório para que possa manter a consistência de seu índice. Sendo que estas funcionalidades não fazem parte do escopo deste trabalho.

```
<indice>
  <conceito id="termoA">
    <objeto peso="0,1">objetoA</objeto>
    <objeto peso="0,5">objetoB</objeto>
  </conceito>
  <conceito id="termoB">
    <objeto peso="0,2">objetoB</objeto>
  </conceito>
</indice>
```

Figura 11: Fragmento XML do índice gerado pelo agente indexador

### Processo de indexação de um repositório por um agente indexador

- O agente acessa o repositório identificando sua estrutura e o padrão de metadados.
- Para cada objeto existente no repositório o agente deve:
  - Acessar os metadados do objeto
  - Recuperar o conteúdo dos elementos de metadados considerados úteis para a indexação.
  - Efetuar o tratamento com técnicas de RI para identificar os termos que representam o objeto
  - Calcular o peso de relevância de cada termo definido no passo anterior.
  - Armazenar o índice para o objeto de acordo com a figura 11.

### 5.2.2 Agente Recuperador

O agente recuperador possui comportamento capaz de receber solicitações do usuário, que efetua consulta ao sistema em busca de OAs. O agente recebe os termos informados pelo usuário através da interface de busca e também o domínio de conhecimento, caso escolhido pelo usuário. De posse dos termos o agente é capaz de executar a expansão da consulta utilizando a ontologia referente ao domínio escolhido pelo usuário, ou caso o usuário não tenha definido domínio utilizar mais de uma ontologia.

Após a realização da expansão da consulta com auxílio de ontologias o agente recuperador é capaz de enviar mensagens, aos agentes indexadores ativos em seu contêiner, requisitando OAs relevantes à solicitação do usuário.

O agente recuperador possui comportamento capaz de processar as mensagens recebidas dos agentes indexadores, que contêm os objetos localizados por estes e que são relevantes para a solicitação realizada. O agente então organiza os objetos enviados por todos os agentes indexadores que responderam, considerando os pesos de cada objeto que compõe o resultado da consulta. Feito isso o agente mostra ao usuário, utilizando a interface de busca, os OAs encontrados em ordem de relevância definida pelos pesos calculados na indexação.

É desejável que o agente possua capacidade, para a partir do conjunto de resultados, oferecer ao usuário opções para refinamento da consulta, recuperar detalhes de um determinado objeto e acessar os OAs apresentados no resultado da consulta. Sendo que as funcionalidades de opções de refinamento da consulta e detalhamento de objetos estão fora do escopo deste trabalho.

#### **Processo de execução de uma consulta por um usuário**

- O usuário acessa a interface de busca, informa os termos que deseja para a consulta de objetos de aprendizagem, se desejar escolhe um domínio sobre o qual os objetos devem estar relacionados e aciona a consulta
- O agente recuperador recebe a solicitação de consulta do usuário e:
  - Efetua a expansão da consulta utilizando a base de ontologias de domínio.

- Envia mensagem para os agentes indexadores ativos no ambiente, contendo os termos da consulta.
- Cada agente indexador efetua a busca em seu índice por objetos relevantes a consulta e envia mensagem ao agente recuperador informando os objetos encontrados.
- O agente recuperador recebe as mensagens de cada agente indexador acionado, contendo os objetos encontrados.
  - Efetua a organização dos objetos encontrados considerando os pesos de relevância calculados na indexação.
  - Retorna para o usuário os objetos encontrados e organizados que são exibidos na interface do serviço de busca.
- O usuário acessa os objetos que satisfazem a sua necessidade.

### 5.2.3 Protocolo

O protocolo de comunicação definido para interação entre os agentes, que compõe o sistema multiagente tem como base o protocolo FIPA-Request. Como mostra a figura 12, o agente recuperador envia a mensagem Solicitação para o agente indexador, contendo os termos da consulta e utilizando o ato performativo *request*. O agente indexador envia a mensagem Resultado para a agente recuperador, contendo os objetos encontrados no índice para a solicitação realizada e utilizando o ato performativo *inform*.

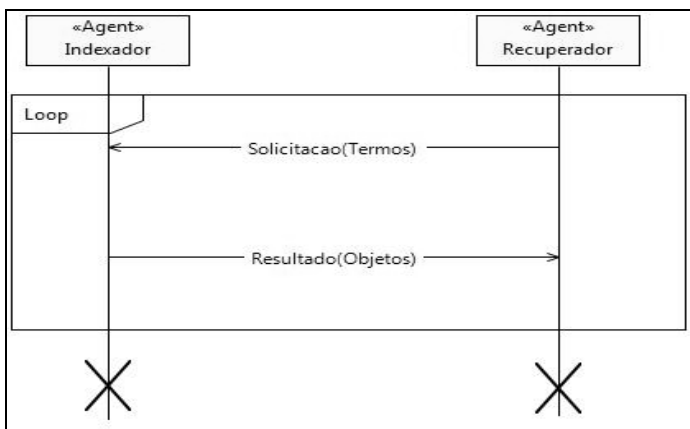


Figura 12: Diagrama de protocolo

### 5.2.4 Ciclo de Vida dos agentes no SMA

Para cada repositório de OAs, tratado pelo sistema proposto deve existir uma instância do agente indexador registrada no contêiner do sistema multiagente, que é responsável pela indexação do respectivo repositório e pela recuperação de objetos do repositório quando solicitado.

Para cada busca efetuada por algum usuário, deve ser criada uma instância do comportamento do agente recuperador, que tem objetivo de tratar a respectiva consulta. O comportamento é finalizado após a disponibilização do resultado da consulta ao usuário.

### 5.3 DESENVOLVIMENTO DO PROTÓTIPO

O protótipo desenvolvido tem objetivo de exemplificar o problema tratado pelo trabalho de pesquisa e também demonstrar a viabilidade do modelo de recuperação de OAs apresentado, através do sistema proposto. O desenvolvimento teve como base a linguagem de programação Java e a utilização de tecnologias já consolidadas. Algumas das tecnologias utilizadas para implementação da arquitetura proposta são:

- **Framework JADE** (JADE, 2010): utilizado para a implementação do sistema multiagente, permitindo a criação dos agentes, a configuração do ambiente de agentes e a comunicação entre agentes através da linguagem FIPA ACL.
- **JSP e Servlets**: tecnologias utilizadas para implementação da interface de busca que compõe o serviço de busca e é a parte do sistema com a qual o usuário interage.
- **Servidor Tomcat 6** (APACHE, 2010): servidor web que disponibiliza a interface de busca.
- **Framework Jena** (JENA, 2010): utilizado para implementação da comunicação dos agentes recuperadores com as ontologias de domínio. Permite a realização de consultas e inferências a partir de ontologias em OWL.
- **XMLRPC** (XMLRPC, 2003): protocolo utilizado para a comunicação entre o sistema multiagente e o serviço de busca. Permite a troca de conteúdo em XML a partir de chamadas a processos remotos.
- **APIs URL, JLDAP e JDOM**: bibliotecas Java que disponibilizam as funcionalidades para a conexão dos agentes aos repositórios de

OAs, a recuperação dos metadados descritos para OAs e o processamento de XML para extração de informação contida nos elementos de metadados.

- **Framework Protégé** (PROTEGE, 2010): utilizado para o desenvolvimento de ontologias e permite a exportação de ontologias em formato OWL.

### Detalhes da Implementação

Com a inicialização do servidor web Tomcat, fica disponível aos usuários a interface de busca apresentada na figura 13, implementada com JSP e *Servlets*. Através da interface de busca, o usuário efetua a consulta por OAs, definido os termos e o domínio desejado.



Figura 13: Interface de Busca

A inicialização do sistema multiagente, através do *framework* JADE, disponibiliza os agentes indexadores, que efetuam a indexação dos repositórios definidos para o sistema. Para isso os agentes utilizam as bibliotecas Java que permitem a conexão aos repositórios e a recuperação dos metadados, sobre os quais são aplicadas as técnicas de recuperação de informação. O índice resultante do processamento de cada agente indexador é mantido na memória do agente e também gravado em disco, utilizando a linguagem XML.

No momento que o usuário comanda a consulta na interface de busca, a página JSP aciona o *servlet*, no servidor web, que possui a implementação do cliente XMLRPC, responsável pela comunicação do serviço de busca com o sistema multiagente. Na inicialização do sistema multiagente também é disponibilizado o agente recuperador, que implementa o servidor XMLRPC. O *servlet* faz uma chamada de processo remoto ao servidor XMLRPC, que permite ao agente recuperador, instanciar o comportamento responsável por tratar a consulta. O comportamento do agente recuperador realiza a expansão da consulta, utilizando o *framework* Jena. O agente gera o modelo da ontologia desejada e efetua consultas SPARQL sobre o conteúdo do modelo, em busca de classes e indivíduos existentes na ontologia e que são relacionados aos termos informados pelo usuário. A utilização de ontologias por agentes de software é exemplificada em (LACLAVIK, et al., 2006).

Após a expansão da consulta o agente recuperador efetua a comunicação com os agentes indexadores, recebe os resultados dos agentes indexadores e faz a organização dos objetos recuperados. De posse do resultado o agente recuperador encaminha-o, via XMLRPC para o *servlet* que exhibe os objetos na interface do serviço de busca, onde o usuário pode acessá-los. Conforme mostra a figura 14.

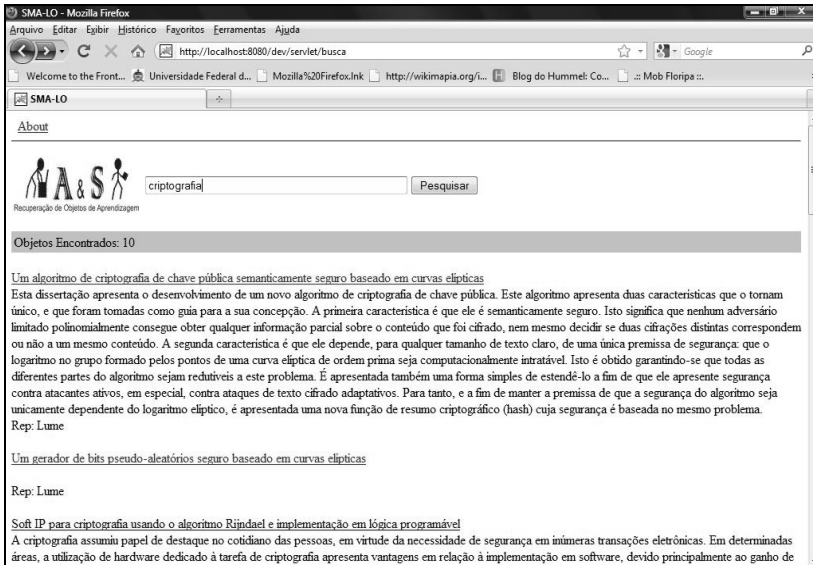


Figura 14: Interface de Resultado

## 6 VALIDAÇÃO DO PROTÓTIPO

A validação do protótipo desenvolvido tem como objetivo demonstrar de forma prática, a utilização do sistema proposto, como possível solução ao problema estudado. Para esta validação foram utilizados dois repositórios de OAs existentes, como será mostrado a seguir. E desenvolvido através do editor *Protégé*, uma ontologia no domínio de segurança da informação, como mostra a figura 15. Esta ontologia é composta de classes do domínio de segurança de informação e estão relacionadas através das propriedade é-um (*is-a*) e parte-de (*part-of*).

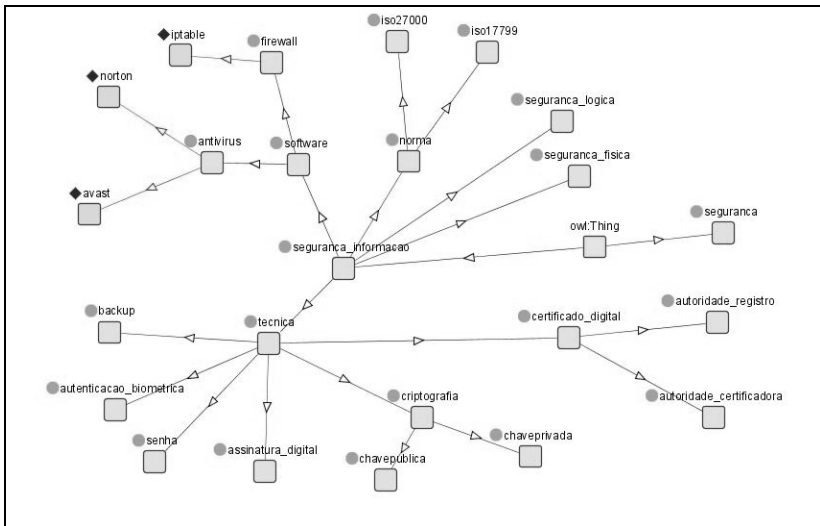


Figura 15: Ontologia para o domínio de Segurança da Informação

Para execução dos testes foi utilizado o ambiente com as seguintes configurações: Notebook HP, com processador AMD Turion64 2x de 2,00 GHz, com 250GB de espaço para armazenamento e memória de 3GB, conectada a internet banda larga. Tanto o sistema multiagente como o servidor web foram inicializados neste ambiente.

## 6.1 REPOSITÓRIOS INDEXADOS

Os repositórios utilizados para a validação do protótipo são:

- **LUME:** Repositório Digital da Universidade Federal do Rio Grande do Sul. Este repositório é estruturado sobre a ferramenta *DSpace* e utiliza o padrão de metadados *Dublin Core*.
- **CESTA:** Coletânea de Entidades de Suporte ao uso de Tecnologia na Aprendizagem. Repositório estruturado sobre um servidor LDAP e utiliza o padrão de metadados LOM.

Neste cenário são instanciados dois agentes indexadores cada um responsável por efetuar a indexação de um repositório, levando em conta suas particularidades e tecnologias envolvidas. Demonstrando assim, o poder de adaptação ao ambiente, que sistemas multiagente permitem.

Cada agente indexador se adapta de forma a conseguir acessar os objetos do repositório e recuperar suas descrições de metadados, de acordo com o padrão utilizado. A partir da descrição de metadados dos objetos são recuperados os valores dos elementos de metadados mais significantes para indexação, como *Title*, *Description* e *Keywords*. Sobre o valor dos elementos de metadados é feito o trabalho de tratamento dos termos como: remoção de *stopwords*, remoção de acentuação e pontuação e também normalizado dos termos para caixa baixa. Após o tratamento dos termos é armazenado o índice do repositório. Para esta validação não foi possível a implementação do cálculo de pesos para os termos indexados, sendo que a relevância dos resultado de consultas é definido a partir da quantidade de termos de uma consulta que faz referência ao mesmo objeto, sendo mais relevante o objeto que esta indexado para o maior número de termos da consulta.

Também é instanciado um agente recuperador responsável por receber da interface de busca, os termos de busca informados pelo usuário. Este agente efetua a expansão da consulta realizada pelo usuário, utilizando a ontologia de domínio desenvolvida.

Definidos os termos para consulta o agente recuperador efetua a comunicação com os dois agentes indexadores, em busca de OAs relacionados aos termos da consulta. Os agentes indexadores efetuam a busca em seus índices, por objetos relevantes e enviam para o agente recuperador o conjunto de objetos localizados, cada objeto é acompanhado por seu título e descrição.



O agente recuperador faz a organização dos objetos enviados pelos agentes indexadores, considerando mais relevante os objetos que possuem o maior número de termos da consulta. Após a organização o agente recuperador envia o resultado para o servidor web, que exhibe os objetos na interface de busca, onde o usuário pode acessá-los.

## 6.2 TESTES

Para execução de testes que viessem a validar o protótipo, foram utilizadas as ferramentas de busca nativas dos repositórios utilizados na validação. Foram efetuadas consultas individualmente em cada repositório e também através do protótipo implementado. Para as consultas foram utilizados termos no domínio de Segurança da Informação, utilizados também para desenvolver a ontologia, apresentada anteriormente. A escolha do domínio levou em conta o conhecimento do autor, com objetivo de permitir a identificação da relevância dos objetos retornados pelas consultas. Devido aos testes serem executados pelo autor, que executa uma determinada consulta em um ambiente e deve analisar os resultados, determinando se cada objeto retornado é ou não relevante para a devida consulta.

A tabela 5 mostra as consultas realizadas, com os respectivos termos e os resultados quantitativos, obtidos através da ferramenta de busca do repositório LUME. Para definição dos objetos relevantes e não relevantes foram analisados os 10 primeiros resultados da consulta, como critério de amostragem.

Consultas	LUME		
	Resultado	Objetos Relevantes	Objetos Não Relevantes
Segurança da Informação	279	9	1
Segurança Física	117	1	9
Segurança Lógica	90	2	8
Certificado Digital	1412	1	9
Assinatura Digital	2794	2	8
Criptografia	397	4	6
Autoridade certificadora	3264	0	10
Software de segurança	150	3	7
Firewall	96	6	4
Antivírus	3	1	2
Técnica de segurança	118	2	8

Norma de segurança	214	0	10
Backup	173	2	8

Tabela 5: Resultado quantitativo das consultas realizadas no repositório LUME.

A tabela 6 apresenta os resultados quantitativos dos testes aplicados através da interface de busca do repositório CESTA. Devido ao baixo número de objetos existentes no repositório, mais especificamente no domínio escolhido para os testes, não foi possível realizar boas comparações.

Consultas	CESTA		
	Resultado	Objetos Relevantes	Objetos Não Relevantes
Segurança da Informação	2	1	1
Segurança Física	0	-	-
Segurança Lógica	0	-	-
Certificado Digital	0	-	-
Assinatura Digital	0	-	-
Criptografia	0	-	-
Autoridade certificadora	0	-	-
Software de segurança	2	1	1
Firewall	1	1	0
Antivírus	0	-	-
Técnica de segurança	2	1	1
Norma de segurança	2	0	2
Backup	0	-	-

Tabela 6: Resultado quantitativo das consultas realizadas no repositório CESTA.

Os testes realizados no protótipo são apresentados na tabela 7. Foi utilizado o mesmo conjunto de termos e expandindo as consultas, utilizando a ontologia apresentada anteriormente. Para expansão foram utilizadas apenas as relações de subclasse da ontologia, agregando a cada consulta as subclasses das classes representadas pelos termos informados pelo usuário. Para definição dos objetos relevantes e não relevantes foram analisados os 10 primeiros resultados da consulta, como critério de amostragem.

Consultas	Consulta Expandida	Protótipo		
		Resul. (tempo)	Obj. Rel.	Obj. Não Relev.
Segurança da Informação	Segurança, informação, criptografia, chave privada, firewall, chave, publica, norma, autenticação biométrica, autoridade registro, segurança lógica, autoridade certificadora, assinatura digital, iso17799, técnica, antivírus, backup, certificado digital, software, iso27000, segurança física, senha.	2012 (130s)	7	3
Segurança Física	-	909 (67s)	1	9
Segurança Lógica	-	213 (17s)	2	8
Certificado Digital	certificado digital, autoridade registro, autoridade certificadora.	258 (21s)	0	10
Assinatura Digital	-	223 (18s)	2	8
Criptografia	Criptografia, chave privada, chave publica.	10 (1,08s)	8	2
Autoridade certificadora	-	5 (0,87s)	0	5
Software de segurança	Software, segurança, firewall, antivírus.	405 (31s)	7	3
Firewall	-	8 (0,95s)	7	1
Antivírus	-	0 (0,12s)	-	-
Técnica de segurança	Técnica, segurança, criptografia, chave privada, chave publica, autenticação, biométrica, autoridade registro, autoridade certificadora, assinatura digital, backup, certificado digital, senha.	612 (52s)	5	5
Norma de segurança	Norma, segurança, iso17799, iso27000	178 (15s)	3	7
Backup	-	0 (0,12s)	-	-

Tabela 7: Resultado quantitativo das consultas realizadas através do protótipo implementado no trabalho.

Os tempos informados para o resultado das consultas sob o protótipo correspondem ao tempo gasto para carregar todos os objetos do resultado visto que não foi implementado um mecanismo de paginação do resultado. Um mecanismo de paginação é muito útil em ferramentas de busca pois mesmo quando o resultado é grande o tempo de reposta para o usuário considera apenas uma página de resultados.

### 6.3 RESULTADOS

Os testes realizados mostram que a aplicação do modelo de recuperação de OAs proposto, permite a reutilização e o compartilhamento de OAs, de diferentes repositórios. Pode também aumentar a cobertura e a precisão na recuperação de objetos, através da utilização de ontologias que possibilitam considerar os relacionamentos semânticos, para os termos consultados. O gráfico da figura 16 apresenta o comparativo, entre os testes realizados com as ferramentas de busca nativas dos repositórios utilizados na validação, que são baseadas em busca por palavras-chave e o protótipo do sistema proposto. É possível observar no gráfico uma melhora na precisão dos objetos recuperados, apesar de ser uma precisão relativa por considerar a amostragem dos resultados, sendo que esta precisão pode ser melhorada, através do aprimoramento das técnicas utilizadas para indexação.

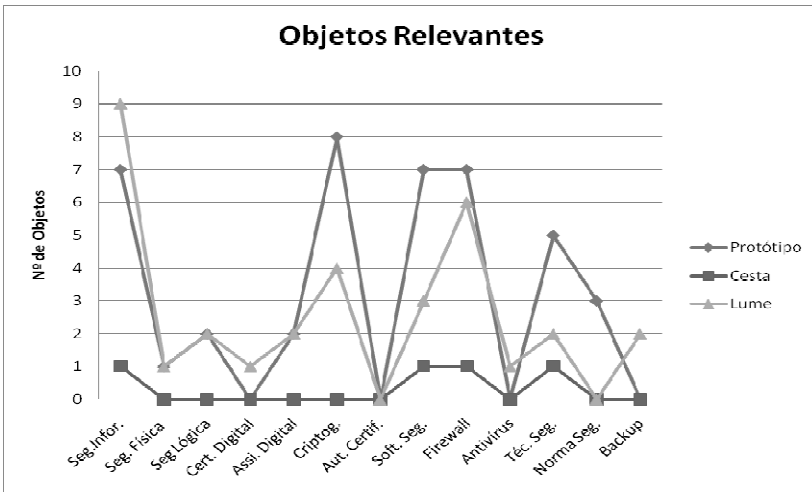


Figura 16: Resultado comparativo

A cobertura atingida com os testes, com a agregação de termos às consultas, através da expansão das mesmas é considerada melhor, porém devido ao volume de objetos resultante da soma dos dois repositórios utilizados, que deveriam ser analisados para se chegar ao número de objetos relevantes no conjunto total de objetos, não permitiu a geração de um gráfico específico para tratar a cobertura.

O desempenho do sistema, visto as limitações do ambiente utilizado e a implementação ter sido realizada em caráter de protótipo é considerado satisfatório se analisado o tempo gasto pelo usuário para efetuar as consultas separadamente nos dois repositórios.



## 7 CONSIDERAÇÕES FINAIS

A utilização da solução proposta no trabalho permite a busca federada em repositórios distribuídos e heterogêneos, como definido por (CORDRA, 2006). A ideia de busca federada de OAs vem sendo pesquisada por (MANEPALLI, et al., 2006; FEB, 2010), sendo que o modelo apresentado neste trabalho não necessita da replicação de OAs já armazenados.

Vários trabalhos têm proposto técnicas e ferramentas com objetivo de permitir maior reutilização de OAs, como em (ZHENG, et al., 2008). Outros trabalhos inclusive utilizando sistemas multiagente e ontologias como (WANG, et al., 2004; PENG, et al., 2005; LEE, et al., 2006; CARBONARO, 2008), tem proposto soluções para problemas de interoperabilidade e recuperação de informação. Porém as características específicas dos trabalhos não resolvem problemas em escalas mais abrangentes, como é caso da reutilização de OAs.

A recuperação de OAs em repositórios distintos e heterogêneos, de forma transparente para o usuário traz incentivo a reutilização e o compartilhamento de OAs.

Com este trabalho é possível concluir que a utilização de OAs é de extrema importância em ambientes de E-learning, e é uma área de grande interesse da comunidade científica. Porém a diversidade de especificações e padrões produzidos nesta área tem trazido problemas para reutilização dos objetos já produzidos, justamente por esta heterogeneidade impossibilitar a interoperabilidade das aplicações utilizadas para recuperação de OAs.

O modelo proposto traz uma solução para o problema encontrado na reutilização e compartilhamento de OAs, através de um sistema multiagente, que possibilita a interoperabilidade entre repositórios de OAs heterogêneos e distribuídos. Além de proporcionar a reutilização de OAs, é possível melhorar a precisão e a cobertura na recuperação de objetos, através da indexação dos mesmos e a utilização de ontologias de domínio, para expansão de consultas tratadas pelo sistema. Também é possível melhorar o desempenho de ferramentas de busca, permitindo a pesquisa no índice do repositório, ao invés de diretamente nos metadados dos objetos armazenados.

A solução apresentada para o cenário de recuperação e reutilização de OAs pode ser adaptada para aplicação em outros cenários, onde exista heterogeneidade e necessidade de interoperabilidade.

O sistema proposto também pode ser integrado com um LMS (Learning Management System), com propósito de utilizar os objetos recuperados, diretamente no processo de ensino e aprendizagem.

## 7.1 TRABALHOS FUTUROS

Criação de ambiente adequado para operação do sistema, com finalidade de melhorar o desempenho e a permitir a realização de testes com usuários finais.

Melhorar o desempenho do sistema através de aprimoramentos na forma de comunicação entre os agentes e na forma de armazenamento do índice.

Criação de mecanismo que permita a inclusão de outros repositórios, inclusive com outros padrões de metadados, de forma parametrizada.

Realizar testes com outros repositórios e outros domínios, com a necessidade de novas ontologias. Com isso, o usuário terá opção de escolher o domínio de interesse, fazendo com que o sistema detecte qual ontologia deve utilizar.

Integrar o sistema proposto, a um LMS permitindo que as consultas possam ser realizadas automaticamente pelo LMS, através de necessidades de informação detectadas ou comandadas, por estudantes ou professores, com a finalidade de atender estratégias pedagógicas.



## 8 REFERÊNCIAS BIBLIOGRÁFICAS

ADL. **Advanced Distributed Learning**, 2010. Disponível em: <<http://www.adlnet.gov/Pages/Default.aspx>>. Acesso em: abr. 2010.

AGENTTOOL. **The agentTool III Project**, 2009. Disponível em: <<http://agenttool.cis.ksu.edu/>>. Acesso em: dez. 2009.

AICC. **Aviation Industry CBT Committee**, 2010. Disponível em: <<http://www.aicc.org/dev/>>. Acesso em: abr. 2010.

APACHE. **Apache Tomcat**, 2010. Disponível em: <<http://tomcat.apache.org/>>. Acesso em: jan. 2010.

ARIADNE. **Ariadne Foundation**, 2006. Disponível em: <<http://www.ariadne-eu.org/>>. Acesso em: abr. 2010.

BARKER, P. **CETIS, Metadata and Digital Repositories**, 2005. Disponível em: <<http://metadata.cetis.ac.uk/guides/WhatIsLOM.pdf>>. Acesso em: nov. 2009.

BELLIFEMINE, F.; CAIRE, G.; GREENWOOD, D. **Developing Multi-Agent Systems with JADE**. Chichester, England: John Wiley & Sons, 2007.

BERGENTI, F.; GLEIZES, M.-P.; ZAMBONELLI, F. **Methodologies and Software Engineering for Agent Systems: The Agent-Oriented Software Engineering Handbook**. Massachusetts: Kluwer Academic Publishers, 2004.

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

BEZ, M. R. et al. **OBAA project: An approach to interoperable learning objects based on Web and digital television**. World Conference on Computers in Education (WCCE 2009), Bento Gonçalves - RS, 2009.

BIOE. **Banco Internacional de Objetos Educacionais**, 2008. Disponível em: <<http://objetoseducacionais2.mec.gov.br/>>. Acesso em: mai. 2010.

BORDINI, R. H.; HUBNER, J. F. **Jason - A Java-based interpreter for an extended version of AgentSpeak**. SourceForge, 2007.

BRADSHAW, J. M. **An introduction to software agents**. Massachusetts: MIT Press, 1997.

- BRAY, T.; PAOLI, J. M., SPERBERG-MCQUEEN C.; MALE, E.; YERGEAU, F. **Extensible Markup Language (XML)**, 2008. Disponível em: <<http://www.w3.org/TR/2008/REC-xml-20081126/>>. Acesso em: jan. 2010.
- CARBONARO, A. **Ontology-based Video Retrieval in a Semantic-based Learning Environment**. Journal of e-Learning and Knowledge Society, Trento, 2008.
- CARDOSO, O. N. P. **Recuperação de Informação**. Revista Infocomp, Lavras-MG, 2000.
- CARTER, G. **LDAP System Administration**. Sebastopol-CA: O'Reilly Media, 2003.
- CESTA. **Coletânea de Entidades de Suporte ao uso de Tecnologia na Aprendizagem**. Centro de Estudos Interdisciplinares em Novas Tecnologias na Educação (CINTED), 2010. Disponível em: <<http://www.cinted.ufrgs.br/CESTA/>>. Acesso em: abr. 2010.
- CORDRA. **CORDRA: Content Object Repository Discovery and Registration/Resolution Architecture**. mar. 2006. Disponível em: <<http://cordra.net/introduction/>>. Acesso em: mar. 2010.
- CURRIER, S. **Metadata for Learning Resources: An Update on Standards Activity for 2008**. Ariadne, 2008. Disponível em: <<http://www.ariadne.ac.uk/issue55/currier/>>. Acesso em: jan. 2010.
- DCMI. **The Dublin Core Metadata Initiative**, 2010. Disponível em: <<http://dublincore.org/>>. Acesso em: mar. 2010.
- DELOACH, S. A.; OJEDA, J. C. G.; VALENZUELA, J.; OYENAN, W. H. **Organization-based Multiagent System Engineering (O-MASE) - Framework Description**. Manhattan: Kansas State University, 2007.
- DOWNES, S. **Learning Objects: Resources for distance education worldwide**. The International Review of Research in Open and Distance Learning, v. 2, n. 1, 2001.
- DSPACE. **DSpace**, 2010. Disponível em: <<http://www.dspace.org/>>. Acesso em: abr. 2010.
- EUZENAT, J.; SHVAIKO, P. **Ontology Matching**. Heidelberg: Springer-Verlag Berlin, 2007.

- FEB. **FEB – Federação Educa Brasil**, 2010. Disponível em: <<http://feb.ufrgs.br/>>. Acesso em: jan. 2010.
- FEDORA. **Fedora Commons Repository Software**, 2010. Disponível em: <<http://www.fedora-commons.org/>>. Acesso em: mai. 2010.
- FIPA. **Foundation for Intelligent Physical Agents**, 2010. Disponível em: <<http://www.fipa.org/>>. Acesso em: dez. 2009.
- GUHA, R.; McCOLL, R.; MILLER, E. **Semantic search**. WWW '03 Proceedings of the 12th international conference on World Wide Web, ACM Press, 2003.
- IEEE-LTSC. **WG12: Learning Object Metadata**. IEEE Learning Technology Standards Committee, 2005. Disponível em: <<http://ltsc.ieee.org/wg12/>>. Acesso em: jun. 2010.
- IMS. **IMS Global Learning Consortium**, 2010. Disponível em: <<http://www.imsglobal.org/>>. Acesso em: mar. 2010.
- ISO. **ISO International Organization for Standardization**, 2010. Disponível em: <[http://www.iso.org/iso/iso\\_catalogue/catalogue\\_tc/catalogue\\_tc\\_browse.htm?commid=45392&development=on](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_tc_browse.htm?commid=45392&development=on)>. Acesso em: abr. 2010.
- JADE. **JADE (Java Agent DEvelopment Framework)**, 2010. Disponível em: <<http://jade.tilab.com/>>. Acesso em: jan. 2010.
- JENA. **Jena – A Semantic Web Framework for Java**, 2010. Disponível em: <<http://jena.sourceforge.net/>>. Acesso em: mar. 2010.
- JENNINGS, N. R. **Coordination Techniques for Distributed Artificial Intelligence**. Foundations of distributed artificial, New York, 1996.
- JISC-CETIS. **Repository systems**. JISC CETIS - Supporting Innovation and Interoperability in Educational Tchenology, 2006. Disponível em: <[http://wiki.cetis.ac.uk/Repository\\_systems](http://wiki.cetis.ac.uk/Repository_systems)>. Acesso em: jan. 2010.
- JISC-CETIS. **Educational metadata standards**. JISC CETIS - Supporting Innovation and Interoperability in Educational Tchenology, 2008. Disponível em: <[http://wiki.cetis.ac.uk/Educational\\_metadata\\_standards](http://wiki.cetis.ac.uk/Educational_metadata_standards)>. Acesso em: mar. 2010.

LACLAVIK, M.; BALOGH, Z.; BABIK, M.; HLUCHY, L. **AGENTOWL: Semantic Knowledge Model and Agent Architecture**. Computing and Informatics, 2006.

LEE, M.-C.; TSAI, K. H.; WANG, T. I. **An Ontological Approach for Semantic-Aware Learning Object Retrieval**. Sixth IEEE International Conference on Advanced Learning Technologies, Netherlands, 2006.

LUME. **Lume - Repositório Digital da Universidade Federal do Rio Grande do Sul**, 2010. Disponível em: <<http://www.lume.ufrgs.br/>>. Acesso em: fev. 2010.

MANEPALLI, G.; JEREZ, H.; NELSON, M. L. **FeDCOR: An Institutional CORDRA Registry**. Reston, v. 12, 2006.

MANGOLD, C. **A survey and classification of semantic search approaches**. Int. J. Metadata, Semantics and Ontology, 2007.

MANNING, C. D.; RAGHAVAN, P.; SCHUTZE, H. **Introduction to Information Retrieval**. England: Cambridge University Press, 2008.

MARTINS, A. B.; RODRIGUES, E.; NUNES, M. B. **Repositórios de informação e ambientes de aprendizagem**. Rede de Bibliotecas Escolares. Minho- PT, 2008.

MCGREAL, R. **Online Education Using Learning Objects**. London: Routledge, 2004.

MERLOT. **MERLOT - Multimedia Educational Resources for Learning and Online Teaching**, 2010. Disponível em: <<http://www.merlot.org/merlot/index.htm>>. Acesso em: mar. 2010.

NASH, S. S. **Learning Objects, Learning Object Repositories, and Learning Theory: Preliminary Best Practices for Online Courses**. Interdisciplinary Journal of Knowledge and Learning Objects, New York, 2005.

OAI. **Open Archives Initiative**, 2010. Disponível em: <<http://www.openarchives.org/>>. Acesso em: abr. 2010.

OBAA. **Portal OBAA**, 2010. Disponível em: <<http://www.portalobaa.org/obaac/padrao-obao/relatorios-tecnicos/RT-OBAA-01.pdf/view>>. Acesso em: mar. 2010.

OJEDA, J. C. G. D. S. A.; ROBBY, O. W. H.; VALENZUELA, J. **O-MaSE: A Customizable Approach to Developing Multiagent**

**Development Processes.** International Workshop on Agent Oriented Software Engineering, Honolulu, 2007.

**O-MASE. Organization-based Multiagent Systems Engineering.** Multiagent & Cooperative Robotics Lab, 2010. Disponível em: <[http://macr.cis.ksu.edu/index.php?option=com\\_content&view=article&id=18](http://macr.cis.ksu.edu/index.php?option=com_content&view=article&id=18)>. Acesso em: dez. 2009.

**OWL-W3C. OWL Web Ontology Language,** 2009. Disponível em: <<http://www.w3.org/TR/owl-features/>>. Acesso em: mar. 2010.

PENG, L.; COLLIER, R.; MUR, A.; LILLIS, D.; TOOLAN, F.; DUNNION, J. **A Self-configuring Agent-Based Document Indexing System.** Multi-Agent Systems and Applications IV, Heidelberg, 2005.

**PROTEGE. Protégé Ontology Editor,** 2010. Disponível em: <<http://protege.stanford.edu/>>. Acesso em: jan. 2010.

REZENDE, D. A. **Engenharia de software e sistemas de informação.** Rio de Janeiro: Brasport, 2005.

**RIVED. RIVED - Rede Interativa Virtual de Educação,** 2010. Disponível em: <[http://rived.mec.gov.br/site\\_objeto\\_lis.php](http://rived.mec.gov.br/site_objeto_lis.php)>. Acesso em: abr. 2010.

RUSSEL, S.; NORVIG, P. **Inteligência Artificial.** Ed. Elviesier, 2<sup>a</sup> ed, 2004.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Sistema de Banco de Dados.** São Paulo: Ed. Makron Books, 1999.

SILVA, J. M. C. **Desenvolvimento de um Framework para Objetos Inteligentes de Aprendizagem aderente a um Modelo de Referência para construção de Conteúdos de Aprendizagem,** 2007. Disponível em: <<http://www.tede.ufsc.br/teses/PGCC0814-D.pdf>>. Acesso em: dez. 2009.

VICARI, R. M. ; GLUZ, J. C. ; PASSERINO, L. ; SANTOS, E. R. ; PRIMO, T. ; ROSSI, L. H. L. ; BORDIGNON, A. ; BEHAR, P. ; FERREIRA FILHO, R. ; ROESLER, V. **The OBAA Proposal for Learning Objects Supported by Agents.** In: International Conference on Autonomous Agents and Multiagent Systems, Toronto, 2010.

WANG, Y.; SHAKSHUKI, E. **A Multi-agent System for Semantic Information Retrieval.** Advances in Artificial Intelligence, Heidelberg, 2004.

WILEY, D. **The Instructional Use of Learning Objects**, 2000.

Disponível em: <<http://reusability.org/read/chapters/wiley.doc>>. Acesso em: jan. 2010.

WOOLDRIDGE, M. **An Introduction to MultiAgent Systems**.

England: John Wiley & Sons, 2002.

XMLRPC. **XML-RPC Home Page**, 2003. Disponível em:

<<http://www.xmlrpc.com/>>. Acesso em: jan. 2010.

ZHENG, L.; LIU, Y.; WANG, J.; YANG, F. **Multiple Standards Compatible Learning Resource Management**. Eighth IEEE

International Conference on Advanced Learning Technologies, 2008.

## ANEXOS

### Anexo 1 - Publicações

VIAN, Jonas; SILVEIRA, Ricardo Azambuja; FILETO, Renato. **Proposal of a Multi-agent System for Indexing and Recovery applied to Learning Objects.** In: 9th IFIP World Conference on Computers in Education, WCCE 2009, Bento Gonçalves-RS, 2009.

VIAN, Jonas; SILVEIRA, Ricardo Azambuja; FILETO, Renato. **Sistema Multiagente para indexação e recuperação aplicado a objetos de aprendizagem.** 4º Conahpa - Congresso Nacional de Ambientes Hiperfídia para Aprendizagem. Florianópolis-SC, 2009.

SILVEIRA, Ricardo Azambuja; VIAN, Jonas. **Sistema Multiagente para indexação e recuperação de objetos de aprendizagem.** SBIE – Simpósio Brasileiro de Informática na Educação. João Pessoa-PB, 2010.