Faculdade de Engenharia da Universidade do Porto

Doutoramento em Engenharia Electrotécnica e de Computadores

A Web Semântica no Contexto Educativo

Um sistema para a recuperação de objectos de aprendizagem baseado nas tecnologias para a Web Semântica, para o e-Learning e para os agentes

Vitor Manuel Barrigão Gonçalves

Mestre em Tecnologia Multimédia pela

Faculdade de Engenharia da Universidade do Porto

Dissertação submetida à Universidade do Porto para cumprimento dos requisitos necessários à obtenção do grau de Doutor em Engenharia Electrotécnica e de Computadores, elaborada sob a orientação do Professor Doutor Eurico Manuel Elias de Morais Carrapatoso do Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto.

Porto, 2007

UNIÃO EUROPEIA

Fundo Social Europeu

O projecto de investigação desta dissertação foi parcialmente financiado pelo Programa de Desenvolvimento Educativo para Portugal (PRODEP III), Eixo 3 – Sociedade da Aprendizagem, Medida 5 (FSE) – Formação de Docentes e outros agentes, Acção 5.3 – Formação Avançada de Docentes do Ensino Superior.

Resumo

A Web pode ser vista como uma mediateca de documentos à escala mundial. Constitui actualmente o maior repositório de informação, disponibilizando conteúdos multimédia, contudo a sua localização não é fácil, devido essencialmente ao facto da sua semântica ou significado apenas poder ser capturada dentro do contexto e na perspectiva dos humanos.

Nos últimos anos, a comunidade científica internacional tem vindo a desenvolver esforços significativos no sentido de melhorar a localização, recuperação e reutilização de objectos de informação, inacessíveis e armazenados em servidores dispersos na Web profunda ou invisível.

Os metadados e as ontologias, as metalinguagens e as ferramentas de anotação e de criação de ontologias e mapas de tópicos, os agentes inteligentes e sistemas de agentes móveis, entre outros avanços tecnológicos das Ciências da Computação e da Inteligência Artificial no âmbito da Gestão da Informação e do Conhecimento e dos Sistemas Distribuídos para a Web, constituem elementos essenciais para o desenvolvimento de soluções que a pouco e pouco contribuirão para alterar a realidade da Web actual.

A iniciativa que mais se tem destacado é a Web Semântica, cujo principal objectivo é a integração, o intercâmbio e a compreensão semântica da informação, tanto na óptica dos humanos, como na óptica das máquinas, através da transformação da Web actual numa Web de informação semântica que tem como particularidade descrever, interrelacionar e compreender os conteúdos através de metadados, ontologias e agentes de software.

Neste contexto, a presente dissertação especifica uma arquitectura para um sistema de recuperação de objectos de aprendizagem baseado nas tecnologias para a Web Semântica, para o e-Learning e para os agentes de software, com o objectivo principal de resolver o problema da descoberta de objectos de aprendizagem e de cursos de formação. De acordo com essa arquitectura, foi também desenvolvido um protótipo experimental para a pesquisa semântica de objectos de aprendizagem armazenados em sistemas de e-Learning, repositórios de objectos de aprendizagem e noutros servidores Web de conteúdos educativos.

Palavras-chave:

Web Semântica; e-Learning; Agentes de Software; Sistemas de Agentes Móveis; Metadados; Ontologias; Mapas de Tópicos; Inferência.

Abstract

The Web can be understood as a multimedia library of documents at world level. It currently consists of the largest information repository which makes multimedia contents available. However, their localization is not an easy task, mainly due to the fact that their semantics or meaning can only be captured in their context and in accordance to human perspective.

In the past years, the international scientific community has been carrying out significant efforts with view to improving the localization, retrieval and reuse of information objects, which may be inaccessible or stored in servers scattered around the deep Web or the invisible Web.

Metadata and ontologies, metalanguages, annotation tools, tools for the creation of ontologies and topic maps, intelligent agents and mobile agent systems, among other technological developments of Computer Sciences and of Artificial Intelligence within the scope of Information and Knowledge Management and Distributed Web Systems, are the key elements for the development of solutions that will gradually lead to changing the present Web reality.

The project that has been receiving more attention is the Semantic Web whose main purpose is the integration, interchange and semantic understanding of information not only from the viewpoint of humans, but also from the perspective of machines by means of the change of the current Web into a Web of semantic data. This would then allow for the description, interrelation and understanding of contents through metadata, ontologies and software agents.

In this context, the present thesis specifies the architecture for a retrieval system of learning objects based on the technologies of the Semantic Web, e-Learning and software agents, aiming to solve the problem of determining learning objects and training courses. According to this architecture, we also developed an experimental prototype for the semantic search of learning objects stored in e-Learning systems, learning object repositories and other Web servers for educational contents.

Keywords:
Semantic Web; e-Learning; Software Agents; Mobile Agent Systems; Metadata; Ontologies; Topic Maps; Inference.

Résumé

Le Web peut être vu comme une médiathèque de documents à l'échelle mondiale. Il constitue actuellement le plus grand recueil d'information, mettant à notre disposition des contenus multimédia ; toutefois, sa localisation n'est pas facile en raison essentiellement du fait que sa sémantique ou son sens ne peuvent être saisis que dans le contexte et la perspective des humains.

Au cours des dernières années, la communauté scientifique internationale a développé des efforts considérables dans le sens d'améliorer la localisation, la récupération et la réutilisation d'objets d'information inaccessibles et emmagasinés dans des serveurs dispersés dans le Web profond ou invisible.

Les métadonnées et les ontologies, les métalangages et les outils d'annotation et de création d'ontologies et de cartes de topiques, les agents intelligents et les systèmes d'agents mobiles, entre autres progrès technologiques des Sciences Informatiques et de l'Intelligence Artificielle dans le cadre de la Gestion de l'Information et de la Connaissance et des Systèmes Distribués pour le Web, constituent des éléments essentiels pour le développement de solutions qui, peu à peu, contribueront pour altérer la réalité du Web actuel.

L'initiative qui s'est le plus fait remarquer est le Web Sémantique dont l'objectif principal est l'intégration, l'échange et la compréhension sémantique de l'information, aussi bien du point de vue des humains que du point de vue des machines, à travers la transformation du Web actuel en un Web d'information sémantique qui a comme particularité décrire, mettre en rapport et comprendre les contenus à travers des métadonnées, des ontologies et des agents de software.

Dans ce contexte, la présente dissertation précise une architecture pour un système de récupération d'objets d'apprentissage basé sur les technologies pour le Web Sémantique, pour le e-Learning et pour les agents de software, ayant comme but principal résoudre le problème de la découverte d'objets d'apprentissage et de cours de formation. Selon cette architecture, un prototype expérimental a également été développé pour la recherche sémantique d'objets d'apprentissage emmagasinés dans des systèmes de e-Learning, recueils d'objets d'apprentissage et dans d'autres serveurs Web de contenus éducatifs.

Mots-clé:

Web Sémantique ; e-Learning ; Agents de Software ; Systèmes d'Agents Mobiles ; Métadonnées ; Ontologies ; Cartes de Topiques ; Inférence.

Agradecimentos

Ao longo destes últimos três anos e meio em que estive envolvido na concepção da arquitectura e desenvolvimento do protótipo do projecto de investigação e na redacção desta dissertação, tive a oportunidade de contar com o apoio de diversas pessoas que, directa ou indirectamente, contribuíram para o presente documento.

Em primeiro lugar, agradeço ao Professor Eurico Carrapatoso, meu orientador, pela sua permanente disponibilidade e dedicação, pelo seu sentido crítico, pelo pragmatismo e pelas valiosas opiniões e sugestões na organização e revisão deste documento, bem como pela simplicidade na relação humana entre orientador-orientando e pela constante atitude motivadora sem os quais este projecto de investigação dificilmente teria sido concluído.

Ao Professor Ribeiro Alves (IPB) não só pelas interessantes conversas sobre as teorias da aprendizagem, pela bibliografia recomendada e pela revisão dessa parte da tese, mas também pelos constantes votos de incentivo.

À Professora Paula Oliveira (UTAD), pelas importantes sugestões e pela disponibilidade no esclarecimento de dúvidas no âmbito da exploração das tecnologias de agentes móveis.

Ao Dr. António Alves (IPB), pela disponibilidade e pela partilha de experiências e de informação sobre as ciências documentais, em geral, e os instrumentos de indexação e controlo terminológico, em particular.

À Sofia Rodrigues, não só pela disponibilidade demonstrada na utilização e avaliação dos sistemas de e-Learning, mas também pelo constante incentivo e pelos momentos de convívio e de investigação no âmbito da aplicação da Inteligência Artificial na Educação.

À Esmeralda Gonçalves, pela cuidada revisão da ortografia e gramática do texto do presente documento.

Aos professores e alunos da Escola Superior de Educação de Bragança e do Ensino Básico e Secundário, pela sua colaboração no âmbito da exploração e avaliação dos sistemas de e-Learning e dos protótipos desenvolvidos no contexto deste projecto de investigação.

A todos aqueles que despenderam algum do seu precioso tempo para esclarecer as dúvidas de um desconhecido que dava os primeiros passos na exploração de tecnologias

emergentes. Os contributos de Sven Schwarz no âmbito das linguagens RDF e OWL, de Thomas Wheeler e Jim Bender da *Recursion Software* (Voyager), de Pete Johnston da *Dublin Core Metadata Iniciative*, de Steve Pepper e Alexander Johannesen no âmbito dos mapas de tópicos e os de Martin Dougiamas, Ger Tielemans, Olivier Catteau e Filipe Pablos no âmbito da plataforma *Moodle*, entre outros membros das comunidades *Moodle*, *Mambo*, *Joomla!*, que, por mais breve que tenha sido a interacção electrónica, marcaram o rumo deste projecto.

Aos meus colegas de departamento, Raquel Patrício, Manuel Meirinhos e Sérgio Sousa não só pelo constante incentivo, mas também por assegurarem o regular funcionamento de projectos nos quais estava envolvido antes de iniciar este doutoramento.

Ao IPB pela concessão de dispensa de serviço docente, sem a qual o desenvolvimento deste trabalho teria sido complicado e, em particular, ao Conselho Directivo da ESEB, pelas condições proporcionadas para a sua concretização.

Por último, não posso deixar de manifestar o meu apreço pela infinita paciência e apoio da minha família e, em particular, dos meus pais, da minha mulher Esmeralda e do meu filho Francisco, aos quais fico grato não só pelas brincadeiras e outros momentos que me deram ânimo para continuar, mas também pela compreensão nestes últimos meses quando deixei de lhes dar a atenção que mereciam.

A todos, os meus sinceros agradecimentos.

Índice

1- Introdução 1
1.1- Contexto e Motivação 5
1.2- Finalidade, Objectivos, Inovação e Contributos 6
1.3- Estrutura do Documento 10
2- Tecnologias para o e-Learning 13
2.1- Sociedade do Primado do Saber 14
2.1.1- A Sociedade da Informação e do Conhecimento em Portugal 15
2.1.2- O Processo de Bolonha e o e-Learning 19
2.2- Ensino a Distância e e-Learning 21
2.3- Plataformas para os Sistemas de e-Learning 29
2.4- Modelo Pedagógico para o e-Learning 32
2.4.1- Teorias da Aprendizagem 32
2.4.2- Modelos para a Estruturação de Conteúdos 37
2.5- Tecnologias ao Serviço do e-Learning 40
2.5.1- ADL SCORM e Objectos de Aprendizagem 42
2.5.1.1- Objectos de Aprendizagem 43
2.5.1.2- Modelo SCORM 45
2.5.2- Instructional Design e Learning Design 49
2.5.2.1- Instructional Design 51
2.5.2.2- Learning Design 53
2.5.3- Mapas Mentais, Mapas de Conceitos e Mapas de Tópicos 57
2.5.3.1- Mapas Mentais 57
2.5.3.2- Mapas de Conceitos 59
2.5.3.3- Mapas de Tópicos 61
2.6- Resumo 62
3- Camadas Estrutural e Sintáctica da Web Semântica 65
3.1- Rumo à Web Semântica 65
3.1.1- Evolução da Web 66
3.1.2- Principais Limitações da Web 68
3.2- A Web Semântica 74
3.2.1- As Mudanças da Web Actual para a Web Semântica 74
3.2.2- Arquitectura para a Web Semântica 77

3.3- Tecnologias para as Camadas Estrutural e Sintáctica 83
3.3.1- XML e XML Schema 83
3.3.2- Metadados 88
3.3.2.1- Dublin Core Metadata 90
3.3.2.2- Learning Object Metadata 93
3.3.2.3- Multimedia Content Description Interface 98
3.3.3- Resource Description Framework 101
3.3.4- Interoperabilidade de Esquemas de Metadados 109
3.4- Resumo 112
4- Camadas Semântica e Lógica da Web Semântica 113
4.1- Representação do Conhecimento 113
4.1.1- Tecnologias Convencionais 115
4.1.1.1- Dicionários e Índices 116
4.1.1.2- Vocabulários Controlados 117
4.1.1.3- Taxonomias 117
4.1.1.4- Tesauros 120
4.1.1.5- Classificações Facetadas 120
4.1.2- Tecnologias para a Camada Semântica 121
4.1.2.1- RDF Schema 128
4.1.2.2- Web Ontology Language 131
4.1.2.3- Interoperabilidade de Ontologias 139
4.1.3- Mapas de Tópicos 140
4.1.3.1- Norma Topic Maps 143
4.1.3.2- Comparação entre RDF e Topic Maps 148
4.2- Tecnologias para a Camada Lógica 150
4.2.1- Linguagens para as Regras de Inferência 151
4.2.2- Linguagens para os Mecanismos de Inferência 154
4.3- Resumo 156
5- Tecnologias para os Agentes 159
5.1- Agentes de Software 159
5.1.1- Atributos dos Agentes 162
5.1.2- Classificação dos Agentes 163
5.1.3- Agentes de Software para a Internet 166
5.1.4- Agentes de Software Inteligentes e Móveis 170
5.2- Sistemas de Agentes Móveis 174

5.2.1- Arquitectura Genérica de um Sistema de Agentes Móveis 174
5.2.2- Infra-estruturas para a Criação de Aplicações Baseadas em Agentes 177
5.3- Resumo 179
6- Arquitectura do Sistema WSE 181
6.1- Planeamento e Requisitos 182
6.1.1- Actividade de Planeamento 182
6.1.2- Análise e Especificação de Requisitos 187
6.2- Modelação do Contexto e dos Requisitos 189
6.2.1- Casos de Utilização 190
6.2.2- Arquitectura de Alto Nível para o Sistema WSE 194
6.2.3- Modelação dos Requisitos de um Sistema de e-Learning 198
6.2.4- Arquitectura de Alto Nível para um Sistema de e-Learning 201
6.3- Modelação do Comportamento 203
6.4- Modelação da Estrutura 207
6.5- Modelação da Arquitectura 208
6.5.1- Agente de Interface 212
6.5.2- Servidor de Agentes 216
6.5.3- Agente de Informação 225
6.6- Resumo 229
7- Desenvolvimento do Sistema WSE 231
7.1- Actividade de Desenvolvimento 231
7.2- Protótipo do Sistema WSE 238
7.2.1- Agente de Interface 243
7.2.2- Servidor de Agentes 247
7.2.3- Agente de Pesquisa 250
7.2.4- Agente de Informação 254
7.2.5- Utilitários para Metadados, Ontologias e Mapas de Tópicos 258
7.3- Sistema de e-Learning Moodle 267
7.3.1- Módulos de Metadados para os LOs 276
7.3.2- Módulos de Metadados para as Disciplinas 283
7.3.2- Bloco de Pesquisa Semântica 285
7.4- Website do Professor para a Web Semântica 289
7.5- Avaliação do protótipo do Sistema WSE 293
7.6- Resumo 302
8- Considerações Finais 303

8.1- Retrospectiva do Trabalho Realizado 304
8.2- Comparação com outros Trabalhos ou Projectos 308
8.3- Perspectiva do Trabalho Futuro 312
8.4- Conclusão 314
Referências Bibliográficas e Webliográficas 317
Anexos 333
Anexo A – Mapa de Conceitos da Tese 334
Anexo B – Cérebro Global 335
Anexo C – Aplicação das Tecnologias para a Web Semântica 345
Anexo D – Linguagens para as Ontologias 354
Anexo E – Ferramentas para as Ontologias 356
Anexo F – Plataformas para o Desenvolvimento de Aplicações Baseadas em Agentes ... 360
Anexo G – Ontologia para o Sistema WSE 365
Anexo H – Ontologia OWL para o Sistema WSE 366
Anexo I – Geração (semi)automática de metadados DCM 369
Anexo J – Geração (semi)automática de metadados LOM 371
Anexo K – Inquéritos de Avaliação 376

Índice de Figuras

Figura 1 – Programa Operacional Sociedade do Conhecimento 16
Figura 2 – Agência para a Sociedade do Conhecimento (UMIC) 18
Figura 3 – Gerações de EAD 23
Figura 4 – Framework para e-Learning 36
Figura 5 – Factores essenciais no desenvolvimento de um Sistema de e-Learning 39
Figura 6 – Metáfora LEGO aplicada aos LOs 43
Figura 7 – Manuais técnicos do SCORM 46
Figura 8 – Editor RELOAD 47
Figura 9 – Editor eXe 48
Figura 10 – Editor de metadados LomPad 48
Figura 11 – Editor SCORMxt 49
Figura 12 – Projecto Instrucional 51
Figura 13 – Pacotes no meta-modelo pedagógico 52
Figura 14 – Editor RELOAD Learning Design 55
Figura 15 – Um Mapa Mental sobre “Mapas Mentais” editado com InteliMap 58
Figura 16 – Um Mapa de Conceitos sobre “Mapa de Conceitos” editado com CmapTools ... 60
Figura 17 – Modelo Cliente-Servidor na Internet 67
Figura 18 – Uma pesquisa de cursos de formação no Google 71
Figura 19 – Web Semântica como extensão da Web actual 75
Figura 20 – Visão geral da Web Semântica 77
Figura 21 – Arquitectura para a Web Semântica (versão 2000) 78
Figura 22 – Arquitectura para a Web Semântica (versão 2002) 80
Figura 23 – Arquitectura para a Web Semântica (versão 2005) 81
Figura 24 – Exemplo básico de XML 84
Figura 25 – Exemplo básico de DTD 85
Figura 26 – Exemplo básico de XML Schema 85
Figura 27 – Exemplo básico de XSL 87
Figura 28 – Esquema genérico de metadados DCMES em XML 92
Figura 29 – Esquema genérico de metadados LOM em XML 96
Figura 30 – Principais elementos MPEG-7 99
Figura 31 – Contexto da norma MPEG-7 100
Figura 32 – Grafo de um Modelo de Dados RDF básico 104
Figura 33 – Ambiente gráfico do IsaViz 105

Figura 34 – Exemplo básico de RDF 105
Figura 35 – Grafo de um Modelo de Dados RDF 106
Figura 36 – Exemplo de RDF 107
Figura 37 – Uma abordagem para gerar instâncias de metadados interoperáveis 110
Figura 38 – Modelo para combinar DCM e LOM através de RDF 111
Figura 39 – Modelo mental genérico 114
Figura 40 – Exemplo de uma taxonomia para os seres vivos 118
Figura 41 – Exemplo de uma taxonomia para a Gestão do Conhecimento 119
Figura 42 – Relação entre Dicionário, Índice, Taxonomia, Tesauro e Ontologia 122
Figura 43 – Componentes básicos de uma ontologia 124
Figura 44 – Modelo conceptual de uma ontologia parcial 125
Figura 45 – Mapa conceptual de uma possível ontologia para José Saramago 126
Figura 46 – Classes e propriedades do esquema RDFS 129
Figura 47 – Um exemplo de aplicação de RDF e RDFS 130
Figura 48 – Um exemplo de aplicação de RDFS 130
Figura 49 – Exemplo básico de início de documento OWL 131
Figura 50 – Exemplo de declaração de classes e subclasses em OWL 131
Figura 51 – Exemplo de declaração de propriedades em OWL 132
Figura 52 – Quadrantes Semânticos 133
Figura 53 – Árvore das Tecnologias do Conhecimento 136
Figura 54 – Independência entre mapas de tópicos e recursos de informação 142
Figura 55 – Exemplo básico de XTM 143
Figura 56 – Definição dos tipos de tópicos em XTM 144
Figura 57 – Definição do contexto em XTM 144
Figura 58 – Sintaxe da XTM 146
Figura 59 – As famílias de normas W3C RDF e ISO Topic Maps 149
Figura 60 – Uma limitação da OWL 151
Figura 61 – Sintaxe abstracta da SWRL 153
Figura 62 – Um exemplo com regras em SWRL 154
Figura 63 – Arquitectura genérica de uma aplicação com motor de inferência SPARQL 155
Figura 64 – Áreas científicas influentes na tecnologia para os agentes 160
Figura 65 – Modelo de agente 161
Figura 66 – Classificação de agentes baseada na presença de atributos primários 165
Figura 67 – Sistemas distribuídos baseados em agentes 171
Figura 68 – Evolução do paradigma dos agentes móveis 172

Figura 69 – Arquitectura genérica de um sistema de agentes móveis 177
Figura 70 – Integração das Tecnologias para a Web Semântica, e-Learning e agentes 180
Figura 71 – Modelo genérico da actividade de planeamento 183
Figura 72 – Modelo genérico da actividade de análise de requisitos 188
Figura 73 – Modelação do contexto actual 189
Figura 74 – Modelação do contexto do sistema proposto 190
Figura 75 – Diagrama UML de casos de utilização 191
Figura 76 – Casos de utilização: procurar LOs 192
Figura 77 – Casos de utilização: submeter/editar curso 193
Figura 78 – Casos de utilização: visualizar cursos ou LOs 194
Figura 79 – Arquitectura genérica do Sistema WSE 195
Figura 80 – Casos de utilização: sistema de e-Learning 200
Figura 81 – Casos de utilização: procurar LOs num sistema de e-Learning 201
Figura 82 – Arquitectura genérica dos módulos e blocos para um sistema de e-Learning 202
Figura 83 – Diagrama de estados do agente de interface 204
Figura 84 – Diagrama de actividade do servidor de agentes móveis 205
Figura 85 – Diagrama de actividade do agente de informação 206
Figura 86 – Modelação da estrutura do servidor de agentes 207
Figura 87 – Arquitectura do Sistema WSE e dos sistemas remotos 209
Figura 88 – Comunicação entre agente de interface e servidor de agentes 212
Figura 89 – Diagrama de actividade: autenticação do utilizador 213
Figura 90 – Diagrama de actividade: criação de um agente de pesquisa móvel 214
Figura 91 – Diagrama de actividade: destruição de um agente de pesquisa móvel 215
Figura 92 – Arquitectura do servidor de agentes 217
Figura 93 – Diagrama de classes: base de dados do sistema WSE 220
Figura 94 – Diagrama de actividade: objecto AgentMServ 223
Figura 95 – Diagrama de actividade: objecto AgentLOServ 224
Figura 96 – Diagrama de actividade: agente de pesquisa móvel 225
Figura 97 – Arquitectura do agente de informação para um sistema remoto 226
Figura 98 – Diagrama de actividade: objecto AMPServ 228
Figura 99 – *Multicasting* de mensagens na plataforma Voyager 242
Figura 100 – *Multicasting* de mensagens para um sub-espaço Voyager 242
Figura 101 – Acesso de uma *applet* à arquitectura distribuída do servidor de agentes 244
Figura 102 – Autenticação de um utilizador 244
Figura 103 – Agente de interface: pesquisa básica 246

Figura 104 – Servidor de agentes: gestão de utilizadores 248
Figura 105 – Servidor de agentes: gestão de sistemas externos 249
Figura 106 – Ciclo de vida genérico de um agente de pesquisa 251
Figura 107 – Aplicação do servidor de agentes e aplicações dos agentes de informação 257
Figura 108 – Modelo da ontologia geral para o Sistema WSE 260
Figura 109 – Modelo da ontologia para os sistemas de e-Learning 260
Figura 110 – Protégé: ontologia para os sistemas de e-Learning 261
Figura 111 – Protégé: criação de uma propriedade 261
Figura 112 – Protégé: editor de instâncias da ontologia para os sistemas de e-Learning 264
Figura 113 – Actividade de programação do projecto WSE no IDE Netbeans 265
Figura 114 – Bibliotecas e ambiente de execução do IDE Netbeans 266
Figura 115 – Diferentes alternativas de instalação e configuração do Moodle e Atutor 268
Figura 116 – Directório do módulo modelo 272
Figura 117 – Adicionar metadados a LOs no Moodle 277
Figura 118 – Processo de adição de metadados LOM a um recurso 278
Figura 119 – Metadados LOM adicionados automaticamente a um recurso 281
Figura 120 – Vistas disponibilizadas pelos índices dos módulos de metadados 282
Figura 121 – Módulos de programas e de sumários com metadados 284
Figura 122 – Bloco de pesquisa: pesquisa básica 286
Figura 123 – Bloco de pesquisa: pesquisa avançada 287
Figura 124 – Bloco de pesquisa: pesquisa semântica 289
Figura 125 – Página do professor para a Web Semântica 291
Figura 126 – Resultado de uma pesquisa semântica 292
Figura 127 – Aplicação das tecnologias para a Web Semântica 293
Figura 128 – Analogia entre Cérebro Humano e Rede Internet 338
Figura 129 – Regra de Hebb 340
Figura 130 – Álbuns de música: código HTML 345
Figura 131 – Álbuns de música: código XML 346
Figura 132 – Álbuns de música: código RDF 346
Figura 133 – Álbuns de música: projecto no Protégé 347
Figura 134 – Álbuns de música: ficheiro RDFS 348
Figura 135 – Álbuns de música: ficheiro RDF 349
Figura 136 – Álbuns de música: optimização do projecto no Protégé 349
Figura 137 – Álbuns de música: adicionar instâncias no Protégé 353
Figura 138 – Álbuns de música: realizar inferência no Protégé 353

Índice de Tabelas:

Tabela 1 – Comparação entre LMS e LCMS 41
Tabela 2 – Caracterização da Web 76
Tabela 3 – Grupos de elementos DCMES 91
Tabela 4 – Comparação entre SWRL e WRL 153
Tabela 5 – Plataformas de agentes móveis 178
Tabela 6 – Classes e interfaces do servidor de agentes 221
Tabela 7 – Classes e interfaces do agente de informação de um sistema remoto 227
Tabela 8 – Classes, subclasses e atributos da ontologia para o Sistema WSE 262
Tabela 9 – Classes, subclasses e atributos da ontologia para os sistemas de e-Learning 263

Lista de acrónimos:

ADL	*Advanced Distributed Learning*
AGLS-MES	*AGLS Metadata Element Set*
AICC	*Aviation Industry CBT (Computer-Based Training) Committee*
aLFanet	*Active Learning For Adaptative interNET*
AML	*Agent Modelling Language*
ANSI	*American National Standards Institute*
AORML	*Agent-Object-Relationship Markup Language*
AOSE	*Agent Oriented Software Engineering*
API	*Application Programming Interface*
ASP	*Active Server Pages*
AUML	*Agent Unified Modelling Language*
BD	*Base de Dados*
CAI	*Computer Assisted Instruction*
CBI	*Computer-Based Instruction*
CBL	*Computer-Based Learning*
CBML	*Case Based Markup Language*
CBT	*Computer-Based Training*
CD-ROM	*Compact Disk - Read Only Memory*
CERN	*Conseil Européen pour la Recherche Nucléaire*
CET	*Curso de Especialização Tecnológica*
CLE	*Constructivist Learning Environments*
CMS	*Content Management Systems*
CORBA	*Common Object Request Broker Architecture*
CPU	*Central Processing Unit*
CSS	*Cascading Style Sheets*
DCB	*Diploma de Competências Básicas em TICs*
DC	*Dublin Core*
DCM	*Dublin Core Metadata*
DCMES	*Dublin Core Metadata Element Set*
DCMI	*Dublin Core Metadata Initiative*
DCOM	*Distributed Component Object Model*
DDC	*Dewey Decimal Classification*

DGES	*Direcção Geral do Ensino Superior*
DKS	*Distributed Knowledge Systems*
DL	*Distance Learning*
DLL	*Dynamic Link Library*
DLP	*Description Logic Programs*
DNS	*Domain Name Server*
Ds	*Descriptors*
DSs	*Description Schemes*
DTD	*Document Type Definition*
DVD	*Digital Versatile Disc*
EAD	*Ensino a Distância / Educação a Distância*
EML	*Educational Markup Language*
E-R	*Entidade-Relação*
ERIC	*Education Resources Information Center*
ERP	*Enteprise Resource Planning*
ESEB	*Escola Superior de Educação de Bragança*
ESE-IPB	*Escola Superior de Educação - Instituto Politécnico de Bragança*
eU	*Electronic University / Universidade Electrónica*
FCCN	*Fundação para a Computação Científica Nacional*
FEUP	*Faculdade de Engenharia da Universidade do Porto*
FOAF	*Friend Of A Friend*
FOCO	*Formação Contínua de Professores*
FTP	*File Transfer Protocol*
GIF	*Graphic Interchange Format*
GIM	*Geographic Information Metadata*
GPS	*Global Positioning System*
GRDDL	*Gleaning Resource Descriptions from Dialects of Languages*
HTML	*Hypertext Markup Language*
HTTP	*Hypertext Transfer Protocol*
IBL	*Internet-based Learning*
ID	*Instructional Design*
IDC	*International Data Group*
IDT	*Instructional Design on Technology*
IEC	*International Electro-technical Commission*
IEEE	*Institute of Electrical and Electronics Engineers*

IEEE/LTSC	*IEEE Learning Technology Standards Committee*
IEEE-LOM	*IEEE - Learning Object Metadata*
IIS	*Internet Information Server*
ILS	*Internet Locator Server*
IMAP	*Internet Message Access Protocol*
IMS	*Instructional Management Systems*
IMS/GLC	*Instructional Management Systems Global Learning Consortium*
IMS-CP	*IMS Content Packaging*
IMS-LD	*IMS Learning Design*
IMS-LIP	*IMS Learner Information Package*
IMS-LRM	*IMS Learning Resource Metadata*
IMS-QTI	*IMS Question and Test Interoperability*
IMS-RCD	*IMS Reusable Competency Definition*
IMS-SS	*IMS Simple Sequencing*
IP	*Internet Protocol*
IPB	*Instituto Politécnico de Bragança*
IRC	*Internet Relay Chat*
ISO	*International Standards Organization*
IST	*Information Society Technology*
JDBC	*Java DataBase Connectivity*
JDK	*Java Development Kit*
JISC	*Joint Information Systems Committee*
JNDI	*Java Naming and Directory Interface*
JPEG	*Joint Photographic Experts Group*
JQIG	*Joint Quality Iniciative Group*
JSP	*Java Server Pages*
KQML	*Knowledge Query and Manipulation Language*
LAMS	*Learning Activity Management System*
LCC	*Library of Congress Classification*
LCMS	*Learning Content Management System*
LCSH	*Library of Congress Subject Headings*
LD	*Learning Design*
LDAP	*Lightweight Data Access Protocol*
LMS	*Learning Management System*
LO	*Learning Objects*

LOM	*Learning Object Metadata*
LOMWG	*Learning Objects Metadata Working Group*
MASIF	*Mobile Agent System Interoperability Facility*
MCT	*Ministério da Ciência e Tecnologia*
MDIS	*Meta Data Interchange Specification*
MDR	*Metadata Registries*
MESH	*Medical Subject Headings*
MIME	*Multipurpose Internet Mail Extensions*
MMMI	*Model for Metadata for Multimedia Information*
MPEG	*Moving Picture Experts Group*
MPEG-7	*Multimedia Content Description Interface*
MSI	*Missão para a Sociedade da Informação*
N3	*Notation 3*
NCSA	*National Center for Supercomputing Applications*
NLMC	*National Library of Medicine Classification*
OCDE	*Organização para a Cooperação e Desenvolvimento Económico*
ODBMS	*Object Oriented Database Management System*
OKS	*Ontopia Knowledge Suite*
OL	*Online Learning*
OLE	*Open Learning Environments*
OMG	*Object Management Group*
OOSD	*Object-oriented System Development*
OPM/MAS	*Object-Process Methodology for Multi-Agent System*
ORB	*Object Request Broker*
OWL	*Web Ontology Language*
P2P	*Peer-to-Peer*
PC	*Personal Computer*
PDA	*Personal Digital Assistant*
PDF	*Portable Document Format*
PGP	*Pretty Good Privacy*
PHP	*Hypertext Preprocessor*
PICS	*Platform for Internet Content Selection*
POSC	*Programa Operacional da Sociedade do Conhecimento*
POSI	*Programa Operacional para a Sociedade da Informação*
PPT	*Power Point (Microsoft)*

RAP	*RDF API for PHP*
RCTS	*Rede de Ciência, Tecnologia e Sociedade*
RDF	*Resource Description Framework*
RDFS	*RDF Schema*
RDQL	*RDF Data Query Language*
REV	*Remote Evaluation*
RFML	*Relational-Functional Markup Language*
RIF	*Rule Interchange Format*
RMI	*Remote Method Invocation*
RPC	*Remote Procedure Calls*
RSS	*RDF Site Summary*
RuleML	*Rule Markup Language*
SAM	*Sistema de Agentes Móveis*
SCO	*Shareable Content Packing*
SCORM	*Sharable Content Object Resource Model*
SGML	*Standard Generalized Markup Language*
SHOE	*Simple HTML Ontology Extension*
SPARQL	*RDF Query Language and Protocol*
SQL	*Structured Query Language*
SWRL	*Semantic Web Rule Language*
TAO	*Topic-Occurrence-Association*
TCP	*Transmission Control Protocol*
TCP/IP	*Transmission Control Protocol / Internet Protocol*
TI	*Tecnologia de Informação*
TIC	*Tecnologia de Informação e Comunicação*
TM4J	*Topic Maps for Java*
TMCL	*Topic Map Constraint Language*
TMQL	*Topic Map Query Language*
uARTE	*Unidade de Apoio à Rede Telemática Educativa*
UCS	*Universal Character Set*
UDC	*Universal Decimal Classification*
UDP	*Unit Datagram Protocol*
UMIC	*Unidade de Missão Inovação e Conhecimento*
UML	*Unified Modelling Language*
UNESCO	*United Nations Educational, Scientific and Cultural Organization*

UoL	*Unit of Learning*
UP	*Universidade do Porto*
URI	*Uniform Resource Identifier*
URL	*Uniform Resource Locator*
URML	*Universal Rule Markup Language*
UTF	*Unicode (ou UCS) Transformation Format*
VLE	*Virtual Learning Environments*
W3C	*World Wide Web Consortium*
WAP	*Wireless Application Protocol*
WBL	*Web-based Learning*
Web	*World Wide Web*
WML	*Wireless Markup Language*
WRL	*Web Rule Language*
WS	*Web Semântica*
WSE	*Web Semântica Educativa*
WWW	*World Wide Web*
XCM	*XML Concept Maps*
XML	*eXtensible Mark-up Language*
XMLS	*XML Schema*
XMP	*eXtensible Metadata Platform*
XRML	*eXtensible Rule Markup Language*
XSD	*XML Schema Definition*
XSL	*eXtensible Stylesheet Language*
XSL	*eXtensible Stylesheet Language*
XSLT	*XSL Transformations*
XTM	*XML Topic Maps*

1- Introdução

Actualmente, os livros, manuais escolares ou outros documentos impressos já não são o único suporte documental do processo de ensino/aprendizagem, sendo em muitos casos complementados por documentos digitais. Estes documentos apresentam relativamente aos documentos em papel claras vantagens: integram múltiplos formatos de media digital; têm maior disponibilidade; são duradouros; podem ser mantidos, actualizados e duplicados facilmente; oferecem melhor suporte a actividades colaborativas e cooperativas; e facilitam a pesquisa de informação. Estas características aliadas às potencialidades das redes digitais impulsionam o acréscimo de fluxos de informação.

Mas a crescente circulação de informação conduz a uma constante desactualização dos conhecimentos e competências. A formação obtida na Escola rapidamente se torna obsoleta, pelo que se exige uma constante e rápida actualização dos conhecimentos para que possamos sobreviver profissionalmente nesta nova sociedade, uma Sociedade da Informação e do Conhecimento. Exige-se um novo conceito de educação: educação ao longo da vida.

A Internet constitui uma das mais importantes ferramentas não só para professores e alunos em geral, mas também para todos aqueles que enveredam por processos de formação contínua, uma vez que permite distribuir e actualizar informação rapidamente, criar comunidades virtuais de aprendizagem, favorecendo a comunicação individual ou em grupo, flexibilizar o acesso a materiais educativos e apoiar a auto-aprendizagem. Com a tecnologia Internet, surgiu um novo modelo de acesso à informação e de comunicação entre os seres humanos e, consequentemente, um novo modelo de formação e educação.

A *World Wide Web* (WWW ou simplesmente Web) pode comparar-se a uma gigantesca mediateca de recursos ao dispor de todos aqueles que tenham acesso a um equipamento com ligação à Internet. Qualquer cidadão pode ter ao seu dispor sistemas hipermédia e aplicações de ensino a distância ou outros ambientes de aprendizagem electrónica aos quais pode aceder de forma intuitiva e, a muitos deles, gratuitamente.

O e-Learning, sendo uma modalidade de aprendizagem electrónica, pode considerar-se um tipo de ensino a distância, desde que baseado nas tecnologias da Internet, onde a aprendizagem ocorre remotamente através, essencialmente, da interacção com conteúdos e materiais de aprendizagem e da interacção com outros elementos do processo educativo (formadores e formandos).

Falamos de formação ou ensino a distância (ou simplesmente EAD) quando estamos perante um processo de ensino/aprendizagem em que o professor e o aluno estão geograficamente distantes e em que a interacção entre ambos é, preferencialmente, estabelecida via meios electrónicos.

Gradualmente, a interiorização do conceito de educação ao longo da vida tem vindo a impulsionar novas formas de auto-aprendizagem, nomeadamente: a aprendizagem através de conteúdos organizados e disponibilizados em sistemas de e-Learning ou outros ambientes de aprendizagem Web e a aprendizagem através de conteúdos Web dispersos localizados por motores de busca.

No que diz respeito à aprendizagem recorrendo a sistemas de e-Learning, a descoberta dos cursos e dos conteúdos que os constituem (mesmo os de acesso livre) raramente é possível a menos que conheçamos a plataforma e façamos parte dela enquanto alunos ou professores. O conteúdo é nuclear em praticamente todas as soluções de e-Learning, mas o seu intercâmbio, pesquisa, reutilização e personalização raramente são tidos em consideração. As experiências e aplicações iniciais do e-Learning preocuparam-se, essencialmente, com questões espaciais e temporais (disponível em qualquer lugar, a qualquer hora), relegando as questões pedagógicas e de organização, estruturação e interoperabilidade dos conteúdos para segundo plano. Por conseguinte, as potencialidades das tecnologias para o e-Learning por si só não serão suficientes para suportar o desenvolvimento de cursos para e-Learning, a menos que sejam articuladas com orientações inerentes à organização dos conteúdos, ao projecto de aprendizagem e às teorias da aprendizagem.

Quanto à aprendizagem recorrendo a motores de busca, as estratégias actuais de indexação e de recuperação de informação mostram-se insatisfatórias e ineficazes face aos requisitos da sociedade contemporânea que necessita de pessoas que aprendam e se adaptem depressa. O crescimento da Web tornou cada vez mais problemática a descoberta e a recuperação de conteúdos de aprendizagem. Por um lado, isso acontece porque a forma de a procurar através de palavras-chave desprovidas de significado e contexto faz com que esses conteúdos de aprendizagem permaneçam perdidos no fundo do mar de informação onde só os humanos os conseguem entender (para as máquinas essa informação passa despercebida, é difícil de localizar ou, simplesmente, não a conseguem interpretar), e por outro, porque poderão estar armazenados em bases de dados (como acontece no caso dos objectos de aprendizagem de um sistema de e-Learning), sendo complicado aceder a eles através dos motores de busca actuais. Esta parte da Web tem vindo a ser designada Web profunda (*Deep*

Web) ou Web invisível (*Invisible Web*), uma vez que as páginas não estão armazenadas em nenhum local, sendo geradas dinamicamente apenas quando o utilizador as solicita directamente em interacção com a base de dados por intermédio da aplicação ou página específica.

A Web foi criada com a visão de que seria um espaço onde a informação teria um significado bem definido, facilitando a cooperação e a comunicação entre as pessoas e os agentes de software (Berners-Lee et al., 2001). Passada mais de uma década sobre o seu lançamento, essa visão ainda não foi alcançada. A Web continua organizada na perspectiva dos humanos (linguagem natural e ênfase na apresentação através da linguagem HTML), o que torna a busca de informação uma tarefa árdua. Há que encontrar soluções de representação, organização, integração, intercâmbio e compreensão semântica da informação, tanto na óptica dos humanos, como na óptica das máquinas ou dos agentes de software. É neste contexto que surge a denominada Web Semântica (*Semantic Web*) (W3C, 2001) como um cenário desejável para a Web.

A Web Semântica é uma iniciativa do *World Wide Web Consortium* (W3C) cujo propósito principal é a integração, intercâmbio e compreensão semântica da informação, tanto na óptica dos humanos, como na óptica das máquinas, através de metadados (dados sobre dados), de ontologias (formas de representar explicitamente a semântica dos dados de um determinado domínio) e de agentes de software (componentes de software capazes de actuar de forma a resolver tarefas em nome do utilizador). A ideia da Web Semântica pode resumir-se à seguinte questão: como fazer com que os computadores entendam o conteúdo da Web? O primeiro passo será organizar e estruturar a informação e o segundo será adicionar semântica à informação da Web, de tal forma que os agentes de software possam compreendê-la.

Com a representação e estruturação dos objectos de aprendizagem em XML (*eXtensible Markup Language*), a representação do significado desses objectos de aprendizagem em metadados e a representação formal comummente aceite sobre o que significam esses objectos em ontologias, estarão criadas as condições para que os agentes de software possam raciocinar sobre objectos de aprendizagem com significado bem definido.

Por conseguinte, tanto os motores de busca como as arquitecturas de ensino a distância baseadas na Web têm que ser repensados, para tirarem partido destas potencialidades, para que possam responder de forma rápida, precisa e personalizada aos requisitos dos utilizadores.

No que diz respeito ao ensino a distância, os vários consórcios com responsabilidades nesta área têm vindo a juntar esforços no sentido de indexar, catalogar e normalizar os recursos de aprendizagem digitais, garantindo a interoperabilidade entre repositórios de informação, com vista a disponibilizar e combinar objectos de aprendizagem (*learning objects*) face aos requisitos de aprendizagem particulares de cada aluno e/ou requisitos de planificação de cada professor. Sabendo que muitas vezes a aprendizagem deriva do acesso a informações bem estruturadas, acreditamos que o uso de mapas de conceitos (representação visual que relaciona conceitos hierarquicamente e que pode ser traduzida num formato XML) para representar o conhecimento nos sistemas de e-Learning e nos repositórios de objectos de aprendizagem favoreceria não só a organização, localização, recuperação e reutilização de recursos educativos (objectos de aprendizagem), mas também o processo de aprendizagem (quer na planificação de cursos, quer na organização mental dos conceitos).

Quanto à iniciativa da Web Semântica, os vários consórcios e grupos de investigação em torno desta visão têm vindo a construir os alicerces de um inovador espaço de informação. A Web Semântica assume-se como uma rede de informação onde os conteúdos podem ser localizados e relacionados semanticamente através de agentes inteligentes. Não se trata de criar uma nova Web, mas sim de redefini-la no sentido de optimizar os seus recursos e modo de funcionamento em benefício do conhecimento humano. Tal como afirma Eric Miller, não se trata de uma revolução, mas sim de uma evolução (Miller, 2006) e a Educação é uma das áreas que mais podem beneficiar da evolução da Web para esta visão.

As duas alternativas de auto-aprendizagem referidas (aprendizagem através de conteúdos disponibilizados por sistemas de e-Learning e aprendizagem através de conteúdos Web recuperados por motores de busca) beneficiariam de uma visão segundo objectos de aprendizagem, devidamente descritos através de metadados, com significado semanticamente explícito e baseado em ontologias não só para permitirem procuras mais precisas desses objectos, mas também para favorecerem a sua combinação em planos de formação personalizados.

Neste sentido, propõe-se, por um lado, o recurso a metadados e ontologias para a pesquisa de objectos de aprendizagem dispersos e, por outro, o uso de metadados e mapas de conceitos para organizar e apoiar a recuperação de objectos de aprendizagem em cursos de e-Learning e a consequente geração de planos de formação personalizados.

Convictos de que as actividades de aprendizagem beneficiarão com esta evolução, mas cientes de que o processo de aprendizagem (como aprender) e os modelos inerentes ao projecto de aprendizagem e à estrutura e sequência dos conteúdos (como ensinar) não devem ser descurados, a presente dissertação explora as vantagens que a visão da Web Semântica pode provocar nos sistemas de e-Learning, repositórios de objectos de aprendizagem e páginas Web que disponibilizem conteúdos educativos, recomendando o recurso a metadados, a ontologias e a mapas de conceitos ou mapas de tópicos (abordagem análoga mais recente) para orientar os agentes de software na localização, recuperação e reutilização de objectos de aprendizagem.

1.1- Contexto e Motivação

Esta tese enquadra-se no âmbito dos Sistemas Distribuídos para a Web, da Tecnologia Educativa e da Gestão da Informação aplicada à Web, mais concretamente no **contexto** da Web Semântica e do e-Learning. Pretende realçar o papel que esta nova visão da Web pode originar nos Websites educativos em geral e nos sistemas de e-Learning em particular, tendo em vista a melhoria dos mecanismos de pesquisa de informação educativa na Internet (ou numa Extranet ou numa Intranet).

A primeira geração da Internet preocupou-se com a implementação da infra-estrutura tecnológica desta rede e de serviços, tais como a Web (o conteúdo era disponibilizado através de páginas estáticas); a segunda incidiu na construção de aplicações Web (o conteúdo passou também a ser disponibilizado através de páginas dinâmicas geradas a partir de base de dados); e, actualmente, a terceira avança rumo a uma Web mais inteligente – a Web Semântica.

O crescimento da Web proporciona a professores, alunos e cidadãos em geral um vasto conjunto de oportunidades de aprendizagem ou de geração de novo conhecimento que nem sempre é possível concretizar dadas as dificuldades de localização das fontes de informação úteis.

A ausência de um modelo de estruturação e organização da informação e de mecanismos que proporcionem efectivamente pesquisas de informação educativa na Web

providas de significado e dentro de um contexto ou domínio específicos são os principais factores deste problema e constituíram a **motivação** fundamental deste projecto de investigação. Com a integração das tecnologias para a Web Semântica nos actuais sistemas e aplicações educativas Web, pretendemos demonstrar que a localização e recuperação de informação passará a ser mais rápida, precisa e em conformidade com as necessidades do utilizador.

1.2- Finalidade, Objectivos, Inovação e Contributos

A **finalidade** deste projecto de doutoramento era explorar a aplicação das tecnologias para a Web Semântica no contexto educativo. Recorrendo a um estudo de campo, este projecto teve como propósito propor uma arquitectura para a recuperação de conteúdos educativos e, com base nela, implementar um protótipo experimental e determinar a influência de um conjunto de variáveis associadas (especificações para os metadados e para as ontologias e mecanismos para a inferência) num determinado contexto educativo – os sistemas de e-Learning e os repositórios de objectos de aprendizagem.

Concretamente, pretendíamos avaliar se a utilização de metadados para descrever objectos de aprendizagem e o recurso a ontologias para orientar a localização e recuperação de informação permitiriam pesquisas mais rápidas, precisas e em conformidade com as necessidades do utilizador.

Para tal, definiram-se como **objectivos principais** os seguintes:

a) Definir uma arquitectura para um sistema de pesquisa de conteúdos educativos baseada nas tecnologias para a Web Semântica, nas tecnologias para o e-Learning e nas tecnologias para a computação distribuída, que permitisse procurar objectos de aprendizagem em servidores de sistemas de e-Learning, de repositórios de objectos de aprendizagem ou noutros servidores Web de conteúdos educativos;

b) Desenvolver um protótipo do sistema proposto capaz de efectuar pesquisas mais precisas e profundas (baseadas em metadados e orientadas por ontologias), devolvendo informação o mais próxima possível das necessidades do utilizador;

c) Projectar um componente para relacionar a informação devolvida, gerando informação adicional que fosse de encontro aos requisitos das comunidades educativas em geral e dos professores e alunos em particular;

d) Identificar e implementar módulos, que favorecessem a indexação, classificação ou anotação de conteúdos educativos ou objectos de aprendizagem através de metadados e ontologias, destinados aos sistemas de e-Learning, repositórios de objectos de aprendizagem e páginas Web educativas;

e) Implementar um mecanismo de pesquisa baseado em metadados para um sistema de e-Learning, com vista a melhorar as estratégias actuais de pesquisa e de filtragem de conteúdos educativos;

f) Avaliar a receptividade dos professores e formadores na adição de metadados aos recursos educativos;

g) Projectar soluções alternativas para garantir a geração de metadados (geração automática de metadados, integração com outros módulos de um sistema de e-Learning, etc);

h) Comparar a utilização do mecanismo de pesquisa baseado em metadados com outros mecanismos de pesquisa disponíveis no sistema de e-Learning;

i) Usar linguagens, tecnologias e ferramentas recomendadas pelo W3C para o desenvolvimento de uma página educativa de um professor para a Web Semântica;

j) Validar a arquitectura do sistema proposto e o protótipo desenvolvido, bem como os módulos e mecanismo de pesquisa para um sistema de e-Learning.

Genericamente, os objectivos principais explicitam o que pretendíamos obter através de um conjunto de actividades de investigação laboratorial. Para a concretização desses objectivos, foram realizadas actividades que permitiam alcançá-los e que podem ser especificadas através dos seguintes **objectivos específicos**:

a) Identificar e analisar as recomendações, linguagens e tecnologias para a Web Semântica;

b) Identificar e analisar as infra-estruturas de suporte ao desenvolvimento de aplicações baseadas em agentes;

c) Identificar e analisar as diferentes arquitecturas de ensino a distância ou plataformas de e-Learning;

d) Instalar e configurar duas plataformas de e-Learning, uma para o contexto do Ensino Básico e Secundário, outra para o contexto do Ensino Superior para corroborar a aplicabilidade do sistema WSE proposto;

e) Instalar e configurar duas plataformas de gestão de conteúdos para confirmar a validade do sistema WSE proposto;

f) Estudar as normas e as especificações para descrever e classificar o conteúdo das páginas Web educativas;

g) Seleccionar as linguagens de programação, tecnologias de base de dados e outras tecnologias que melhor se adequem à solução proposta;

h) Comparar a arquitectura proposta ou os seus componentes com projectos existentes ou que entretanto viessem a surgir;

i) Apresentar linhas de trabalho futuro decorrentes das actividades de desenvolvimento realizadas.

A **inovação** neste projecto de doutoramento reside na apresentação de um modelo de integração e partilha de informação educativa, na definição de uma metodologia para a organização, estruturação e tratamento semântico da informação educativa, sugerindo o uso de disciplinas ou cursos de formação de elevada granularidade, cujos objectos de aprendizagem sejam automaticamente descritos com metadados, e na construção de um mecanismo de pesquisa educativo, baseado em metadados e ontologias, capaz de devolver informação semanticamente correcta, contornando algumas das principais limitações dos motores de busca e sistemas de e-Learning actuais.

As actividades de concepção e desenvolvimento orientadas pelos objectivos acima referidos resultaram, essencialmente, num sistema de recuperação de objectos de aprendizagem para a Web Semântica. Em termos práticos, os **contributos** deste projecto de investigação foram os seguintes:

- Uma arquitectura para um sistema de pesquisa de objectos de aprendizagem baseado nas tecnologias para a Web Semântica, nas tecnologias para o e-Learning e nas tecnologias para o desenvolvimento de aplicações baseadas em agentes móveis e inteligentes. As características principais desta arquitectura são:

- Ser um ambiente onde agentes estacionários coexistem com agentes móveis, tornando a gestão destes transparente para os responsáveis pela administração da infra-estrutura distribuída;
- Oferecer os serviços necessários para que os utilizadores solicitem a localização e recuperação de conteúdos educativos, sem que se apercebam da complexidade do processo de pesquisa de objectos de aprendizagem remotos;
- Disponibilizar as funcionalidades para melhorar o desempenho das pesquisas, evitando que os agentes móveis permaneçam no servidor para o qual se deslocam, seguindo o seu percurso enquanto a inferência é realizada nesse servidor remoto;
- Fornecer os serviços para submeter ou editar ontologias e mapas de tópicos com vista a anotar semanticamente os sistemas remotos e externos, orientar os agentes na pesquisa de conteúdos educativos dispersos e apoiar a navegação semântica, organizando os objectos de aprendizagem no plano de curso ou plano de aprendizagem seleccionado (mapa de tópicos).

- Um protótipo computacional do sistema proposto, provisoriamente denominado "Web Semântica Educativa (WSE)", baseado em agentes móveis e inteligentes, orientados por ontologias ou mapas de tópicos para pesquisar conteúdos de aprendizagem devidamente descritos por metadados e armazenados em servidores remotos de sistemas de e-Learning, repositórios de objectos de aprendizagem ou Websites educativos.
- Dois módulos para a geração (semi)automática de metadados destinados a uma plataforma de e-Learning, em conformidade com as principais especificações de metadados;
- Um bloco de pesquisa e um motor de inferência de objectos de aprendizagem baseados em metadados e ontologias desenvolvidos para uma plataforma de e-Learning;
- Uma página Web do professor desenvolvida totalmente com tecnologias para a Web Semântica, demonstrando as potencialidades das linguagens, tecnologias e ferramentas no âmbito dos metadados e ontologias e das linguagens de consulta.

Face à dimensão deste projecto de investigação em termos de tecnologias envolvidas, decidimos seguir uma abordagem generalista com vista à integração das camadas e tecnologias para a Web Semântica e sua aplicação no e-Learning, não menosprezando a especificidade de cada uma dessas camadas e tecnologias, sempre que necessário.

1.3- Estrutura do Documento

Com vista a clarificar a visão da Web Semântica e o impacto que a mesma poderá provocar no contexto educativo, demonstrando e validando os diferentes conceitos introduzidos no modelo e o desenvolvimento dos componentes do sistema "Web Semântica Educativa", este documento encontra-se dividido em seis capítulos ou secções principais.

O **primeiro capítulo** apresenta o contexto e a motivação da presente dissertação, a finalidade e os principais objectivos do projecto de investigação e a estrutura deste documento. Este capítulo pode ser complementado com o mapa de conceitos do anexo A.

O **segundo capítulo** diz respeito às tecnologias para o e-Learning. Após a caracterização da Sociedade da Informação em que vivemos, é efectuada uma retrospectiva sobre a crescente aposta em tecnologias educativas e sobre a evolução do ensino a distância. De seguida, são classificadas as tecnologias postas ao serviço do e-Learning, referidas as principais plataformas para e-Learning, justificada a importância das teorias pedagógicas no e-Learning e apresentada a filosofia dos objectos de aprendizagem como a forma adequada de desenvolver conteúdos de e-Learning e o *Learning Design* (projecto de aprendizagem) como a metodologia a adoptar. São também descritos os mapas mentais e mapas de conceitos, enquanto instrumento de organização, representação e sintetização do conhecimento.

O **terceiro capítulo** apresenta a evolução da rede Internet rumo a uma Web mais inteligente, bem como a evolução da arquitectura para a Web Semântica, analisando de seguida as principais tecnologias para as camadas estrutural e sintáctica: XML para a estruturação da informação e os esquemas de metadados e as metalinguagens para descrever e atribuir significado a essa informação.

O **quarto capítulo** incide sobre as principais tecnologias para as camadas semântica e lógica: as ontologias e mapas de tópicos, para percebermos universalmente o significado da informação, e as regras e mecanismos de inferência, para definirmos como é que os agentes inteligentes poderão raciocinar sobre essa informação de forma a gerar conhecimento e satisfazer os requisitos dos utilizadores.

Face à importância dos agentes de software no âmbito deste projecto de investigação, o **quinto capítulo** descreve os conceitos básicos sobre agentes e os requisitos principais dos sistemas multiagente e apresenta a arquitectura genérica para um sistema de agentes móveis,

para além de comparar sucintamente algumas infra-estruturas Java que suportam o desenvolvimento de aplicações baseadas em agentes móveis e inteligentes.

Fruto dos cenários que a visão da Web Semântica poderá originar no contexto educativo em geral e no e-Learning em particular, dos fundamentos apresentados, do estado da arte das tecnologias e especificações referidas e do processo de análise de requisitos, no **sexto capítulo** é proposto um modelo ou arquitectura para um sistema de pesquisa de conteúdos educativos para a Web Semântica, denominado Sistema WSE (Web Semântica Educativa). Genericamente, é apresentada a modelação do contexto e dos requisitos, a modelação do comportamento e da estrutura e, finalmente, a modelação da arquitectura.

A actividade de desenvolvimento do sistema proposto, descrita no **sétimo capítulo**, foi orientada por uma metodologia híbrida que permite seguir diversas metodologias consoante o subsistema que pretendemos modelar e especificar. Genericamente, são descritos os principais componentes do protótipo e as respectivas funcionalidades, bem como as características de implementação mais relevantes. Finalmente, são apresentados alguns dos principais resultados referentes à validação do protótipo.

Cientes de que o futuro da Web passa pela Web Semântica e de que a Educação pode ser um dos campos que mais proveito poderão tirar desta evolução, no **oitavo capítulo** é apresentada uma análise global do trabalho realizado e dos resultados decorrentes da avaliação e validação do sistema, para além de ser perspectivado o trabalho futuro.

Alguns dos assuntos aparecem muitas vezes tratados na bibliografia e Webliografia como temas separados. Em última instância, a redacção deste documento teve como intuito organizar essas ideias e pensamentos, noções e conceitos, tecnologias e plataformas com vista a integrá-las em prol daqueles que decidirem dar os primeiros passos no desenvolvimento de aplicações para a Web Semântica.

De referir também que apesar deste documento conter alguns endereços de páginas Web, não pretende ser um roteiro para o e-Learning ou para a Web Semântica. Descreve comportamentos humanos inerentes à utilização das tecnologias da informação e da comunicação, mas não é um ensaio sociológico. Não está escrito em linguagem para iniciados, no entanto, estes não terão dificuldades na leitura, reflexão e aprendizagem das ideias base. Não é um trabalho de antevisão ou de ficção científica, embora descreva alguns passos rumo a um futuro próximo. Finalmente, este documento não tem a aspiração de abordar todas as iniciativas existentes, contudo, esperamos ter apresentado os principais

projectos, metodologias, linguagens e ferramentas para Web com vista a clarificar as implicações que a visão da Web Semântica terá nos sistemas de e-Learning e noutros ambientes Web educativos.

2- Tecnologias para o e-Learning

As experiências e aplicações de e-Learning, ou de outros ambientes de Ensino a Distância (EAD) baseado nas tecnologias da Internet, nem sempre se traduzem numa verdadeira ruptura com o paradigma tradicional da educação (centrada na sala de aulas e no professor), privilegiando a cooperação e construção do saber. A rádio, a televisão e outras tecnologias (incluindo os sistemas tutoriais inteligentes e outros sistemas da Inteligência Artificial) prometeram revolucionar a forma de ensinar e aprender. A Internet, e mais concretamente as tecnologias para o e-Learning, trouxeram essa mesma promessa (Ferreira e Santiago, 1999). Estaremos perante uma revolução no processo de ensino/aprendizagem ou é apenas mais uma etapa na evolução da educação? E as mais recentes mudanças educativas, impulsionadas pelo "Processo de Bolonha", levarão os mais cépticos a acreditar nas potencialidades destas tecnologias? Estas são algumas das questões que servem de mote para o início deste capítulo.

A caracterização da Sociedade da Informação e do Conhecimento em que vivemos e a apresentação de uma retrospectiva sobre as medidas e projectos que apontam para a crescente aposta em tecnologias educativas representam o início deste segundo capítulo. Posteriormente, após referir sucintamente a evolução do ensino a distância (ou da educação a distância) e clarificar os conceitos de "Educação a Distância" e "e-Learning", são classificadas as tecnologias postas ao serviço do e-Learning e referidas as principais plataformas para o desenvolvimento de sistemas de e-Learning.

As tecnologias por si só não são suficientes para obter sistemas de e-Learning adequados às necessidades dos docentes e discentes, pelo que o modelo pedagógico inerente à organização e estruturação dos conteúdos não deve ser descurado. Referida a importância das teorias pedagógicas no e-Learning, é apresentada a filosofia dos objectos de aprendizagem como a forma adequada de desenvolver conteúdos de e-Learning e o *Learning Design* como uma das metodologia a adoptar no projecto da aprendizagem. Nesta perspectiva, são apresentadas sucintamente as normas e especificações que mais se destacam com vista a normalizar o processo de desenvolvimento, estruturação, partilha, reutilização e interoperabilidade de objectos de aprendizagem, com especial destaque para o modelo SCORM (*Sharable Content Object Resource Model*) da iniciativa *Advanced Distributed Learning* (ADL, 2004) e para as especificações IMS (*Instructional Management Systems*) do

consórcio IMS *Global Learning Consortium* (IMS, 2004), não esquecendo os esforços de outros consórcios, entidades ou iniciativas tais como: IMS/GLC (*Instructional Management Systems Global Learning Consortium*), IEEE/LTSC (*Institute of Electrical and Electronics Engineers Learning Technology Standards Committee*), AICC (*Aviation Industry CBT (Computer-Based Training) Committee*) e ARIADNE (*ARIADNE Foundation*), entre outras. Finalmente, abordamos os mapas mentais e mapas de conceitos, enquanto instrumento de organização, representação e sintetização do conhecimento (ou mesmo das estruturas cognitivas dos alunos) e sugerimos a sua integração no e-Learning através da sua tradução em mapas de tópicos.

2.1- Sociedade do Primado do Saber

As constantes mutações na Sociedade, algumas delas provocadas pela evolução tecnológica, obrigam as pessoas a alterarem mentalidades, posturas, atitudes e comportamentos. Reconhecer a importância da aprendizagem ao longo da vida, encarar a constante necessidade de actualização dos conhecimentos motivada pelos novos desenvolvimentos que provocam a mudança de funções, ou mesmo de profissão, são atitudes que os cidadãos devem adoptar.

Na Sociedade da Informação em que vivemos, o conhecimento é um bem de valor inestimável, pelo que é necessário promover a criação de mecanismos que contribuam para a sua consolidação e difusão. Aceder à informação disponível para gerar conhecimento constituirá uma necessidade básica para os cidadãos e compete às diversas entidades garantir que esse acesso se efectue de forma rápida e eficaz e numa base equitativa. A Sociedade da Informação é, portanto, uma sociedade do primado do saber.

A aprendizagem ao longo da vida tornou-se necessária em todos os níveis da sociedade e deve ser encarada como uma construção contínua da pessoa humana, dos seus saberes, aptidões e da sua capacidade de discernir e agir. O conceito de educação deve, por isso, evoluir, ultrapassando as fronteiras do espaço e do tempo ao longo do qual o aluno faz o seu percurso de escolarização, passando pelos diferentes níveis de ensino do sistema

educativo, para dar lugar a um processo de aprendizagem durante toda a vida, isto é, facultando a cada indivíduo a capacidade de saber conduzir o seu destino, num mundo onde a rapidez das mudanças se conjuga com o fenómeno da globalização (MSI, 1997).

O paradigma tradicional de educação, fruto da revolução industrial, não é visto como adequado nesta era digital. Uma educação baseada em livros e monólogos ou palestras dos professores não é suficientemente cativante e motivadora nos dias de hoje. Actualmente, o ensino pode ser muito mais interactivo, rico e diversificado em elementos e suportes de informação e assente em fluxos de comunicação suportados por várias tecnologias da informação e da comunicação.

2.1.1- A Sociedade da Informação e do Conhecimento em Portugal

Os avanços tecnológicos na infra-estrutura da Internet e na Indústria do Multimédia, bem como a massificação do uso da Internet, nomeadamente da Web, marcaram a década de 90 e contribuíram decisivamente para a investigação de novas estratégias de difusão de informação (e-mail, mensagens instantâneas, fóruns de discussão, Websites pessoais, Weblogs, portais temáticos, entre outros sistemas de informação para a Web), de novos modelos de comunicação (interacção pessoal e profissional a qualquer momento e em qualquer local, assumindo-se cada vez mais que um indivíduo é um nó da Rede que consome e publica informação) e de novas formas de ensino, formação e aprendizagem (aprendizagem através de conteúdos dispersos na Web e aprendizagem através de tutores inteligentes, sistemas hipermédia adaptativos e ambientes de aprendizagem a distância).

No contexto português, o Livro Verde para a Sociedade da Informação (1997) sugeria, no âmbito da medida 6.4, o fomento de programas de ensino a distância e a disponibilização dos seus *curricula* e das aulas em suporte multimédia na Internet ou noutra rede telemática (MSI, 1997). Posteriormente, o Plano de Acção para a Sociedade da Informação (2003) assumiu a aprendizagem electrónica como uma das principais prioridades a adoptar, acentuando a importância dos ambientes de aprendizagem aberta, interactiva e do ensino virtual, bem como o intercâmbio cultural e pedagógico a distância, para além de outras medidas para o contexto da formação e educação, nomeadamente as que fazem parte dos pilares 1 (Uma Sociedade da Informação para Todos), 2 (Novas Capacidades) e 7 (Conteúdos

Atractivos), ou para a administração pública com programas de formação contínua através de e-Learning (pilar 3 – Qualidade e Eficiência dos Serviços Públicos) (UMIC, 2003).

As Tecnologias de Informação e Comunicação (TICs) têm vindo a impulsionar o aparecimento de novas soluções para a formação e educação, quer em ambientes presenciais, quer a distância. Algumas das principais iniciativas no contexto educativo derivam de projectos de Universidades, Editoras, Centros de Competência e de programas como o "Internet nas Escolas", "Nónio-Século XXI - Projectos de Informação sobre Educação (Internet)", "Ciência Viva", "Cidades Digitais", "Email para todos" e "TIC para todos", entre outros. Genericamente, a infra-estrutura tecnológica implementada baseia-se na Rede de Ciência, Tecnologia e Sociedade (RCTS) sob a orientação da Fundação para a Computação Científica Nacional (FCCN) e a componente educativa e de conteúdos ficou a cargo da Unidade de Apoio à Rede Telemática Educativa (uARTE) aquando da criação do Ministério da Ciência e da Tecnologia (MCT).

Neste últimos anos, foram levados a cabo outros programas e projectos tais como "Programa Internet@EB1", "Programa 1000 salas TIC", "Programa e-U - *Campus Virtuais*", "Escolas Virtuais", "Iniciativa Nacional para a Banda Larga", "Um Computador por Professor", "Portal da Formação", "Projecto Estúdios" (video-conferência/video-difusão e sessões de ensino a distância), entre outros programas e projectos promovidos pelo Estado Português que visam estimular o acesso à Internet, o uso de redes electrónicas para efeitos pedagógicos e a disponibilização de conteúdos de interesse educativo. O POSI (Programa Operacional para a Sociedade da Informação), actualmente designado POS_Conhecimento (Programa Operacional da Sociedade do Conhecimento), é o mais recente programa de apoio financeiro (POSC, 2006).

Figura 1 – Programa Operacional Sociedade do Conhecimento

Paralelamente, têm vindo também a decorrer acções de formação através do Programa FOCO (Formação Contínua de Professores) ou de programas similares financiados pelo PRODEP com vista a promover a utilização das tecnologias educativas pelos cidadãos em geral, dos quais se destaca o Sistema Nacional de Formação e Certificação em TICs que permite a qualquer cidadão obter o DCB (Diploma de Competências Básicas em TICs).

O aparecimento de Sistemas Hipermédia veio consolidar as potencialidades da utilização das TICs na Educação. Enquanto que grande parte dos produtos de Ensino Assistido por Computador e outras soluções multimédia que se lhe seguiram apresentavam um modelo relativamente rígido e linear, os produtos hipermédia (locais ou em rede) trouxeram ao utilizador maior liberdade de navegação na informação e mais interactividade.

Genericamente, a Internet tem sido usada no contexto educativo de três formas, que muitas das vezes se sobrepõem ou complementam: como fonte de informação (modo colector), como meio de publicação (modo produtor) e como meio de interacção (modo comunicador). Actualmente, recorre-se também cada vez mais ao modo colaborativo ou cooperativo, assumindo-se o utilizador como mais um nó da rede, consumindo e partilhando ideias e informação. Que futuro para a Internet na Escola? A existirem incógnitas nesta questão, não será tanto do lado da Internet, mas mais do lado da Escola. Que as tecnologias da Internet têm e terão um forte impacto na educação, disso não há dúvida. A questão será qual o espectro desse impacto (Freitas, 1999; Gonçalves, 2006).

Pouco a pouco, tornou-se imprescindível adoptar novas formas de ensino/aprendizagem que capacitassem alunos e professores para alcançar ou mesmo superar as suas metas, de forma a enriquecer as situações de ensino/aprendizagem e todo o ambiente educativo com o recurso às mais recentes tecnologias. A Internet assume-se pois como a infra-estrutura ideal para suportar a mudança que se exige tanto nos processos de negócio como nos educativos (Gonçalves, 2002).

Ninguém duvida de que há fortes constrangimentos sobre o que as escolas podem fazer. Estas raramente são livres de experimentar novas abordagens e tecnologias educativas, tais como o e-Learning. Mas, ao longo destes últimos anos, as barreiras tecnológicas, culturais e de formação têm vindo a ser reduzidas consideravelmente. Para além das instituições do Ensino Superior, todas as escolas públicas (Ensinos Básico e Secundário) estão finalmente ligadas à Internet de banda larga, desde Janeiro de 2006. O Plano Tecnológico, enquanto agenda de mobilização para toda a sociedade portuguesa, indicando uma estratégia de desenvolvimento e

competitividade (UMIC, 2006a), é mais uma medida política que aposta no derrube dessas barreiras. Um dos objectivos estratégicos do Eixo 1 - Conhecimento (Qualificar os Portugueses para a Sociedade do Conhecimento) define como objectivo "Fomentar a aprendizagem ao longo da vida" e, no âmbito da Educação e Formação, destaca-se o objectivo de "Estimular a Formação Profissional, a Certificação e a Investigação e Desenvolvimento em Tecnologias". Para ajudar a qualificar os portugueses para a sociedade do conhecimento e vencer o atraso científico e tecnológico, têm vindo a ser celebrados vários protocolos entre o Estado Português e instituições reconhecidas internacionalmente (por exemplo, recentemente, a Microsoft, alinhada com a agenda do Plano Tecnológico, decidiu lançar no contexto da Educação uma ampla iniciativa denominada *ITeach*).

No ano lectivo de 2005/06, no âmbito da CRIE – Equipa de Missão Computadores, Redes e Internet na Escola, o programa Internet@EB1 deu lugar ao programa CBTIC@EB1 abarcando actividades mais amplas em TIC e a integração das actividades nas várias áreas curriculares do 1º Ciclo. Também em 2005/06, a CRIE promoveu a formação de formadores de professores em TIC, envolvendo 573 participantes, 228 entidades formadoras, 34 acções de formação, 18 Centros de Competência em TIC com plataforma Moodle. Promoveu ainda a formação de 15.109 professores em 175 projectos e a disseminação da utilização de plataformas de e-Learning Moodle envolvendo 2.094 professores (UMIC, 2006b). No âmbito dos projectos Campus Virtual (e-U), têm sido desenvolvidos, nas universidades e nos politécnicos, um vasto leque de conteúdos educativos. Também, no âmbito dos Ensinos Básico e Secundário, decorreu um concurso de produção de conteúdos educativos em formato digital que levou à aprovação de 256 projectos que se encontram em curso (UMIC, 2006a).

Figura 2 – Agência para a Sociedade do Conhecimento (UMIC)

Finalmente, destaca-se também o facto do Plano Tecnológico Português manifestar a intenção de aumentar a percentagem da participação da população em aprendizagem ao longo da vida de 4.8% para 12.5% de 2004 até 2010 (OCDE, 2006).

Todas estas iniciativas, medidas e projectos, promovidas pela UMIC - Agência para a Sociedade do Conhecimento do Ministério da Ciência, Tecnologia e Ensino Superior, ao fomentarem a aprendizagem ao longo da vida e ao criarem as condições tecnológicas e de

formação que permitem aceder aos ambientes de aprendizagem, estão, por um lado, a preparar os cidadãos para a utilização das TICs, fornecendo-lhes as bases para a utilização das TICs em geral e dos sistemas de e-Learning em particular, e, por outro, a alimentar a necessidade de sistemas de e-Learning ou outros ambientes virtuais de aprendizagem (*Virtual Learning Environments* ou simplesmente VLE).

2.1.2- O Processo de Bolonha e o e-Learning

Segundo o relatório da OCDE "*Reviews of National Policies for Education - Tertiary Education in Portugal*", a abordagem educativa portuguesa predominante é o ensino baseado na exposição e a aprendizagem como recepção passiva do conhecimento, descurando formas de interagir, questionar e experimentar. Estas são as principais razões para as taxas baixas de participação e de sucesso educacionais.

O mesmo relatório recomenda que deve ser dada uma especial atenção à disponibilização de oportunidades de aprendizagem formais e informais. Isto requererá o desenvolvimento de novas formas de suporte da aprendizagem e novas modalidades de ensino/aprendizagem e formação em lugares, em momentos e nos formatos mais convenientes e apropriados aos formandos (OCDE, 2006).

A alteração da Lei de Bases do Sistema Educativo introduzida pelo Processo de Bolonha abre várias oportunidades para o desenvolvimento e utilização do e-Learning em Portugal.

Segundo o Decreto-Lei nº 42/2005 de 22 de Fevereiro (Diário da República – I Série – A n.º 37, pág.1494), o Processo de Bolonha representa um vector determinante para o cumprimento da Estratégia de Lisboa para 2010, aprovada em Março de 2000 pelos presidentes e chefes de governo dos países da União Europeia, que visa tornar a Europa, até 2010, o espaço económico mais dinâmico e competitivo do mundo, baseado no conhecimento e capaz de garantir um crescimento económico sustentável, com mais e melhores empregos e com maior coesão social.

A importante mudança em curso nos paradigmas da formação e da educação reconhece o conhecimento como um bem universal, a premência de tornar o Ensino Superior mais atractivo e mais próximo dos interesses da sociedade e a necessidade de adaptar o

processo de aprendizagem aos conceitos e perspectivas da sociedade moderna e aos meios tecnológicos disponíveis.

A importância da educação ou ensino a distância é, finalmente, reconhecida formal e legalmente através do paralelismo equitativo estabelecido em relação ao ensino presencial no artigo 8.º (Ensino a distância) do Decreto-Lei nº 42/2005:

1) Nos cursos ministrados total ou parcialmente em regime de ensino a distância aplica-se o sistema de créditos curriculares.
2) Às unidades curriculares oferecidas, em alternativa, em regime presencial e a distância é atribuído o mesmo número de créditos.

Neste contexto, as práticas de e-Learning configuram-se como uma das possibilidades, com particular destaque para o contributo que podem dar ao nível de três das principais linhas de acção do Processo de Bolonha (Gomes, 2006):

- Promoção da mobilidade;
- Promoção da dimensão europeia do ensino superior;
- Promoção da aprendizagem ao longo da vida.

Esta última linha de acção já tinha sido também sublinhada nos descritores de Dublin para o 1º e 2º ciclo desenvolvidos pelo grupo JQIG (*Joint Quality Iniciative Group*). Ou seja, o descritor "competências de auto-aprendizagem (competências que permitam aos estudantes uma aprendizagem ao longo da vida, com elevado grau de autonomia)" é uma das competências a desenvolver pelos alunos para que lhes seja atribuído o grau referente ao 1.º ou ao 2.º ciclo. Refira-se que estes descritores surgiram no âmbito do "Projecto *Tuning* - sintonizar as estruturas educativas da Europa" e tinham por objectivo facilitar a comparação de ciclos de formação à escala europeia (DGES, 2005).

Naturalmente, a Lei n.º 49/2005 de 30 de Agosto, referente à segunda alteração à Lei de Bases do Sistema Educativo e primeira alteração à Lei de Bases do Financiamento do Ensino Superior, passou a consagrar o ensino a distância como uma das modalidades especiais de educação escolar (artigo 19º) e especifica no seu artigo 24º (Ensino a distância) que:

1) O ensino a distância, mediante o recurso aos multimédia e às novas tecnologias da informação, constitui não só uma forma complementar do ensino regular mas pode constituir também uma modalidade alternativa da educação escolar.

2) O ensino a distância terá particular incidência na educação recorrente e na formação contínua de professores.

Por conseguinte, a Internet em geral, e o e-Learning em particular, assumem-se como o instrumento ideal para suportar a mudança que se exige nos processos educativos não só no âmbito na educação recorrente e na formação contínua, mas também em praticamente todos os níveis de ensino. Para além de constituirem uma alternativa à mobilidade de professores e alunos entre instituições educativas através da disponibilização de um ambiente virtual de mobilidade, evitam que razões de natureza financeira, profissional, familiar, geográfica ou qualquer outra condicionem ou impossibilitem processos de ensino/aprendizagem.

A formação ou ensino a distância surge assim como uma alternativa que visa eliminar algumas barreiras, nomeadamente as da distância e do tempo, permitindo a cada indivíduo aprender de acordo com as suas disponibilidades e ritmos de aprendizagem. Mas, convém não esquecer que as experiências de ensino a distância levadas a cabo ao longo destes últimos anos tiveram como principal preocupação as questões geográficas e temporais, às quais o ensino presencial se mostrava incapaz de dar resposta. Actualmente, é fundamental conceber soluções de formação ou ensino a distância e, em particular, de e-Learning, que flexibilizem o acesso aos recursos de aprendizagem (qualquer sítio, qualquer hora); implementem estratégias pedagógicas adequadas a uma melhor aprendizagem; disponibilizem experiências com casos reais; suportem relações de cooperação; ajudem a aprendizagem, apoiando-se nas tecnologias de informação e comunicação mais recentes e sobretudo mais eficazes (Lima e Capitão, 2003).

Tal como refere Bruner (2000), não nos encontramos no fim da estrada no tocante à Educação. Na verdade, há boas razões para acreditar que estaremos justamente a arrancar para uma nova jornada. Actualmente, não poderíamos estar mais de acordo.

2.2- Ensino a Distância e e-Learning

Electronic Learning (e-Learning), *Online Learning* (OL), *Distance Learning* (DL), *Internet-based Learning* (IBL), *Web-based Learning* (WBL), *Virtual Learning Environments*

(VLE), *Content Management System* (CMS), *Learning Management System* (LMS), *Learning Content Management System* (LCMS), entre outros, são termos e siglas cada vez mais comuns no âmbito do EAD. Mais importante do que expor diferentes designações ou definições, importa identificar diferenças significativas muitas vezes dissimuladas na terminologia usada.

Embora não seja fácil nem consensual, primeiramente convém clarificar os principais conceitos deste campo de estudo, nomeadamente os conceitos de "Educação ou Ensino a Distância" e de "e-Learning", com vista a evitar eventuais confusões.

Para compreender a existência de um amplo e diversificado leque de definições e conceptualizações em torno do conceito de e-Learning, M.ª João Gomes (2005) recorreu às ideias de Scheffler sobre as características das definições no domínio da Educação. Embora alguns autores pretendam assumir as suas propostas como definições científicas, já que se baseiam em conhecimento específico, o qual é usado para construir uma rede teórica adequada a todos os factos disponíveis e que se pretendem incluídos na definição em causa, no caso concreto do e-Learning, supomos estar numa fase de desenvolvimento em que predominam as definições gerais: assume-se que determinado termo deve ser entendido de determinada forma no contexto de certas discussões ou debates.

O conceito de Educação ou Ensino a Distância (EAD) não é um conceito recente. Já na segunda metade do século XIX, o pedagogo português João de Deus descrevia algumas características que hoje em dia reconhecemos à Internet, quando se referia àquilo que ficou entre nós conhecido como as Escolas Móveis de João de Deus. O conceito de facilitar a comunicação e de levar a informação até aos alunos, em vez de os obrigar a deslocarem-se à sala de aula, assume-se cada vez mais como a alternativa mais viável face às limitações do paradigma de educação predominante (Ferreira e Santiago, 1999).

Tipicamente, falamos de EAD quando estamos perante um processo de ensino/aprendizagem em que o professor e o aluno estão geograficamente distantes e a interacção entre ambos é estabelecida, preferencialmente, via meios electrónicos. A última geração do EAD caracteriza-se por comunidades virtuais e sistemas de e-Learning mais fáceis de usar, mais interactivos, mais acessíveis e que permitem maior flexibilidade temporal e espacial do que os sistemas das gerações anteriores:

- 1.ª Geração EAD: Ensino por correspondência (texto impresso para a disponibilização de conteúdos e o correio postal ou carta para a comunicação entre professor e aluno);

- 2.ª Geração EAD: Tele-educação através da rádio, televisão e cassetes de áudio e vídeo (Universidade Aberta);
- 3.ª Geração EAD: Serviços Telemáticos baseados em comunicações assíncronas tais como e-mail e fóruns de discussão, para além de começarem a aparecer os primeiros conteúdos publicados através de páginas Web substituindo os CD-ROM e outros suportes digitais) (Santos, 2000; Lima e Capitão, 2003; Carvalho e Cardoso, 2004).

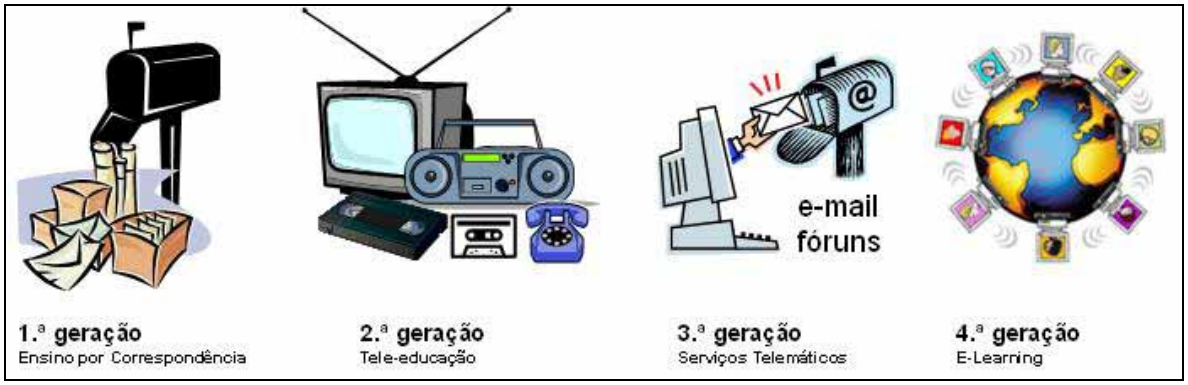


Figura 3 – Gerações de EAD

O termo e-Learning significa "electronic learning" ou aprendizagem suportada por meios electrónicos. Alguns autores aplicam o termo a qualquer utilização das TICs em contextos educativos. Outros aplicam-no apenas à forma de ensino e formação através de plataformas que utilizam a Internet como meio de comunicação entre formandos/alunos e o formador/professor.

O e-Learning pode ser um tipo ou modalidade de EAD, desde que baseado nas tecnologias da Internet, onde a aprendizagem ocorre remotamente. Mas EAD não é necessariamente e-Learning, uma vez que o e-Learning tem uma abrangência um pouco mais restrita que o EAD porque não inclui os cursos por correspondência, de televisão, em cassetes de áudio ou vídeo, entre outros cenários de EAD mais convencionais. Se bem que estes cenários de EAD convencionais ainda existam, eles fogem ao âmbito desta investigação.

A rádio, a televisão e outras tecnologias, incluindo os sistemas tutoriais inteligentes e outros sistemas da Inteligência Artificial, prometeram revolucionar a forma de ensinar e aprender. A Internet, nomeadamente através do e-Learning, trouxe essa mesma promessa. Será finalmente possível uma verdadeira ruptura relativamente ao paradigma tradicional da educação centrada na sala de aulas e no professor, privilegiando a construção do saber? Estamos perante uma revolução ou é apenas mais uma etapa na evolução da educação?

Algumas definições de e-Learning valorizam os aspectos tecnológicos, outras parecem preocupar-se mais com os aspectos da aprendizagem e outras ainda sugerem extensões de forma a optimizar as potencialidades das tecnologias e metodologias de aprendizagem a distância. Por exemplo, o crescente recurso a redes sem fio (*Wireless*) e a tecnologias móveis (Telemóveis, PDAs, computadores portáteis e terminais similares) tem vindo a impulsionar uma nova modalidade de ensino a distância – *Mobile Learning* ou simplesmente m-Learning. Para além de permitir o acesso aos conteúdos e outras funcionalidades do sistema, esta modalidade poderá facultar a possibilidade de descarregar um pacote software para utilização *off-line*, permitindo posterior sincronização para actualizar o registo do formando ou formador.

O e-Learning, sendo uma modalidade de EAD, proporciona uma aprendizagem personalizada, em conformidade com a necessidade, a disponibilidade e o ritmo do indivíduo, independentemente da plataforma usada para aceder à Internet. Esta metodologia de ensino derruba barreiras temporais e espaciais (*anytime, anywhere*), pelo que minimiza problemas decorrentes da impossibilidade de frequentar aulas devido a distância ou tempo e doença ou deficiência, incluindo os alunos com necessidades educativas especiais. Poder aprender sem limitações de horário e espaço físico é, sem dúvida, a situação ideal para todos os que têm uma actividade profissional exigente ou que estão geograficamente distantes dos centros de ensino e formação. Em suma, o e-Learning estimula a auto-aprendizagem, pelo que se insere no conceito de educação ao longo da vida.

Mas, afinal, onde reside a vantagem desta modalidade de EAD quando comparada com os restantes modelos de EAD? Genericamente, a mais valia está nas características de comunicação, interacção e interactividade disponibilizadas pelas tecnologias da Internet. Quer o acesso aos conteúdos, quer aos intervenientes do processo educativo, bem como o processo de publicação, distribuição e actualização dos recursos educativos, são mais fáceis, rápidos e frequentes.

Porém, para clarificar a definição de e-Learning, existem alguns cenários de utilização das TIC na Educação que devem ser excluídos da concepção de e-Learning pretendida no contexto desta investigação (Gomes, 2005):

- Recurso a DVDs, CD-ROMs, bibliotecas e museus *on-line* ou outros suportes digitais como meio de auto-estudo por si só (embora sejam consideradas excelentes alternativas de acesso a recursos de aprendizagem para construir conhecimento);

- Utilização da Internet para acesso a Websites lúdicos ou educativos, mesmo que enquadrada no âmbito de disciplinas ou cursos de formação e orientada por roteiros de exploração, Webquests ou outras estratégias similares (não menosprezando as potencialidades destas estratégias, nem invalidando a sua utilização no âmbito do e-Learning);
- Utilização da Web para publicar ou aceder a programas das disciplinas, sumários das aulas, apontadores relevantes para temas da disciplina, apresentações electrónicas ou outros documentos de apoio usados nas sessões presenciais (não menosprezando o seu valor no contexto dos *Campus Virtuais*, nem invalidando a sua utilização no âmbito do e-Learning, mas que por si só limita o conceito de e-Learning centrado na aprendizagem);
- Interacção *on-line* síncrona ou assíncrona para esclarecer dúvidas e fomentar debates, através de correio electrónico, de fóruns de discussão ou de programas de mensagens instantâneas (por exemplo *MSN Messenger da Microsoft*).

Em suma, excluímos as abordagens que defendem que qualquer utilização das TICs para apoiar a aprendizagem pode ser considerada e-Learning. Apostamos numa definição integradora de todos estes cenários e centrada na aprendizagem mediada através de tecnologias que se complementem, formando um ambiente de e-Learning como um todo.

Actualmente, o e-Learning assume-se como um método de ensino que rivaliza, para além de complementar ou constituir uma alternativa válida, formal e legalmente aceite, tal como referido anteriormente, com métodos mais convencionais de EAD ou mesmo com o ensino presencial, em que professor e alunos se encontram no mesmo espaço físico e ao mesmo tempo e a responsabilidade e gestão da aula é do professor e a ele cabe perceber e alterar a motivação e interesse dos alunos. Não obstante, o ensino presencial continuará a ser apropriado e necessário para leccionar determinados conteúdos e para algumas camadas da população estudantil (Chute et al., 1999).

Independentemente das diversas designações e outras tantas definições de e-Learning, é requisito fundamental que o ambiente de aprendizagem de um sistema de e-Learning esteja integrado numa rede (Rosenberg, 2001).

Por conseguinte, e-Learning representa um ambiente de aprendizagem, no qual a distribuição de conteúdos multimédia, a interacção social e o apoio na aprendizagem são suportados pela Internet ou por uma Intranet ou Extranet.

A utilização de e-Learning pode levantar algumas dúvidas no que diz respeito à sua eficiência e eficácia. No ensino dito tradicional, a relação pedagógica era necessariamente presencial, entre o professor que ensina e o aluno que aprende. No momento em que se amplia o conceito de ensino, o âmago da questão deixa de ser o modelo formal e passa a ser sobre "o que é educar?" e "o que é educação?". Impõe-se uma nova cultura da aprendizagem considerando a conjunção de diversas mudanças sociais, tecnológicas e culturais. A exigência de mudança não é apenas resultado da Sociedade da Informação, mas sobretudo da Sociedade da Aprendizagem (Amaral, 2005).

Alguns formadores, professores ou educadores chegam a questionar se esta modalidade de ensino permite atingir os mesmos resultados que o ensino presencial (ou mesmo superá-los), argumentando, por exemplo, que o seu papel é menosprezado. Os alunos ou formandos do ensino ou formação a distância atingirão níveis de aprendizagem aceitáveis? Quais os níveis de motivação, desempenho, produtividade e aproveitamento dos alunos envolvidos num processo de ensino a distância quando comparado com um processo de ensino presencial? Os conteúdos dos cursos presenciais poderão migrar facilmente para cursos de e-Learning? Se já é difícil convencer os professores das vantagens de ter as suas aulas num formato digital, como convencê-los dos benefícios do desenvolvimento de e-cursos (cursos de e-Learning) em conformidade com as normas actuais? Diversas comparações no âmbito de trabalhos de investigação em torno destas questões (Moore e Thompson, 1997; Driscoll, 1998; Santos, 2000; Khan, 2001; Kruse, 2004; Romiszowski, 2003) têm demonstrado que o ensino a distância é tão ou mais eficiente e eficaz que o ensino dito tradicional (*face-to-face*), desde que a metodologia e a tecnologia sejam usadas de forma apropriada.

Em relação a estas questões, podemos referir dois aspectos importantes: a interacção com os conteúdos e a interacção com os intervenientes do processo educativo (professor e alunos).

Quanto à interacção com os conteúdos, convém não esquecer que a transposição simples e directa dos conteúdos do ensino presencial para um formato de ensino a distância não é a solução adequada para criar cursos de e-Learning com qualidade e susceptíveis de proporcionar resultados benéficos.

Já há algum tempo que se reconhece que a aprendizagem deve ser muito mais do que a simples recepção de informação e aquisição de conhecimento. Tal como refere Roger Schank (1993), um bom software educativo é activo, ou seja, deve permitir que os alunos façam

qualquer coisa e que não se limitem apenas a ver qualquer coisa (atitude passiva). A criação dos conteúdos deve pois basear-se em orientações pedagógicas e pressupostos específicos e inerentes aos fins particulares a que se destinam (Lima e Capitão, 2003).

No que diz respeito à interacção com o professor, convém esclarecer que o e-Learning não pretende substituir o professor. Necessitaremos sempre de orientação, alguém a quem perguntar ou de alguém com quem comentar ou discutir. Essa função continua a ser do professor. Não obstante, o seu papel, bem como o do aluno e a relação entre ambos, terão de mudar. O professor transforma-se num facilitador de aprendizagem, uma vez que a aprendizagem é centrada no aluno, em detrimento da centralização no professor. Por seu lado, o aluno deverá assumir uma atitude mais pró-activa, uma vez que os conteúdos (recursos de aprendizagem) se encontram sempre disponíveis e a ele cabem o controlo, a organização, a condução e a decisão sobre o método de estudo (Santos, 2000; Lima e Capitão, 2003). Ao aluno são exigidas responsabilidade e autonomia no processo de aprendizagem, colaborando com os restantes intervenientes do processo educativo na construção do conhecimento.

Por conseguinte, o e-Learning exige alguma maturidade, autodisciplina e motivação (factores que só se adquirem com a idade), pelo que se aceita facilmente que esta forma de ensino atinja melhores resultados com adultos. Mas, não basta garantir a transmissão de conteúdos, é também necessário assegurar a interacção aluno-professor e aluno-aluno, de preferência em tempo real. Assim, as comunidades virtuais são de grande importância para o sucesso do processo de aprendizagem ao favorecerem a interacção e colaboração entre os intervenientes em cada curso de e-Learning (Pallof e Pratt, 2002).

Embora as tecnologias síncronas ou assíncronos que promovem a interacção e colaboração entre os intervenientes no processo educativo sejam cruciais no sucesso do processo de ensino/aprendizagem, elas não serão detalhadas neste documento uma vez que não são relevantes no contexto desta investigação.

Estas hipotéticas limitações inerentes à transmissão de conteúdos e à relação pedagógica justificam que o e-Learning tenda a ser complementado com sessões presenciais (esta combinação das duas formas de ensino designa-se por *Blended Learning* ou simplesmente b-Learning) (Hofmann, 2002), maximizando os proveitos e minimizando os prejuízos que ambos possam acarretar. Não obstante, há que ter consciência de que tanto podemos cair no erro de ter uma educação a distância tradicional como podemos ser muito inovadores num modelo dito tradicional.

O e-Learning representa uma mudança paradigmática não apenas para os alunos, mas também para os professores, administradores, técnicos e pessoal de apoio. Enfim, para a instituição como um todo (Khan, 2005).

Embora os preconceitos em relação à educação a distância não tenham sido definitivamente postos de parte, passada mais de uma década após a criação dos primeiros *Interactive Computer Based Training* e da publicação na *Training Magazine* de um dos primeiros artigos sobre "*Intranet based training*" que impulsionou, segundo alguns autores, o e-Learning, podemos retirar algumas lições e recomendações dos vários estudos e artigos, entretanto publicados, que analisaram o sucesso ou fracasso de diversos projectos de e-Learning (Zenger e Uehlein, 2001; Clark e Mayer, 2002; Greenagel, 2002; Phillips e Philips, 2002; Kruse, 2004; Khan, 2005).

Normalmente, as falhas no e-Learning podem ocorrer em três níveis interligados: ao nível do produto (*design* inadequado do curso e infra-estrutura tecnológica inadequada); ao nível do aprendiz (alunos inadequadamente preparados, falta de motivação e falta de tempo); e ao nível organizacional (falta de suporte de gestão, falta de um adequado sistema de recompensas) (Romiszowski, 2003).

Não sendo objectivo deste capítulo a apresentação detalhada das vantagens e desvantagens ou do sucesso e fracasso do EAD, e em particular do e-Learning, convém referir que é desejável que as tecnologias para o e-Learning usadas na mediatização dos conteúdos e da comunicação entre professores e alunos, e destes entre si, estimulem a formação ao longo da vida, não descurando os aspectos pedagógicos, de modo a que os resultados obtidos sejam iguais ou melhores, quando comparados com os da formação dita tradicional.

Segundo Ferreira da Silva (Santos, 2000), o e-Learning procura responder aos requisitos de um processo de aprendizagem através da integração de duas vertentes principais:

1) Aplicação e utilização das TICs (Aplicações Web, correio electrónico, fóruns de discussão, chats, vídeo-conferência);
2) Exploração e adaptação dos recursos educativos (Conteúdos Web ou materiais de estudo).

Em suma, o e-Learning preocupa-se, por um lado, com a comunicação entre o professor e o aluno e, por outro, com os conteúdos do curso. Os avanços tecnológicos e correspondente diminuição da importância das restrições inerentes à largura de banda e à integração dos diferentes elementos de multimédia em aplicações hipermédia, têm conduzido ao aparecimento de ferramentas de comunicação e de criação de conteúdos cada vez mais

atractivas e, consequentemente, a conteúdos mais ricos e estruturados e a mecanismos de interacção mais motivadores e cooperativos que impulsionam a aprendizagem.

Atendendo à teoria de Moore (1991), um processo de EAD é influenciado pelas seguintes variáveis: a Estrutura, o Diálogo e a Distância Transaccional. Assim, é fundamental preparar previamente todos os materiais e conteúdos para o curso (estrutura), os canais de comunicação a utilizar durante o processo de aprendizagem (diálogo) e as transacções educativas entre os alunos e os professores, os alunos e os conteúdos ou os alunos e as aplicações ou componentes multimédia (distância transaccional), não esquecendo a importância do grau de auto-controlo do próprio aluno durante o processo de aprendizagem (Santos, 2000).

Neste sentido, podem ser identificados cinco componentes comuns a sistemas de EAD: materiais e conteúdos; professores ou formadores; sistema de interacção (modelo de comunicação unidireccional ou bidireccional); tecnologias de distribuição de conteúdos e de comunicação (Web, e-mail, vídeo/áudio) e sistemas de avaliação (do aluno, da formação e do sistema) (Carmo, 1999).

2.3- Plataformas para os Sistemas de e-Learning

Genericamente, o e-Learning preocupa-se, por um lado, com a comunicação entre o professor e o aluno (interacção social e intra-pessoal) e, por outro, com os conteúdos do curso (interacção com os recursos de aprendizagem).

Embora não existam tecnologias ideais para a implementação de sistemas de e-Learning, uma vez que a sua aplicação é mais ou menos apropriada consoante a situação de aprendizagem, podemos agrupá-las em (McGreal et al., 2000):

1) Tecnologias de comunicação que promovem a interacção entre os intervenientes do processo de aprendizagem, bem como o trabalho colaborativo:
    a) Assíncronas: correio electrónico, listas ou fóruns de discussão, Wikis e Blogs (estas duas últimas tecnologias fornecem um maior poder ao utilizador para

publicar conteúdos e são conotadas com o que actualmente se designa de e-Learning 2.0);

b) Síncronas: *Internet Relay Chat* (IRC ou simplesmente Chat), audioconferência, videoconferência, *WhiteBoard* e mensagens instantâneas.

2) Tecnologias Web que permitem a distribuição dos conteúdos e a implementação de ambientes colaborativos (turmas virtuais) através de um ambiente integrado de aprendizagem distribuída (*Integrated Distributed Learning Environment*). Para o desenvolvimento de recursos, destacam-se as ferramentas de criação de páginas Web e as ferramentas de criação dos elementos multimédia que integrarão aplicações multimédia ou hipermédia. A actualização da informação num sistema de e-Learning é cada vez mais fácil e intuitiva, ficando desde logo disponível para todos os alunos onde quer que eles estejam, uma vez que as ferramentas que o integram permitem a criação e distribuição de cursos.

3) Ferramentas de Autor para a criação de aplicações multimédia *on-line* que permitem disponibilizar conteúdos Web interactivos, favorecendo a auto-aprendizagem. As aplicações multimédia deverão resultar da interacção entre os especialistas de Conteúdos, especialistas em Pedagogia e Didáctica e especialistas em Comunicação Educacional Multimédia (Correia, 1999).

Podemos ainda acrescentar as ferramentas para a gestão administrativa e académica (processos de inscrição e matrícula, registo de avaliações, estatísticas, expediente, etc.).

A combinação destes grupos de tecnologias para suportar o ensino a distância baseado na Internet permite obter sistemas de e-Learning integrados (actualmente designados por LMS ou LCMS como veremos adiante), tendo em atenção as necessidades dos alunos, os objectivos da instituição e os requisitos inerentes à estruturação dos conteúdos (Chute et al., 1999). Independentemente do grau de aplicação de cada uma das tecnologias, é fundamental que o sistema de e-Learning derive do alinhamento com a teoria pedagógica que melhor se adeqúe à situação de aprendizagem em questão. Por conseguinte, para além da importância das características técnicas das ferramentas de criação, distribuição e gestão de experiências educativas na Web, são fundamentais as características educativas que elas possam proporcionar (grau de comunicação interpessoal e de trabalho colaborativo, forma de acompanhamento do progresso e trajecto do aluno, possibilidades de criação de diversas

modalidades de avaliação e autoavaliação e tipos de acesso e estruturação dos conteúdos de aprendizagem) (Crosetti, 2000).

Nesta última década, têm surgido no mercado plataformas adequadas à criação de ambientes de aprendizagem hipermédia ou sistemas de e-Learning proprietários, comerciais ou *open source*, que integram tecnologias de comunicação para favorecer a interacção síncrona e assíncrona, tecnologias Web para distribuir os conteúdos e implementar ambientes colaborativos (comunidades virtuais) e ferramentas de autor para criar conteúdos multimédia. Actualmente, as plataformas que mais se destacam são:

- *WebCT Campus Edition* e *WebCT Vista* da *WebCT Company* (*http://www.webct.com*);
- *Lotus Learning Management System, Lotus Virtual Classroom* e *Workplace Collaborative Learning* da *IBM Lotus Software* que substituem desde Abril de 2004 as plataformas *LearningSpace Forum, LearningSpace Core* e *LearningSpace Collaboration* (*http://www.lotus.com/lotus/offering3.nsf*);
- *TopClass e-Learning Suite* (*LCMS, LMS, Virtual Classrooms, Mobile, Publisher, Competencies* e *XML Toolkit*) da *WBT Systems Inc.* (*http://www.wbtsystems.com*);
- *Luvit Learning Management System* da *Luvit AB* (*http://www.luvit.com*);
- *Blackboard Academic Suite* (*Learning System, Content System* e *Community System*) da *Blackboard Inc.* (*http://www.blackboard.com/products*);
- *Claroline* do *Centre de Recherche et de Développement de l'ECAM* (*http://www.claroline.net*);
- *Virtual-U* do *Sloan Consortium* (*http://www.virtual-u.org*);
- *Atutor* do *Adaptive Technology Resource Centre* (*http://www.atutor.ca*);
- *Moodle course management system* do *Moodle Pty Ltd* (*http://moodle.org*);

Para além de muitas outras plataformas e sistemas genéricos de gestão de conteúdos na Web, tais como: *Zope* da *Zope Corporation* (*http://www.zope.org*) ou *Mambo* da *Miro International Pty Ltd* (*http://www.mamboserver.com*) ou *Joomla!* da *OSM/Joomlaya Core Team* (*http://www.joomla.org*).

Desde o seu aparecimento, o e-Learning preocupou-se com aspectos tecnológicos inerentes à optimização da infra-estrutura com vista a fornecer serviços e conteúdos para qualquer pessoa, em qualquer lugar. Contudo, o projecto, o desenvolvimento e a avaliação dos

conteúdos de aprendizagem não podem ser descurados. Nesta perspectiva, a filosofia dos objectos de aprendizagem assume-se como a forma mais adequada de desenvolver conteúdos de e-Learning, nomeadamente através da aplicação do modelo de referência ADL SCORM e das especificações IMS *Content Packaging* e IMS *Learning Design*, para além de outros conjuntos de normas e especificações de organizações, tais como: ARIADNE, AICC, IEEE, IMS, entre outras.

2.4- Modelo Pedagógico para o e-Learning

Tal como referimos anteriormente, as tecnologias por si só não são suficientes para obter sistemas de e-Learning adequados às necessidades dos docentes e discentes, pelo que o modelo pedagógico orientado por teorias de aprendizagem e por modelos de organização e estruturação dos conteúdos assume particular importância nos projectos de e-Learning.

2.4.1- Teorias da Aprendizagem

Recuando à década de 50, as teorias de aprendizagem comportamentais defendidas por Watson, Pavlov e Skinner, entre outros, deram origem ao ensino programado ou Instrução Assistida por Computador (CAI – *Computer Assisted Instruction*). A ideia do Comportamentalismo (Behaviorismo) aplicado à aprendizagem é muito simples (Alves, 2003):

- Dividir exaustivamente os conteúdos a aprender;
- Repetir cada conteúdo até aprender a resposta correcta às constantes perguntas que são colocadas a quem aprende;
- Quando a resposta é correcta, dá-se o reforço, caso contrário, dá-se a repetição;
- A situação de avaliação é rigorosamente a mesma que a situação de aprendizagem;
- Todos passam pelas mesmas etapas, embora com ritmos diferentes.

Por conseguinte, nesta corrente de aprendizagem não há espaço para a compreensão ou para a criatividade, uma vez que a mente é uma "caixa negra" sobre a qual nada podemos saber.

Com o aparecimento e vulgarização do computador pessoal e com a emergência de um conjunto de abordagens cognitivistas (R. Miller e Jean Piaget, entre outros) apareceram os primeiros programas computacionais para Instrução Baseada no Computador (CBI - *Computer-Based Instruction*), Formação Baseada no Computador (CBT - *Computer-Based Training*) e Aprendizagem Baseada no Computador (CBL - *Computer-Based Learning*), recorrendo à analogia entre o processamento de informação nos seres humanos e o processamento de informação nas máquinas. Esta analogia caracteriza a teoria do computacionalismo e pode ser resumida em três metáforas: "o cérebro é um computador digital", "a mente é um programa computacional" e "as operações do cérebro podem ser simuladas através de um computador digital" (Castañon, 2005). Tal como o computador, a nossa mente processa símbolos segundo regras para gerar um resultado (*output*). Se a resposta é incorrecta o processamento é deficiente ou inadequado para o problema em questão. O Cognitivismo pode ser resumido nas seguintes ideias (Alves, 2003):

- Saber a resposta correcta não é compreender a resposta;
- Compreender acarreta o uso de capacidades internas;
- As capacidades internas podem não ser inatas mas desenvolvem-se progressivamente durante um dado período de tempo;
- Não podemos ensinar senão em função das capacidades e limites do sujeito;
- Tanto o sucesso como o fracasso da aprendizagem estão associados à construção de uma crença sobre si e sobre os outros (auto-estima, auto-conceito, expectativas, etc).

As teorias da aprendizagem assentam essencialmente na intersecção das questões sobre a natureza da mente com as questões sobre a natureza da cultura. O conhecimento adquirido é mais útil ao aluno, sobretudo quando é descoberto através dos esforços cognitivos do próprio aluno, pois fica então relacionado com o significado ou conhecimento anterior e usado em referência a ele. Mas, compreender alguma coisa de um modo não exclui compreender de outras maneiras, já que a formação e a transformação dos significados ou a construção das representações da realidade estão também intimamente ligadas à cultura, bem como às interacções com os outros indivíduos dos nossos mundos (Bruner, 2000).

Na opinião de Alves, ou mesmo de Bruner, o Cognitivismo poderia desdobrar-se em dois movimentos que se podem resumir em duas formas de cognição distintas, mas complementares:

- Cognitivismo numa óptica racional, intimamente ligada ao conteúdo das matérias, às estruturas do conhecimento científico do indivíduo e ao processo de construção do saber científico;
- Cognitivismo numa óptica social, centrada no "eu" e nos "outros", em que a cognição (acto ou processo de conhecer, que envolve atenção, percepção, memória, raciocínio, juízo, imaginação, pensamento e linguagem) ou a construção do saber são realizadas com os outros.

No contexto de um sistema de e-Learning, podemos associar a primeira perspectiva à interacção com os recursos digitais (ênfase nas estruturas cognitivas internas do sujeito para formar o conhecimento) e às correspondentes preocupações actuais de estruturação e projecto dos conteúdos e das actividades de aprendizagem. E associar a segunda perspectiva à interacção entre professor-aluno e aluno-aluno (ênfase na partilha e colaboração com os outros na construção e compreensão do resultado de aprendizagem pretendido).

Na década de 90, a Internet impulsionou a migração de uma sociedade (pós-)industrial para uma Sociedade da Informação e do Conhecimento e, naturalmente, a aprendizagem deixou de ser influenciada apenas por aspectos cognitivos e passou também a ser influenciada por aspectos sociais (teorias sócio-construtivistas).

O Construtivismo pode ser visto como uma evolução do Cognitivismo ou como um elo que se desprendeu do grande movimento pedagógico conhecido por Escola Nova ou Activa e assumiu diversas correntes: Construtivismo Piagetiano (Jean Piaget e outros), Construcionismo Social (Gergen e outros), Construtivismo Radical (Glasserfeld e outros) e Construtivismo Social (Simons, Latour, Lev Vygotski, James, Dewey, entre outros). Embora estas correntes assumam posições ontológicas e epistemológicas bastante distintas entre si, podemos afirmar que o Construtivismo é uma corrente pedagógica contemporânea que assume que:

- Os estádios de desenvolvimento de um indivíduo operam como estruturas e como sistemas que estão abertos à mudança e contínua transformação;

- O conhecimento não é entregue como um produto acabado. Ele constrói-se pela interacção do indivíduo com o meio físico e social, com o simbolismo humano, com o mundo das relações sociais;
- O conhecimento construído (ou o processo da construção no caso do conhecimento a construir ser científico e universalmente aceite) varia de pessoa para pessoa.

Sem entrar em pormenores inerentes à psicologia cognitiva ou à psicologia da aprendizagem e assumindo uma postura pragmática, podemos afirmar que, na perspectiva do Behaviorismo e na do Cognitivismo (pelo menos parcialmente), o conhecimento é absoluto, existente no mundo exterior e universalmente aceite e transmitido ao aluno sem considerar as suas capacidades de compreensão e criatividade. Mas, na óptica do Construtivismo, ele é relativo, uma vez que a sua construção varia de aluno para aluno. Contudo, as questões do domínio afectivo (estratégias de motivação) não devem ser descuradas, uma vez que, se o aluno não tiver motivação suficiente para aprender, acabará por desistir.

A aprendizagem baseada no e-Learning nem sempre ocorre a partir de ensino que promova interacção entre professor e aluno. Muitas vezes, a aprendizagem deriva do acesso a conteúdos bem estruturados, o que corresponde a uma forma inovadora de aprendizagem que requer um papel mais pró-activo por parte do aluno. Assim, um sistema de e-Learning não só deve apostar no ensino *on-line* (interacção entre professor e aluno simulando o ensino presencial), mas também nos sistemas de Gestão do Conhecimento (*Knowlegde Management*), que devem fornecer o suporte necessário para a criação de uma cultura de aprendizagem na instituição (Goñi et al., 2002).

Se à pedagogia compete autorizar os seres humanos a ir além das suas predisposições "naturais", é seu dever facultar o "estojo de ferramentas" que a cultura desenvolveu para o efeito (Bruner, 2000).

Conscientes de que a classificação exposta e respectiva descrição sucinta das teorias pedagógicas poderá não ser consensual, estamos convictos que muito poucos discordarão de que o modelo educacional para a Sociedade da Informação e do Conhecimento terá a ganhar ao basear-se nos contributos das teorias racionalistas (Cognitivismo e Construtivismo) e, mais propriamente, nas teorias sócio-construtivistas.

Por conseguinte, o modelo pedagógico desejável para o e-Learning assenta na abordagem construtivista, segundo a qual a aprendizagem é um processo de índole social e não apenas cognitivo e individual, através do qual o conhecimento é construído pelo aluno

aquando da interacção da sua base de conhecimentos com as novas experiências de aprendizagem e com outros intervenientes no processo educativo (professores, outros alunos ou mesmo outras comunidades virtuais). No entanto, em determinadas situações de aprendizagem, um sistema de e-Learning deverá permitir o recurso a qualquer outra teoria da aprendizagem, nomeadamente no âmbito do projecto de aprendizagem (tal como veremos adiante no âmbito do *Learning Design*).

Neste contexto, é fundamental dispor de informação para pensar, analisar, processar e discutir, uma vez que o conhecimento não reside apenas nos professores. Mas, informação em demasia pode implicar entropia para o processo de aprendizagem, pelo que cada vez mais se exige que o ambiente de aprendizagem seja um parceiro na organização, estruturação e filtragem da informação e na construção do conhecimento e do saber.

Tal como no meio empresarial, já não é tanto no produto que se encontra o verdadeiro interesse e valor educativo de um trabalho escolar, mas sim no processo da sua obtenção. Não é tanto a chegada mas antes o caminho que se percorreu até lá (Freitas, 1999). Satisfazer as necessidades de aprendizagem dos alunos é, em primeira instância, a chave para garantir o sucesso de um modelo de e-Learning.

Ao longo da última década, surgiram diversas abordagens e modelos para minimizar os factores que tipicamente levam a que os projectos de e-Learning falhem.

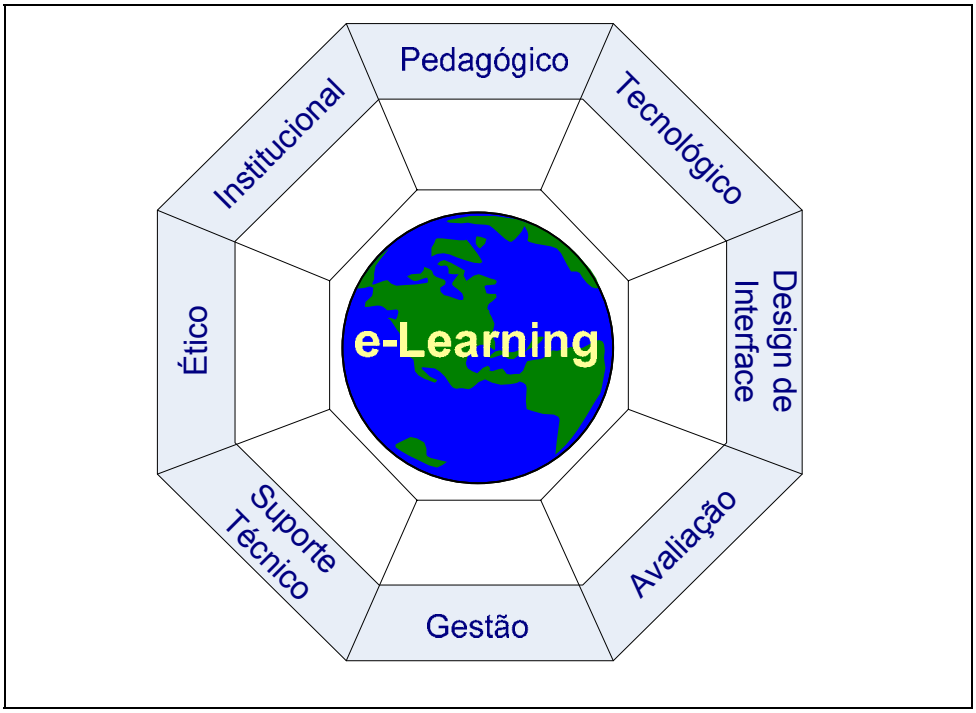


Figura 4 – Framework para e-Learning

No âmbito do projecto e desenvolvimento de sistemas de e-Learning, destacamos os estudos do Professor Badrul Khan (2006) que apontam para uma extensa lista de factores críticos de sucesso, organizados em torno de oito dimensões-chave que reúnem os seguintes aspectos (Khan, 2004): pedagógicos; tecnológicos; *design* de interface; avaliação; gestão; suporte técnico; éticos; e institucionais (ver Figura 4). Cada uma destas dimensões pode ser subdividida em sub-dimensões compostas por factores específicos a ter em conta aquando do projecto de um sistema de e-Learning. Estes factores permitem questionar um vasto conjunto de aspectos durante a análise e especificação de um sistema de e-Learning.

2.4.2- Modelos para a Estruturação de Conteúdos

Um modelo de ensino/aprendizagem tem como principal finalidade projectar "como se ensina" e "como se aprende". Na secção anterior, analisámos sucintamente a influência das teorias pedagógicas; agora, interessa abordar a forma de ensinar no e-Learning, ou seja, os modelos de estruturação de conteúdos nos e-cursos.

Segundo Lima e Capitão (2003) e numa perspectiva construtivista, a estruturação de conteúdos para o e-Learning pode ser:

- Orientada pela definição dos objectivos da instrução do domínio cognitivo, com base nas taxinomias de Bloom (predilecta das Ciências da Educação), Gagné, Reigeluth e Moore;
- Orientada à aprendizagem pela resolução de problemas através de modelos, tais como:
    - CLE – *Constructivist Learning Environments* (Jonassen, 1999);
    - OLE – *Open Learning Environments* (Hannafin et al., 1999);
- Orientada à aprendizagem pela instrução directa através do modelo SOI - *Selecting, Organizing, Integrating* (Mayer, 1999);
- Baseada em princípios elementares de instrução (Merrill, 2002);
- Baseada em princípios de motivação do aluno através do modelo ARCS – *Attention, Relevance, Confidence, Satisfaction* (Keller, 1999).

A tendência de evolução dos modelos para o desenvolvimento de conteúdos tem denunciado uma passagem de modelos de planeamento sequenciais para modelos de planeamento que procuram integrar as perspectivas construtivistas e múltiplas competências. Nesta perspectiva, destacamos os seguintes modelos para o desenvolvimento de e-cursos:

- Modelo proposto por Kemp, Morrison e Ross (1998): ênfase nas necessidades e tipo de alunos, actividades, objectivos, sequência, estratégias de ensino, conteúdos, distribuição, método de avaliação e recursos;
- Modelo ADDIE – *Analysis, Design, Development, Implementation, Evaluation* (Kruse e Keil, 2000; Strickland, 2006): é um modelo genérico que tem servido de base para outras propostas, já que reúne consenso em relação às principais fases para o desenvolvimento de e-cursos. Durante a análise, desenvolve-se uma clara compreensão dos resultados ou comportamentos desejados e dos conhecimentos e habilidades dos destinatários: identificam-se os requisitos, especifica-se o perfil do aluno e suas motivações de aprendizagem, verificam-se os recursos educativos necessários, para além de se analisarem questões administrativas, técnicas ou mesmo didácticas. A fase do projecto documenta os objectivos de aprendizagem específicos, os instrumentos de avaliação, os exercícios e os conteúdos, para além de especificar a interface e a estrutura dos conteúdos de acordo com os requisitos identificados, a especificação da estrutura e da interface e as limitações administrativas e tecnológicas. A criação dos materiais de aprendizagem é concluída na fase do desenvolvimento. Durante a execução, estes materiais são disponibilizados aos estudantes para assegurar que os objectivos de aprendizagem estabelecidos são atingidos. Finalmente, a eficácia dos materiais de aprendizagem será avaliada;
- Modelo R2D2 – *Reflective, Recursive Design and Development* (Willis, 1995): ênfase no projecto e desenvolvimento recursivos e reflexivos, apostando num processo interactivo, não linear e participado, envolvendo todos os elementos da equipa de produção do e-curso;
- Modelo de Smith e Ragan (Smith e Ragan, 1999): é um modelo sistemático e interactivo que engloba três fases: análise, desenvolvimento da estratégia e avaliação.

Podemos portanto afirmar que, independentemente do modelo seguido para o desenvolvimento de e-cursos, um bom ambiente de aprendizagem e-Learning resultará

essencialmente da integração dos aspectos pedagógicos com factores tecnológicos e de recuperação e gestão da informação (ver Figura 5).

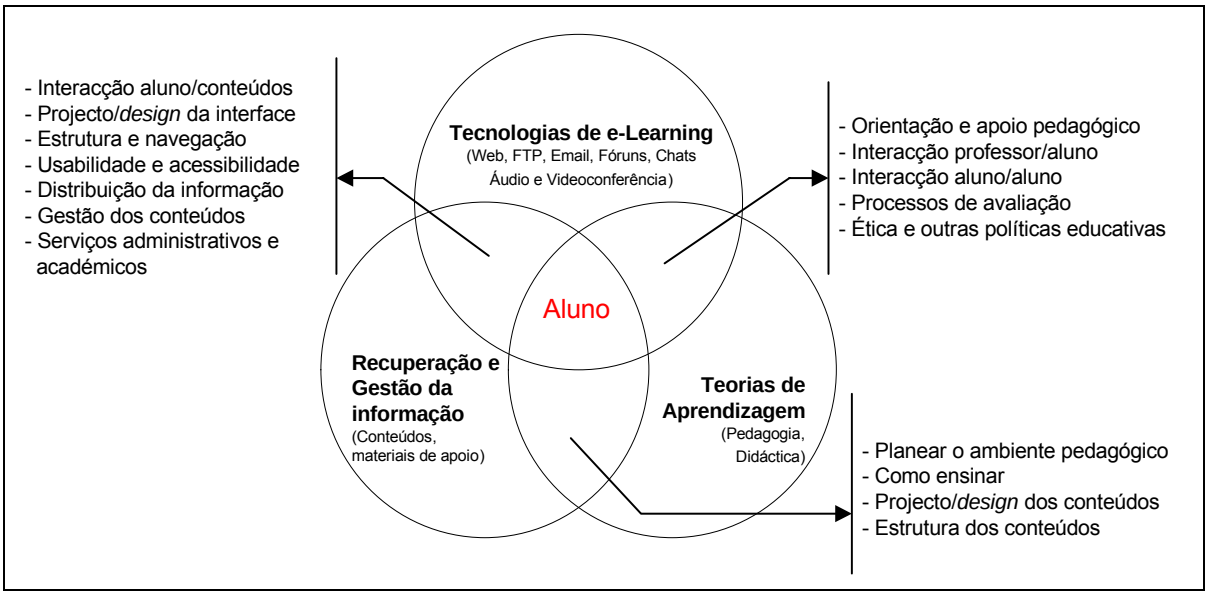


Figura 5 – Factores essenciais no desenvolvimento de um Sistema de e-Learning

A estruturação do conhecimento é a base de um sistema de gestão do conhecimento para a implementação do e-Learning. Um conteúdo bem estruturado, preciso e de fácil utilização favorece a ocorrência de aprendizagem (Goñi et al., 2002).

No âmbito desta investigação, destacamos a interacção do aluno com os conteúdos. Por conseguinte, a produção, organização, estruturação e recuperação de conteúdos de aprendizagem assumem particular importância.

A produção de conteúdos deve ser assegurada por um conjunto de especialistas de vários domínios do conhecimento a trabalhar em equipa. No entanto, muitas vezes essa equipa resume-se ao professor ou formador do e-curso. Reduzir a necessidade de experiência e conhecimento das pessoas envolvidas traduz-se na simplificação do modelo de desenvolvimento de e-cursos. Se o autor dos conteúdos for o responsável pelo processo de desenvolvimento e de publicação, então estamos a reduzir a necessidade de especialistas em e-Learning, ou mesmo de pedagogos, como pode acontecer com o projecto instrucional que abordaremos mais adiante neste capítulo. Portanto, a produção, estruturação, recuperação e gestão de informação deverão ser assistidas por software que siga determinadas normas e especificações e, eventualmente, por agentes de software especialistas.

As normas disponibilizadas pelo IEEE-LTSC, as especificações publicadas pelo IMS *Global Consortium*, os modelos de referência tais como os desenvolvidos pela ADL e as

linguagens de modelação educativa, entre outras normas, especificações, modelos e ferramentas disponibilizados por outros consórcios ou grupos de trabalho, constituem alguns dos mais recentes contributos para o desenvolvimento e distribuição de e-cursos.

2.5- Tecnologias ao Serviço do e-Learning

Os objectos de aprendizagem assumem-se como agentes catalizadores na forma de planear e produzir conteúdos para e-Learning devido às suas potencialidades de reutilização e interoperabilidade (Wiley, 2002). Recorrendo à utilização de objectos de aprendizagem, que permitem criar bibliotecas de conhecimento compostas por blocos de informação capazes de transmitir significado de forma independente, é possível que cursos diferentes utilizem um mesmo objecto. Consequentemente, têm vindo a aparecer no mercado Sistemas de Gestão de Conteúdos de Aprendizagem (LCMSs – *Learning Content Management Systems*) que, tal como o próprio nome indica, têm como principal finalidade a gestão de conteúdos de aprendizagem, permitindo a sua concepção, armazenamento e reutilização em vários cursos. Complementarmente, têm também surgido Sistemas de Gestão de Aprendizagem (LMSs – Learning Management Systems) que têm como principal objectivo automatizar a componente administrativa da formação (gerir processos de inscrição e de avaliação de alunos, gestão das vistas e dos meios de comunicação) (Lima e Capitão, 2003).

Genericamente, um LMS encarrega-se da gestão administrativa das várias formas de ensino/aprendizagem intrínsecas a uma instituição, enquanto que um LCMS concentra-se na gestão dos conteúdos de aprendizagem. Embora tenham finalidades significativamente diferentes, estes sistemas complementam-se, sendo por isso muitas vezes confundidos. A Tabela 1, baseada na investigação conduzida pela *Brandon Hall Research* (Brandon, 2002) e no relatório *The Learning Content Management System - A New eLearning Market Segment Emerges*, publicado pela IDC (*International Data Group*) (Brennan et al., 2001), resume as principais diferenças entre estes dois tipos de sistemas.

A evolução das plataformas de e-Learning tem mostrado uma clara tendência de integração dos LMSs com os LCMSs. Um LMS gere comunidades de utilizadores e disponibiliza o acesso a objectos de aprendizagem armazenados e geridos por um LCMS.

Características ou funcionalidades	**LMS**	**LCMS**
Principais destinatários: alunos, professores e gestores de aprendizagem (comunidade educativa em geral)	Sim	---
Principais destinatários: Produtores de conteúdos, *designers* ou projectistas de conteúdos ou mesmo alunos que necessitem de conteúdos personalizados	---	Sim
Gestão centrada no desempenho dos alunos, nos seus requisitos de aprendizagem e nos planos de aprendizagem	Sim	---
Gestão centrada nos conteúdos de aprendizagem	---	Sim
Gestão da aprendizagem *on-line*	Sim	Sim
Gestão de formas tradicionais de formação (por exemplo: formação centrada no professor)	Sim	Não
Suporte para b-Learning	Sim	Não
Gestão do perfil do aluno e dos processos de registo e inscrição nas unidades de formação	Sim	Não
Gestão de funcionalidades administrativas (por exemplo: registo da assiduidade)	Sim	Não
Suporte para a colaboração entre alunos	Sim	Sim
Relatórios de desempenho (interacção e progresso dos alunos e resultados da formação)	Sim	Sim
Suporte para a partilha de dados com sistemas integrados de gestão ERP (*Enteprise Resource Planning*)	Sim	Não
Distribuição de conteúdos fornecendo interfaces para os alunos e controlo na navegação	Não	Sim
Suporte para a criação de conteúdos de aprendizagem	Não	Sim
Organização de conteúdos reutilizáveis	Não	Sim
Ferramentas de *workflow* para a gestão do processo de criação de conteúdos	Não	Sim
Instrumentos para criação e gestão de testes, quer para a avaliação da aprendizagem, quer para a avaliação da plataforma	Sim	Sim
Instrumentos para análise de competências e habilidades do aluno	Sim	Não
Suporte para avaliação dinâmica e aprendizagem colaborativa	Não	Sim
Agenda para programação dos eventos	Sim	Não

Tabela 1 – Comparação entre LMS e LCMS

Mas os LMSs/LCMSs têm-se preocupado essencialmente com a distribuição de objectos de aprendizagem. Reconhece-se que o uso de objectos de aprendizagem é muitas vezes conotado com o acto de folhear um livro, levando os alunos a ter a sensação de que estão perante conteúdos estáticos e, consequentemente, provocando a sua desmotivação. Reconhece-se também que a aprendizagem pode ocorrer sem objectos de aprendizagem (sem o consumo de conteúdos), uma vez que a aprendizagem pode ocorrer quando os alunos colaboram/cooperam para resolver problemas. Assim, os aspectos pedagógicos não podem ser negligenciados nos ambientes de aprendizagem baseados na Web.

Nestes últimos anos, temos assistido a uma constante evolução das normas, especificações, modelos e guias de boas práticas para ambientes de e-Learning, com vista a estimular a reutilização, permitir a interoperabilidade e flexibilizar ou personalizar o processo de aprendizagem.

A crescente utilização das tecnologias para o e-Learning pelos professores/alunos tem vindo a traduzir-se num aumento dos requisitos, nomeadamente ao nível dos aspectos pedagógicos.

Tanto os últimos relatórios realizados, como os mais recentes artigos publicados no âmbito do e-Learning convergem para o mesmo objectivo: a aprendizagem personalizada. O *e-Learning Programme* (2004-2006) atribui uma atenção significativa a este objectivo ao referir que é importante assegurar que os produtos de e-Learning sejam capazes de ter em conta as necessidades e estilos de aprendizagem individuais.

Por conseguinte, um ambiente de e-Learning baseado em sequências de actividades de aprendizagem parece reunir mais consenso e implicar maiores vantagens para o processo de aprendizagem do que uma abordagem baseada em sequências de conteúdos ou de objectos de aprendizagem. Nesta perspectiva, o *Instructional Design* (projecto instrucional) ou, mais concretamente, o *Learning Design* (projecto de aprendizagem) não podem ser descurados. Actualmente, a norma IMS-LD (IMS *Learning Design*) é o contributo que mais se tem destacado neste contexto.

A organização e representação dos conceitos subjacentes aos LOs também teriam a ganhar com a integração dos mapas de conceitos (instrumento usado com sucesso há quase três décadas no âmbito das Ciências da Educação) no e-Learning.

2.5.1- ADL SCORM e Objectos de Aprendizagem

Nos últimos anos, surgiram vários termos alternativos para designar materiais ou recursos educativos, tais como: objectos de conteúdo (*Content Objects*), objectos de ensino (*Teaching Objects*), objectos de formação *(Training Objects*), objectos de instrução (*Instructional Objects*) e objectos de aprendizagem (*Learning Objects*). Uma vez que nem todas essas designações consideram que o objecto se destina à aprendizagem, no contexto desta investigação usamos o termo mais utilizado no âmbito do e-Learning: Objecto de Aprendizagem ou *Learning Object* (LO).

2.5.1.1- Objectos de Aprendizagem

Os LOs baseiam-se na filosofia da programação por objectos das ciências da computação, pelo que a ideia é construir pequenas peças de instrução para serem reutilizadas em diferentes contextos de aprendizagem, como se de peças de LEGO se tratasse.

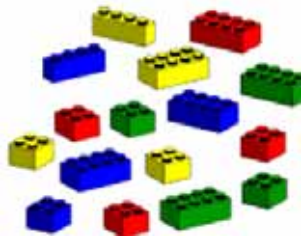
	
Os LOs são como peças LEGO que têm em comum o mesmo modelo.	Os LOs (ou peças LEGO) que partilham o mesmo modelo podem ser encaixadas e reutilizadas.

Figura 6 – Metáfora LEGO aplicada aos LOs

Um LO é um recurso digital (texto, imagem, som, vídeo, *applet* Java, filme flash, programa de simulação, entre outros componentes distribuídos por intermédio de *plug-ins* apropriados) que pode ser reutilizado para apoiar a aprendizagem (Wiley, 2002).

Um LO pode ser visto como uma unidade de conteúdo de aprendizagem, independente e autónoma, que pode ser reutilizada em diversos contextos educativos ou instrucionais (Polsani, 2003).

Embora alguns autores excluam os objectos não digitais, o LOMWG (*Learning Objects Metadata Working Group*) do IEEE-LTSC apresenta uma definição mais ampla: LO é uma qualquer entidade, digital ou não (incluindo livros ou outros documentos em papel, cassetes de vídeo e áudio e CD-ROMs ou outros suportes educativos digitais ou não), que pode ser utilizada na aprendizagem, educação ou formação (IEEE-LTSC, 2002).

Uma discrepância na definição deriva da aceitação ou não de objectos de aprendizagem não digitais, outra deriva da área de aplicação do objecto de aprendizagem. No contexto desta tese, utilizamos o termo LO para descrever unidades digitais, independentes e autónomas, passíveis de serem (re)utilizadas em distintos contextos de (auto)aprendizagem.

Ainda que não exista uma definição amplamente aceite, alguns requisitos parecem reunir consenso: reutilização, interoperabilidade, durabilidade e acessibilidade.

A reutilização corresponde à capacidade de incorporar conteúdos em múltiplas aplicações e contextos (fácil de usar e de modificar). A interoperabilidade refere-se à

capacidade de intercâmbio de conteúdos entre diferentes plataformas (adapta-se facilmente a diferentes hardwares, softwares ou *browser*s). A durabilidade equivale à capacidade de garantir a operacionalidade dos conteúdos com a mudança da tecnologia (não necessita de alterações significativas face a novas versões do sistema de e-Learning). E a acessibilidade é a capacidade de aceder remotamente a conteúdos e de os distribuir por diferentes localizações (ser pesquisável e estar disponível para quem dele necessita).

Convém salientar que a aplicação dos LOs não se limita apenas aos sistemas de e-Learning. Progressivamente, eles têm vindo a ser usados também em bibliotecas, museus ou outras entidades do saber com vista a construir repositórios de LOs e desenvolver ambientes virtuais de aprendizagem ou aplicações multimédia educativas *on-line*. Com o objectivo de usar os recursos de informação destas entidades no contexto educativo, assegurando que os mesmos estejam remotamente disponíveis, surgiram alguns projectos para estudar as mudanças a implementar nos sistemas de informação, usando a representação dos recursos na perspectiva de LOs. Um exemplo é o *The Nacional Museum of Australia* (Payne e Peacock, 2004).

Os LOs são normalmente descritos por metadados e geridos pelos LCMS que os criam, armazenam, combinam e distribuem (Mortimer, 2002). A associação de metadados aos LOs permite que os mesmos possam ser distribuídos individualmente ou combinados com outros, formando conteúdos de aprendizagem maiores, para além de facilitar a recuperação de um determinado objecto de aprendizagem quando submetida uma pesquisa no LCMS.

Mas a recuperação e a reutilização dos LOs são influenciadas pelo grau de detalhe dos conteúdos (granularidade). A granularidade corresponde ao nível de detalhe de um componente ou parte dos conteúdos de aprendizagem existentes em materiais de aprendizagem. Uma disciplina ou curso de formação possui elevada granularidade se for composta por vários LOs de tamanho reduzido. Assim, estes LOs podem ser combinados para compor outro curso ou uma peça de aprendizagem maior.

A granularidade de um LO pode variar da simples imagem ou gráfico até ao currículo completo de uma lição ou curso (LOMWG, 2002). Não obstante, quanto maior for a dimensão do LO, menor será a sua possibilidade de reutilização. Logo, quanto maior for a granularidade de uma disciplina ou curso de formação, maior é a sua flexibilidade e, consequentemente, do sistema de e-Learning.

Actualmente, existem diversas normas e especificações que documentam regras, directrizes e características para a concepção e projecto de LOs, permitindo que os mesmos sejam desenvolvidos e utilizados independentemente da ferramenta de autoria, do ambiente de aprendizagem e da plataforma de hardware e software.

A catalogação de LOs usando a especificação IMS-LOM (*IMS Learning Object Metadata*) (IMS, 2004), que inclui indicações para a interoperabilidade, permitindo reutilização, foi o primeiro passo rumo à normalização da produção de conteúdos para e-Learning. Essa especificação deu origem à norma IEEE-LOM 1484.12 (*IEEE Standard for Learning Object Metadata*) e tem por objectivo garantir a interacção entre diferentes sistemas de e-Learning e contribuir para a redução de custos e para a flexibilização da produção de recursos educacionais, através de especificações inerentes aos metadados, aos conteúdos, ao nível de gestão do sistema, aos perfis de alunos e professores e ao interface entre o nível de gestão e o exterior (Ferreira e Santiago, 1999; Goñi et al., 2002). Esta norma será detalhada mais à frente no âmbito das tecnologias para metadados no capítulo III – Camadas estrutural e sintáctica da Web Semântica.

2.5.1.2- Modelo SCORM

Com vista a promover a utilização da tecnologia de metadados LOM, assegurar a interoperabilidade e partilha dos LOs e a compatibilidade entre os sistemas de e-Learning, diversas organizações (ADL, ARIADNE, IMS, IEEE, AICC, entre outras) uniram esforços para o desenvolvimento de um modelo de referência que normalizasse a produção de conteúdos de aprendizagem. Este modelo, designado por SCORM (*Sharable Content Object Resource Model*), é um conjunto de normas, especificações e orientações técnicas para o desenvolvimento de conteúdos de aprendizagem, de forma a garantir a reutilização, interoperabilidade, durabilidade e acessibilidade (ADL, 2004).

O modelo SCORM não é mais do que um manual de boas práticas composto de 4 manuais técnicos (ADL, 2004):

a) *SCORM Overview Book*: representa a introdução aos conceitos chave do SCORM, entre outras informações conceptuais;

b) *SCORM Content Aggregation Model* (CAM): modelo de agregação de conteúdos que especifica como encontrar, combinar, agregar, descrever, sequenciar e mover recursos de aprendizagem, usando metadados na importação ou exportação entre sistemas;

c) *SCORM Run-Time Environment* (RTE): ambiente de execução que especifica como executar os conteúdos e como registar o percurso do aluno, tendo como objectivo a interoperabilidade entre recursos de aprendizagem e LMSs;

d) *SCORM Sequencing and Navigation* (SN): modelo de sequenciação e navegação que descreve como os conteúdos podem ser ordenados para o aluno.

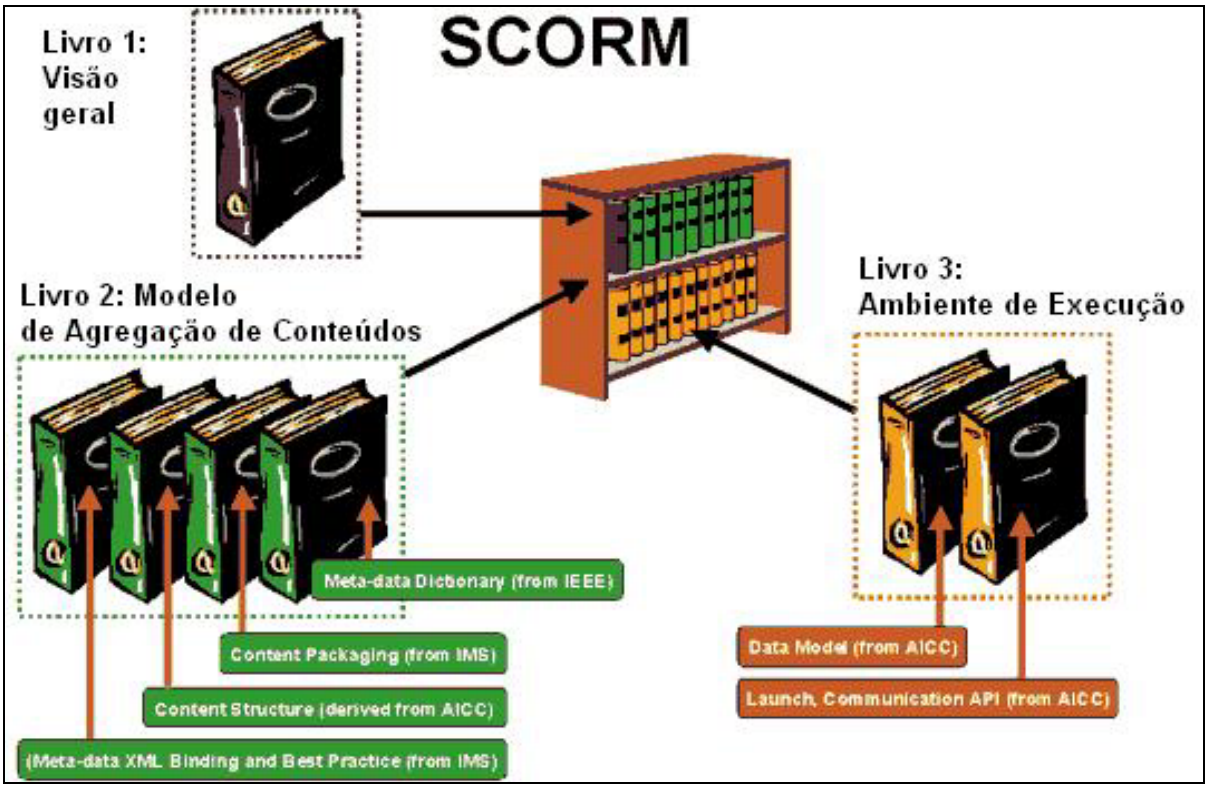


Figura 7 – Manuais técnicos do SCORM

Em primeira instância, o modelo SCORM visa normalizar e promover a utilização de objectos de aprendizagem e de metadados, assegurar a interoperabilidade e partilha dos recursos educativos, bem como a compatibilidade entre os sistemas de e-Learning de diversas organizações.

Em última instância, o SCORM apresenta um modelo de dados comum que os cursos podem usar para trocar dados com o LMS, permitindo que os cursos sejam acompanhados de uma forma que outras soluções descuravam. Ou seja, não basta que os alunos façam um curso, é necessário recolher dados (por exemplo: o número de vezes que acederam a um

determinado conteúdo, a sequência de aprendizagem, o número de tentativas bem ou mal sucedidas para completar um questionário ou um trabalho, a avaliação da aprendizagem do conteúdo ou o tempo de permanência no mesmo), agregá-los e reagir em conformidade com as conclusões obtidas.

O desenvolvimento, o intercâmbio e a reutilização de objectos de aprendizagem permitem optimizar os sistemas de e-Learning facilitando a criação novos e-cursos e mover objectos de aprendizagem entre e-cursos ou mesmo entre sistemas de e-Learning. Nesta perspectiva, o recurso a metadados IMS LOM e ao IMS-CP (IMS *Content Packaging*) é crucial. IMS-CP não é mais do que uma especificação que permite gerar um pacote em formato zip (formato de compressão), contendo todos os conteúdos e um ficheiro XML que inclui os metadados, a sequência de navegação e todos os recursos associados. Para auxiliar este processo, existem ferramentas de geração de metadados e ferramentas com interface gráfica que permitem desenvolver cursos Web segundo as normas SCORM, tais como:

- Editor RELOAD – *Metadata and Content Package Editor*: ferramenta que permite o desenvolvimento de pacotes de conteúdos SCO (*Shareable Content Packing*) e edição de metadados em conformidade com as normas IMS/SCORM. O SCORM Player permite a visualização e navegação em pacotes de conteúdos SCO (*http://www.reload.ac.uk/editor.html*);

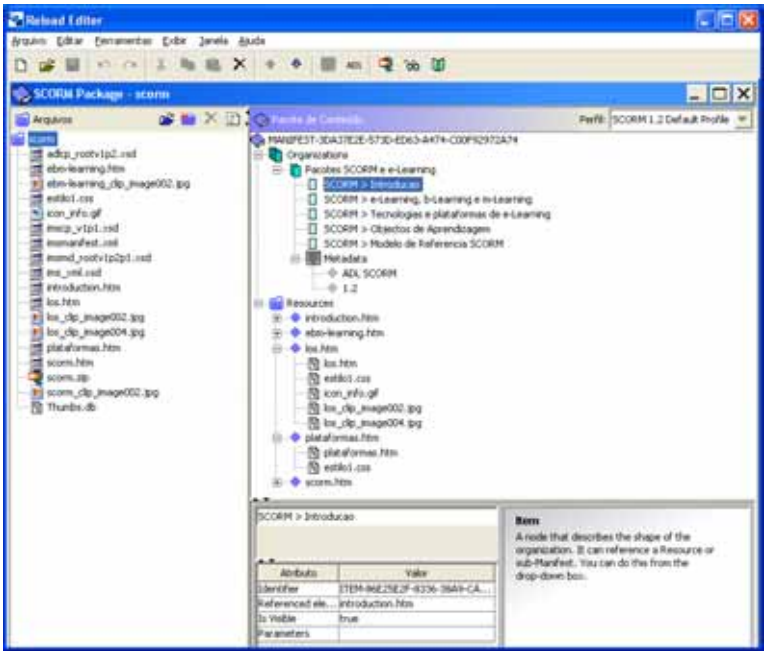


Figura 8 – Editor RELOAD

- Editor eXe – *E-learning XHTML Editor*: suporta a especificação *Dublin Core Metadata* para a edição de metadados e SCORM/IMS para o empacotamento de conteúdos (*http://exelearning.org*);

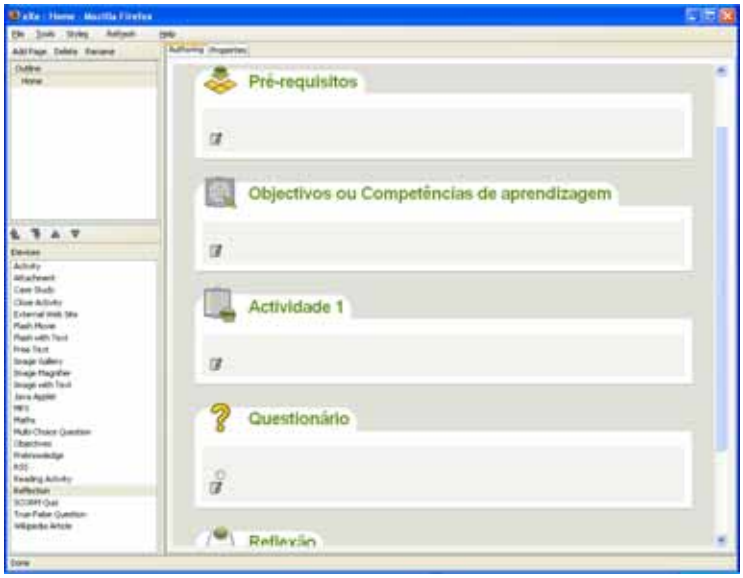


Figura 9 – Editor eXe

- Editor LomPad: ferramenta para a edição de metadados de objectos de aprendizagem. Suporta as especificações IEEE-LOM, SCORM e CANCORE (*https://sourceforge.net/projects/lompad*);

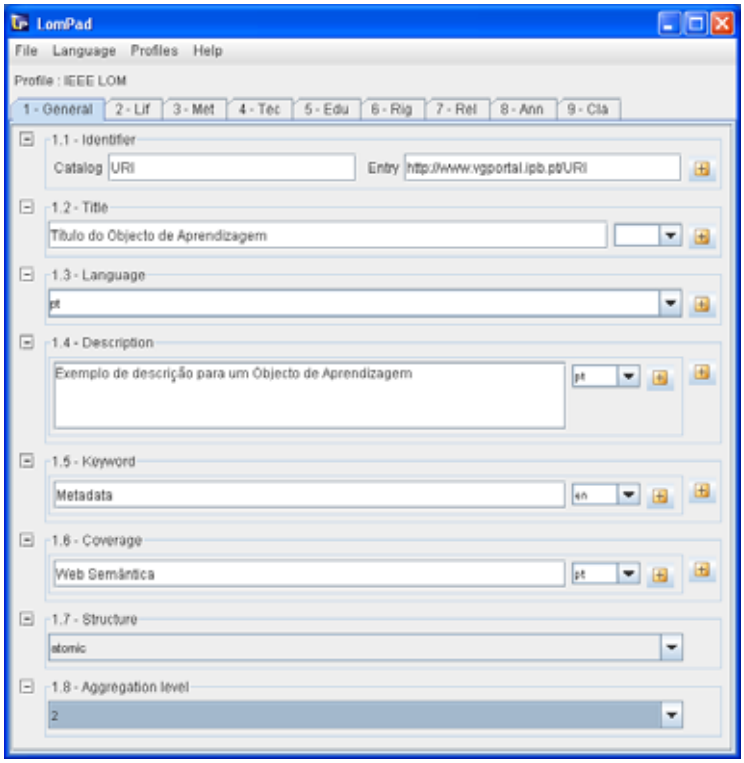


Figura 10 – Editor de metadados LomPad

- SCORMxt: permite o desenvolvimento e empacotamento de e-cursos em conformidade com as normas SCORM, a adição de metadados e a publicação *on-line* ou para impressão em papel. Esta ferramenta funciona como uma extensão do QuarkXPress V6.5 (*http://www.westcliffdata.co.uk*).

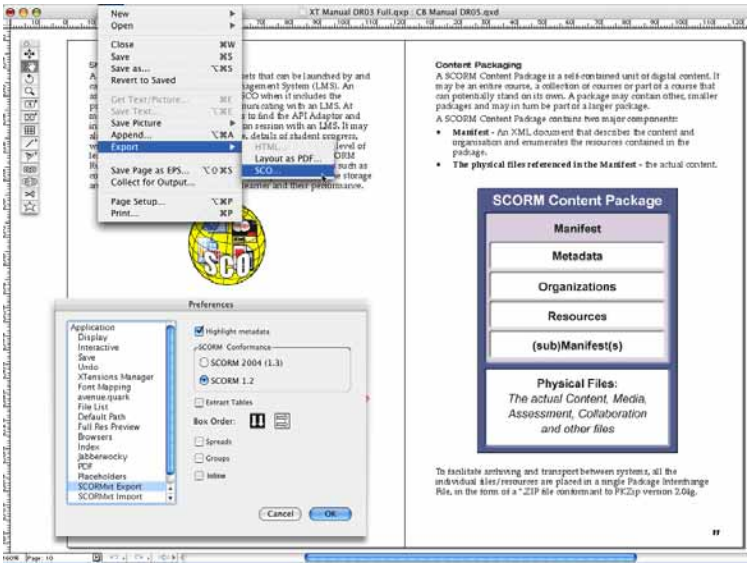


Figura 11 – Editor SCORMxt

2.5.2- Instructional Design e Learning Design

A Educação contemporânea tem vindo gradualmente a considerar o Construtivismo uma importante perspectiva educativa e, no âmbito de cenários de aprendizagem mais complexos, a integrá-lo com outras teorias da aprendizagem.

O modelo de Educação ao longo da vida preconizado pela Sociedade da Informação e do Conhecimento sugere a inclusão de novos métodos no processo de ensino/aprendizagem, de modo a que os alunos atinjam os conhecimentos, objectivos, competências, destrezas e valores indispensáveis para responder aos preceitos da sociedade actual.

No contexto da criação e representação de conteúdos de aprendizagem, podemos apresentar a evolução dos sistemas de e-Learning em 4 etapas (Torres et al., 2006):

1) A primeira etapa caracterizou-se pela publicação de disciplinas ou cursos de formação compostos por conteúdos indivisíveis ou inseparáveis (a baixa granularidade dos cursos e dos LOs muito dificilmente permitia a sua combinação e reutilização).

2) A segunda etapa apostou na publicação dos conteúdos de aprendizagem através de aplicações simples (páginas de texto, páginas HTML e alguns recursos multimédia). A interacção com os conteúdos centra-se em objectos de aprendizagem que podiam ser anotados com metadados e distribuídos através de IMS-CP. A granularidade, a definição dos objectivos, a combinação, a reutilização e a interoperabilidade dos LOs passaram a ser as principais preocupações no desenvolvimento de e-cursos.

3) A terceira etapa tem vindo a apostar em especificações para agregar os conteúdos educativos, nomeadamente especificações simples centradas em sequências de objectos de aprendizagem, tais como, IMS-SS (*Simple Sequencing*), impulsionando a normalização através do SCORM.

4) A quarta etapa avança lentamente para a adopção de linguagens de modelação educativas que permitem o desenvolvimento de processos de aprendizagem com maior flexibilidade pedagógico-didáctica. Contrariamente às etapas anteriores, é colocada ênfase na personalização dos conteúdos e promovida uma participação mais activa do aluno no processo de aprendizagem, através da modelação de unidades de aprendizagem (UoLs – *Units of Learning*) compostas, essencialmente, por cenários ou ambientes, objectivos, actividades, recursos ou objectos de aprendizagem, serviços e perfis. Este processo de modelação é conhecido por *Learning Design* e corresponde ao plano ou projecto de aprendizagem baseado num determinado método pedagógico que deriva do *Instructional Design* (processo sistemático de tradução dos princípios gerais de ensino/aprendizagem em planos para materiais de ensino/aprendizagem, ou seja, a análise sistemática das necessidades dos alunos deve resultar na planificação das estratégias e materiais mais adequados para satisfazer essas necessidades). A proposta que mais interesse tem despertado na comunidade científica tem sido a especificação IMS-LD, enquanto especificação que permite a planificação da aprendizagem e a concepção de recursos educativos adequados, independentemente da metodologia pedagógica adoptada.

Assim, o Projecto Instrucional assistido por Tecnologia (IDT - *Instructional Design on Technology*) assume-se como uma perspectiva emergente para a integração das tecnologias educativas nos currículos. Neste sentido, as linguagens de modelação educativa têm vindo a ser usadas para projectar processos de ensino/aprendizagem que podem adaptar-se aos requisitos do contexto educativo.

2.5.2.1- Instructional Design

Genericamente, o *Instructional Design* ou Projecto Instrucional pode ser descrito como uma teoria que orienta o planeamento dos meios necessários para que se alcancem os resultados de aprendizagem: estabelecer os objectivos da aprendizagem ou competências a adquirir; identificar os métodos pedagógicos que facilitam a ocorrência de aprendizagem e projectar não só os meios que permitem atingir os resultados da aprendizagem, mas também os meios que demonstram que esses resultados foram de facto atingidos.

O Projecto Instrucional pode incorporar e correlacionar diversas teorias de aprendizagem (comportamentalismo, cognitivismo, constructivismo, etc), métodos ou técnicas didácticas (método expositivo, método baseado em casos, aprendizagem baseada em problemas, aprendizagem colaborativa, aprendizagem baseada em projectos, etc) e métodos ou modelos de avaliação (ficha de avaliação convencional, auto e hetero-avaliação, avaliação de trabalhos, avaliação do portfólio, avaliação dos resultados da aprendizagem, etc).

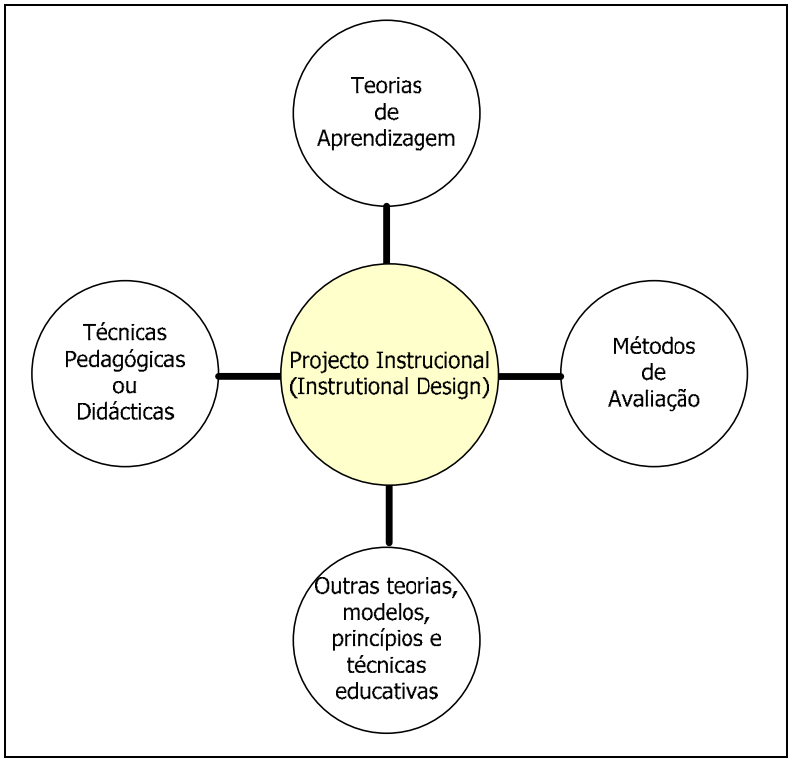


Figura 12 – Projecto Instrucional

Actualmente, existe uma preocupação crescente em perceber de que forma o IDT pode influenciar os processos educativos, nomeadamente naqueles cenários mais complexos em que o projecto instrucional integra várias teorias, modelos, princípios e técnicas de aprendizagem (Torres et al., 2006).

De modo a facultar uma melhor compreensão do Projecto Instrucional como um todo, apresentamos sucintamente a proposta da Universidade Aberta da Holanda (*Open University of The Netherlands*).

Esta proposta teve como motivação responder a uma questão: "Onde está a aprendizagem no e-Learning? (*Where is the learning in e-Learning?*)". A distribuição de informação através do e-Learning não é suficiente para garantir a aprendizagem. É necessário que essa informação seja aprendida pelo aluno através de actividades que permitam gerar novo conhecimento. Assim, esta proposta corresponde a uma investigação educativa, que incidiu especificamente no campo da psicologia da aprendizagem e do projecto instrucional, e traduz-se essencialmente num meta-modelo pedagógico (modelo para modelar modelos pedagógicos) expresso em diagramas UML.

O meta-modelo pedagógico é composto por quatro pacotes principais e baseia-se no conceito de "unidade de estudo", que corresponde à menor unidade capaz de promover eventos de aprendizagem para os alunos, satisfazendo um ou mais objectivos de aprendizagem inter-relacionados (Koper, 2001).

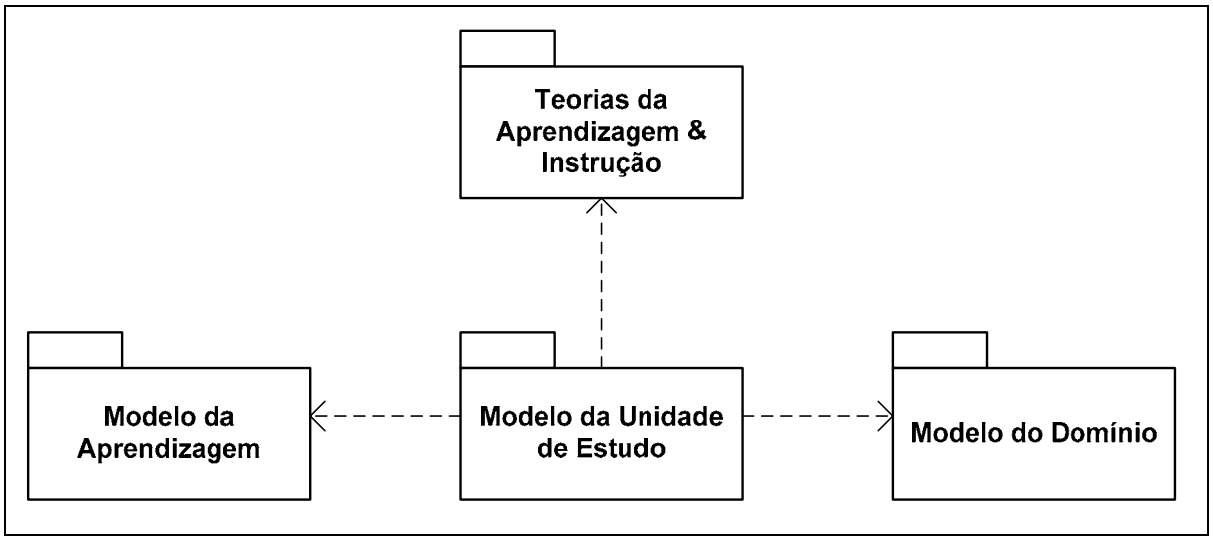


Figura 13 – Pacotes no meta-modelo pedagógico

Os quatro pacotes que compõem o meta-modelo pedagógico são:

- Teorias da Aprendizagem & Instrução: descrevem as teorias, princípios e modelos de instrução tal como são descritos na literatura ou tal como são concebidos e defendidos pelos autores dos mesmos e profissionais que os utilizam. Das várias teorias de aprendizagem, Koper destaca as seguintes correntes principais:

- Empiristas: Comportamentalista ou Behaviorista (*Behaviorist*);
- Racionalistas: Cognitivista (*Cognitivist*) e Construtivista (*Construtivist*);
- Situacionalistas: Construtivista Social (*Social Constructivist*) e Pragmática e Cultural-histórica (*Pragmatist Socio-historic*).

- Modelo da Aprendizagem: descreve como os alunos aprendem com base no consenso generalizado das teorias da aprendizagem;
- Modelo do Domínio: descreve o tipo de conteúdo e a sua organização no âmbito de determinada área do conhecimento (por exemplo: biologia, informática, etc), permitindo uma visão integrada baseada na navegação entre conteúdos;
- Modelo da Unidade de Estudo: descreve como são modeladas as unidades de estudo que são aplicáveis na realidade, de acordo com o modelo de aprendizagem e o modelo de instrução. Este modelo resulta do projecto de aprendizagem (*Learning Design*).

2.5.2.2- Learning Design

O *Learning Design* ou Projecto de Aprendizagem, derivado do *Instructional Design*, pode ser definido como a aplicação do conhecimento do projecto de aprendizagem aquando do desenvolvimento de uma unidade de aprendizagem concreta, por exemplo um curso, uma lição, um currículo, um evento de aprendizagem. Corresponde à descrição de uma metodologia que permite aos alunos atingir determinados objectivos de aprendizagem, executando determinadas actividades de aprendizagem, numa certa sequência, no contexto de um determinado ambiente de aprendizagem (IMS, 2004).

Para facilitar o desenvolvimento da especificação e a sua subsequente implementação, o *Learning Design* é dividido em três partes ou níveis:

- Nível A: inclui as componentes principais do *Learning Design*, fornecendo um modelo comum baseado em diferentes teorias de aprendizagem, nomeadamente a capacidade de especificar o momento em que as actividades serão executadas por alunos e professores, dentro do contexto de um ambiente de objectos ou serviços de aprendizagem;

- Nível B: adiciona propriedades e condições (e mais alguns elementos relacionados), permitindo especificar a personalização, os pré-requisitos, as preferências e a acessibilidade, para além de suportar a aprendizagem baseada em portfólios;
- Nível C: adiciona notificações (e mais alguns elementos relacionados), permitindo que as actividades possam ser ajustadas como consequência de alterações dinâmicas no perfil do utilizador e/ou da ocorrência de eventos durante a execução das actividades de aprendizagem.

Uma das normas que contribuem para a aplicação do *Learning Design* num sistema de e-Learning é a norma IMS-LD, complementada com IMS-CP para distribuir o pacote de conteúdos e IMS-MD/IEEE-LOM para caracterizar os recursos educativos distribuídos.

A IMS-LD é uma especificação do IMS *Global Consortium* para modelar unidades de estudo, unidades curriculares, módulos ou disciplinas. Permite modelar quem faz o quê, quando e com que conteúdos e serviços, com vista a conseguir atingir os objectivos da aprendizagem. Em termos práticos, a criação de um *Learning Design* começa com um esboço de ideias em papel e termina na produção de um documento XML válido.

A IMS-LD baseia-se na linguagem EML (*Educational Markup Language*) desenvolvida pela Universidade Aberta da Holanda e usa a metáfora de uma peça de teatro. Ou seja, um método encontra-se dividido em peças, que contêm diversos actos onde diferentes actores desempenham diferentes papéis sobre um cenário realizando actividades num ambiente específico (Jurado et al., 2006).

A especificação IMS-LD pode ser considerada como uma camada integradora que faz uso, inclui ou é complementada pelas especificações já existentes. A forma de incluir essas especificações é através de XML *Namespaces*. Assim, o IMS-LD deve ser preferencialmente integrado num pacote IMS *Content Package* (especificação IMS *Content Packaging*) para criar a denominada "unidade da aprendizagem". Os metadados IMS-LOM podem ser incluídos nas várias estruturas que compõem o IMS-LD. Também é possível (ou melhor, desejável) incluir IMS *Question and Test Interoperability* (IMS-QTI), IMS *Reusable Competency Definition* (IMS-RCD), IMS *Learner Information Package* (IMS-LIP), IMS *Enterprise*, IMS *Simple Sequencing* (IMS-SS), para além de permitir a inclusão de conteúdos SCORM.

Algumas das limitações apontadas à especificação IMS-LD podem ser minimizadas com a agregação das especificações IMS-LIP (que permitirá inserir suficiente informação

para especificar as características de um utilizador que pertença a um determinado perfil ou com um dado papel) e IMS-SS (que permitirá introduzir um novo modelo de sequência, capaz de permitir uma maior flexibilidade) às actividades de aprendizagem, actividades de suporte, estruturas de actividades, actos, métodos, etc do IMS-LD (Jurado et al., 2006).

Uma vez que o SCORM é a norma adoptada pela grande maioria dos fabricantes de plataformas de LMS/LCMS, prevê-se que a adopção da especificação IMS-LD decorrerá lentamente. Não obstante, as vantagens desta migração justificam claramente o investimento. Actualmente, existem alguns editores e visualizadores ou *players*, dos quais se destacam: RELOAD *Learning Design*, *Coopercore*, LAMS, MOTplus e aLFanet.

- Editor RELOAD LD: suporta completamente a especificação IMS-LD (níveis A, B e C), permitindo a organização e edição de unidades de aprendizagem como um conjunto de "projectos". O projecto LD pode ser validado e exportado como um ficheiro XML IMS-LD juntamente com os metadados associados (IMS-LOM) e os ficheiros de conteúdo num pacote IMS em formato zip (IMS Content Packaging). Esse pacote pode ser visualizado com o RELOAD *LD player* ou com um motor de execução (por exemplo: *Coopercore Learning Design Engine*).

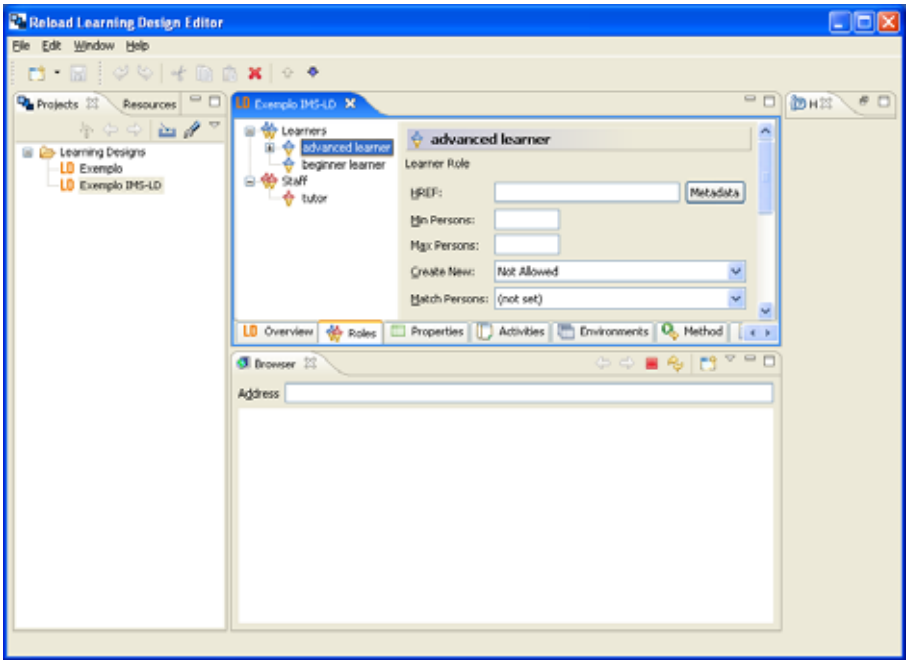


Figura 14 – Editor RELOAD Learning Design

- Editor Motplus: iniciativa da Universidade de Quebec que disponibiliza o editor Mot+. Este editor gráfico suporta o nível A da especificação IMS-LD.

- CooperCore: iniciativa da Universidade Aberta da Holanda que disponibiliza o editor *CopperAuthor* e o motor de execução *CopperCore Learning Design Engine*. O Coopercore suporta os três níveis do IMS-LD (A, B e C). O CooperCore é responsável pelo controlo de todas as actividades, actores e conteúdos de uma unidade de aprendizagem, libertando o ambiente de aprendizagem da complexidade dessas tarefas.
- O LAMS (*Learning Activity Management System*) é uma ferramenta para projectar, gerir e distribuir actividades de aprendizagem colaborativas *on-line*. Fornece aos professores um ambiente autor gráfico e intuitivo para criar sequências de actividades de aprendizagem. Estas actividades podem incluir um nível de tarefas individuais, actividades para pequeno grupo e actividades para grande grupo (turma) baseadas em conteúdos e colaboração. A integração desta ferramenta com as plataformas de e-Learning, como o Moodle, proporciona melhorias significativas no ambiente de aprendizagem, inserindo definitivamente a aprendizagem no e-Learning;
- HyCo (*Hypertext Composer*) é uma ferramenta de autor para a criação de conteúdos multimédia que, desde a versão 3.0, inclui a possibilidade de projectar a aprendizagem com os conteúdos criados. HyCo baseia-se na metáfora Lego para criar o projecto de aprendizagem, permitindo a criação de diferentes componentes do projecto, de tal forma que possam ser integrados para criar componentes mais gerais e assim sucessivamente até chegar ao projecto final de aprendizagem. O primeiro passo na criação do projecto de aprendizagem é a definição dos objectivos e pré-requisitos, agregando os recursos (objectos de aprendizagem) que permitem atingir esses resultados de aprendizagem; e o segundo passo é a inclusão dos componentes (propriedades, perfis, actividades de aprendizagem e estruturas de actividades). Seguidamente, passa-se à criação do fluxo de aprendizagem onde se determina como se devem mostrar as actividades de aprendizagem e a ordem das mesmas. É neste passo que se cria o conjunto de actos que definem a sequência de execução (*play*) das actividades e serviços. A sequência e condições de execução são definidas através dos métodos. O último passo (criação do projecto de aprendizagem) não é mais do que a selecção dos objectivos, dos pré-requisitos e do método que contém o fluxo de aprendizagem a seguir. Finalmente, o projecto de aprendizagem e os respectivos conteúdos podem ser empacotados num ficheiro zip em conformidade com as

especificações IMS-CP. Este pacote zip pode ser visualizado com um *player* ou motor de execução (por exemplo: *SLED Player* ou *Coopercore Learning Design Engine*).

- O projecto aLFanet (*Active Learning For Adaptative interNET*) é uma das iniciativas que mais se destacam no desenvolvimento de um ambiente de aprendizagem personalizado através da Web baseado em diferentes tipos de adaptação. Disponibiliza o editor aLFanet e o *player* aLFanet.

2.5.3- Mapas Mentais, Mapas de Conceitos e Mapas de Tópicos

Em primeira instância, os mapas mentais e mapas de conceitos partilham as mesmas concepções e ideias. São representações mentais da realidade, isto é, representações das estruturas cognitivas que cada indivíduo gerou da realidade. Mais recentemente, surgiu uma abordagem análoga no âmbito das Ciências Documentais denominada mapas de tópicos.

2.5.3.1- Mapas Mentais

Os mapas mentais correspondem a diagramas hierárquicos, criados por Tony Buzan em finais dos anos 60, para representar o conhecimento de forma textual, ilustrada ou ambas, num formato sintético, organizado e por níveis.

São as seguintes as características de um mapa mental (Buzan, 1994):

- É formado por tópicos ligados através de linhas;
- Cada tópico pode conter texto, ilustração ou ambos;
- Existe um tópico principal, também designado título ou raiz, que possui sub-tópicos ligados ao tópico raiz com linhas (ramos). Cada sub-tópico por sua vez pode ter seus próprios sub-tópicos, resultando numa organização hierárquica ou em árvore onde a raiz corresponde ao tópico principal; os ramos aos sub-tópicos e as folhas aos tópicos sem sub-tópicos;

- Os tópicos formam níveis com graus crescentes de detalhe e especificidade, normalmente mostrados no mapa através da formatação: da raiz para as folhas, as linhas vão ficando mais finas e as fontes menores.

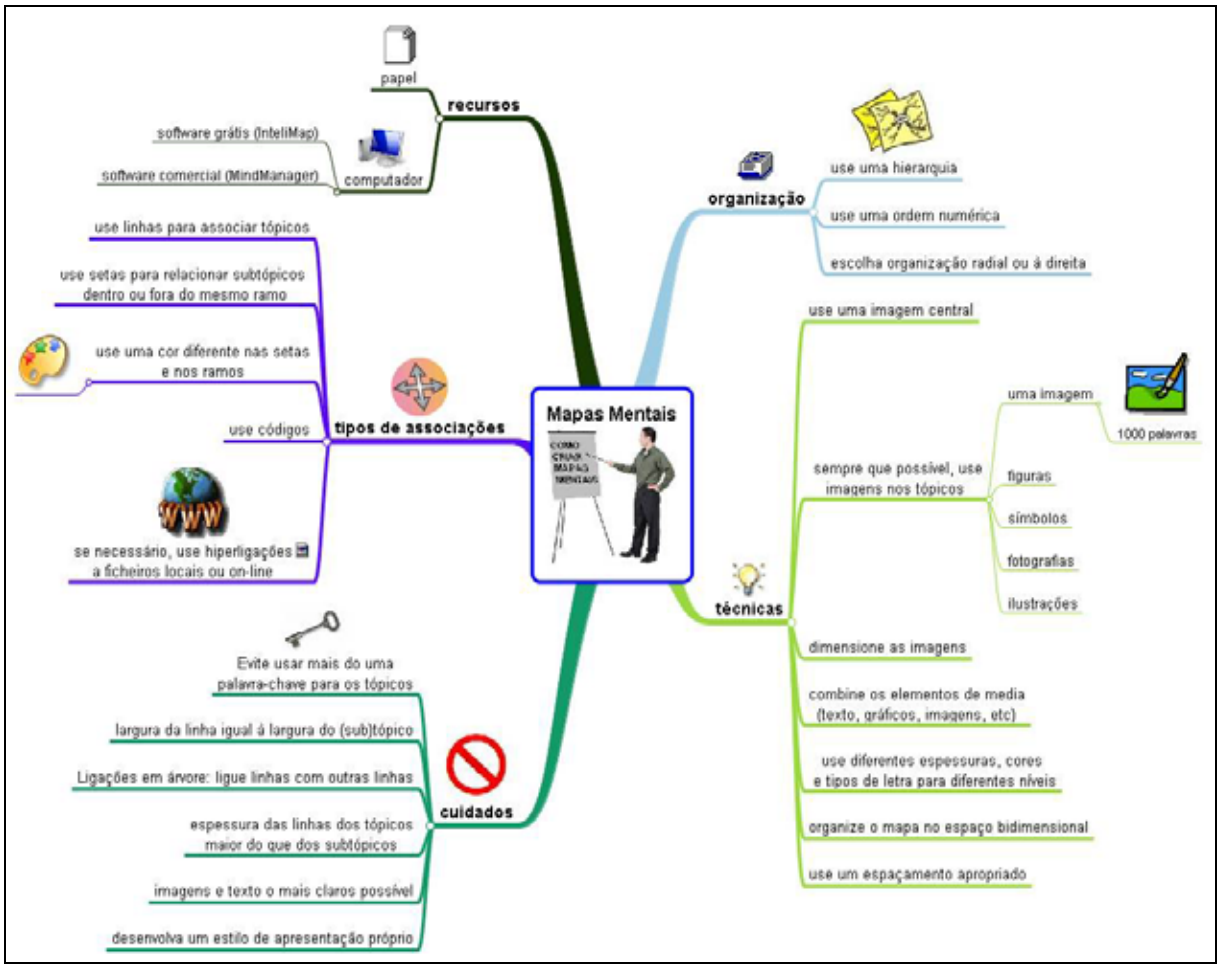


Figura 15 – Um Mapa Mental sobre "Mapas Mentais" editado com InteliMap

Existem algumas ferramentas para a criação de mapas mentais, das quais destacamos:

- Gratuitas: *InteliMap, FreeMind e MindMan Personal*;
- Comerciais: *MindManager*, *MindGenius*, *MindMapper* e *Visual Mind*.

Embora os mapas mentais sejam semelhantes aos mapas de conceitos, estes últimos atribuem uma menor ênfase ao tema principal e um maior foco nas relações entre os diferentes conceitos de um mesmo assunto num determinado domínio ou área do conhecimento.

2.5.3.2- Mapas de Conceitos

No âmbito das Ciências da Educação, os mapas de conceitos baseiam-se nas teorias construtivistas, mais concretamente na teoria cognitiva da aprendizagem significativa formulada por Ausubel (1968). Segundo esta teoria, a compreensão é diversa e depende do sujeito e da sua individualidade, uma vez que cada nova informação se relaciona com as estruturas de conhecimentos particulares de cada aluno. A estrutura de conhecimento ou estrutura cognitiva pode ser descrita como um conjunto de conceitos, organizados de forma hierárquica, que representam o conhecimento e as experiências adquiridas pelo sujeito.

Com base nesta teoria, Novak propôs os mapas de conceitos como um modelo para perceber como é que o conhecimento é armazenado na estrutura cognitiva de um indivíduo.

Genericamente, os mapas de conceitos são um recurso para a representação do conhecimento. São uma representação visual que relaciona conceitos hierarquicamente. Segundo o seu criador (Novak, 1977), constituem simultaneamente um instrumento para apoiar processos de auto-aprendizagem, uma estratégia de organização de materiais educativos e um modelo para localizar e elucidar o significado dos recursos, para além da sua aplicação no processo de avaliação dos alunos. Um mapa de conceitos (Novak, 2000):

- É um modelo hierárquico para conceitos, onde os conceitos mais gerais e inclusivos aparecem no topo e, por baixo destes, os que são cada vez mais específicos;
- É um conjunto de proposições (dois ou mais conceitos ligados através de uma relação que cria uma unidade semântica designam-se proposição);
- É formado por nós (conceitos) ligados através de arcos ou setas (relações);
- É um gráfico onde, normalmente, os conceitos são substantivos e as relações são verbos.

Com base na literatura existente, a noção e a aplicação dos mapas de conceitos podem explicar-se em três níveis de análise (Gaines et. al, 1995):

1) Numa perspectiva de representação abstracta constituída por uma rede de nós e arcos ou setas;
2) Numa perspectiva de visualização sob a forma de diagramas disponibilizam uma relação consistente entre as características visuais como sinais e a sua infra-estrutura semiótica;

3) Numa perspectiva discursiva como instrumento de representação e comunicação do conhecimento através de linguagens visuais para facilitar a interpretação por uma comunidade.

No contexto pedagógico, a articulação das perspectivas mencionadas permite que os mapas de conceitos sejam usados nas mais diversas situações do processo de ensino/aprendizagem: na planificação de currículos ou de unidades didácticas; na organização de conteúdos; na sintetização de informação; na simplificação de problemas complexos; no apoio à apresentação de uma lição, palestra, livro ou conteúdo e do seu autor; na integração de conteúdos de diferentes fontes de informação; na tradução dos conteúdos de aprendizagem sistematizados em conteúdos significativos para o aluno; na orientação da auto-aprendizagem ou do estudo; na representação do conhecimento adquirido pelo aluno; no processo de avaliação da aprendizagem; no apoio a actividades de exploração ou de pesquisa; na descrição gráfica de ideias, pensamentos ou reflexões, despertando a criatividade em sessões de brainstorming; etc.

Existem algumas ferramentas para a criação e navegação em mapas de conceitos, das quais destacamos:

- Gratuitas: *CmapTools, Compendium e Visual Understanding Environment* (VUE);
- Comerciais: *SMART Ideas, Semantica, Logotron* e *Axon Idea Processor*.

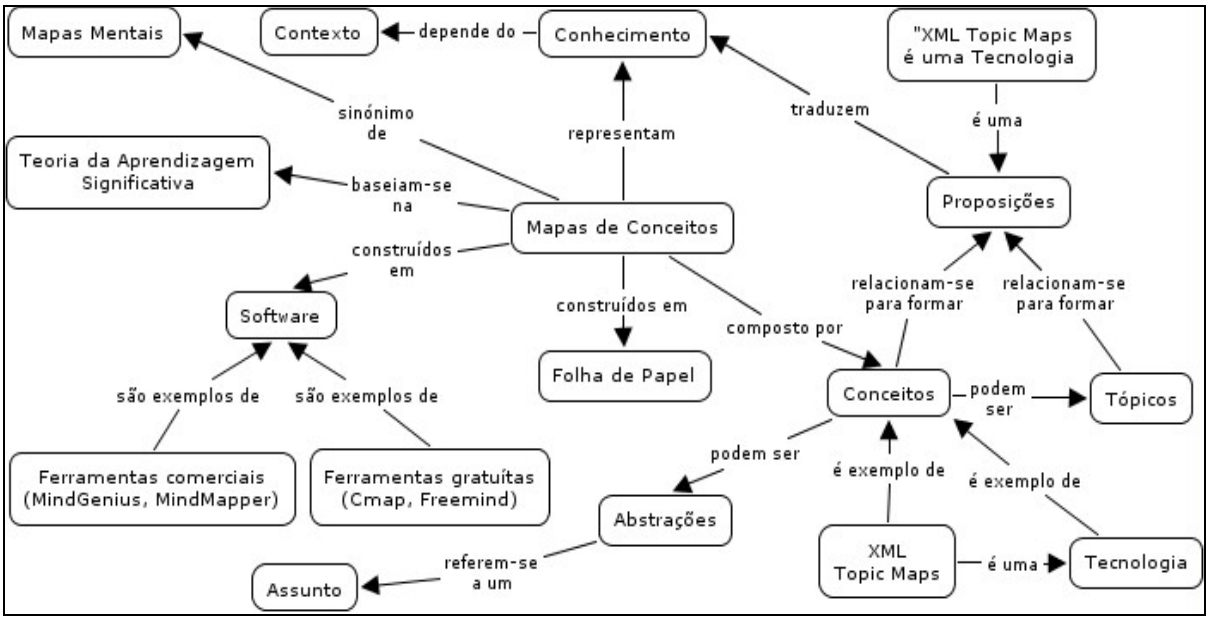


Figura 16 – Um Mapa de Conceitos sobre "Mapa de Conceitos" editado com CmapTools

Actualmente, a grande maioria dos objectos de aprendizagem ou outros recursos de informação existe no formato digital. É portanto desejável que esses recursos sejam ligados aos conceitos dos mapas que os ilustram.

Assim, acreditamos que o uso de Mapas de Conceitos para representar o conhecimento nos sistemas de e-Learning (ou em sistemas de gestão de conteúdos de aprendizagem) favoreceria não só o processo de aprendizagem (quer na planificação de cursos, quer na organização mental dos conceitos), mas também a organização e apresentação dos objectos de aprendizagem e, consequentemente, a localização, recuperação e reutilização dos mesmos.

Mais recentemente, no final da década de 90, surgiu uma abordagem análoga, no âmbito das Ciências da Informação, designada por mapas de tópicos (*topic maps*), para representar informação no âmbito da actividade de indexação das Ciências Documentais com vista a organizar e gerir recursos de informação, através da construção e integração de diferentes esquemas de índices e de glossários inerentes a recursos documentais (Park e Hunting, 2003).

2.5.3.3- Mapas de Tópicos

Os mapas de tópicos representam estruturas de conhecimento, ou seja, estruturas de recursos de informação usadas para definir tópicos e as correspondentes inter-relações. Com vista a normalizar a construção de mapas de tópicos e encontrar uma notação universal para os descrever, surgiram algumas propostas e linguagens, tendo a XTM (XML *Topic Maps*) (Pepper e Moore, 2001) sido escolhida como a norma *Topic Maps* ISO/IEC 13250 para representar os mapas de tópicos. A XTM é uma linguagem baseada em XML que permite a descrição, manipulação e intercâmbio de mapas de tópicos no âmbito da Internet (face às suas potencialidades para modelar um domínio do conhecimento, esta linguagem será detalhada mais adiante no âmbito das tecnologias para a Web Semântica, mais concretamente no capítulo IV).

Em suma, embora com finalidades diferentes, os mapas de conceitos assemelham-se aos mapas de tópicos, sendo possível a tradução de uns nos outros desde que a estrutura (conceitos, relações e *links* para os recursos) dos mapas conceptuais possam ser armazenados, por exemplo, em base de dados para, posteriormente, permitir a construção dos mapas de tópicos através de XTM (Paz et al., 2005). Os mapas de tópicos têm vindo a ser sugeridos como forma de representar o conhecimento nos Websites, logo podem constituir um excelente

instrumento para organizar e representar o conhecimento de um curso de e-Learning, fomentando não só a apresentação e a recuperação de objectos de informação, mas também a interoperabilidade entre e-cursos ou mesmo entre plataformas.

2.6- Resumo

Este capítulo teve como principal objectivo apresentar o estado da arte referente à utilização das TICs nos processos de aprendizagem e, mais concretamente, do e-Learning e do b-Learning enquanto modalidades de educação ou ensino a distância.

As principais iniciativas levadas a cabo nos últimos anos, para minimizar as barreiras tecnológicas, culturais e de formação no acesso à Sociedade da Informação portuguesa e para fomentar a aprendizagem ao longo da vida, têm vindo a criar as condições mínimas necessárias para que qualquer cidadão possa aceder a ambientes de aprendizagem *on-line*.

As recentes alterações na Lei de Bases do Sistema Educativo, grande parte delas impulsionadas pelo Processo de Bolonha, reconhecem formal e legalmente a importância do ensino a distância e abrem diversas oportunidades para o desenvolvimento e utilização do e-Learning em Portugal, quer ao nível do Ensino Superior, quer ao nível do Ensino Básico e Secundário.

Nesta última década, o e-Learning preocupou-se, essencialmente, com questões tecnológicas, espaciais e temporais, relegando as questões pedagógicas e de organização, estruturação e interoperabilidade dos conteúdos para segundo plano.

Reconhece-se que as tecnologias por si só não são suficientes para criar sistemas de e-Learning com cursos capazes de proporcionar níveis de interacção, de motivação, de aprendizagem e de aproveitamento aceitáveis, pelo que as teorias pedagógicas e os modelos de estruturação de conteúdos não devem ser descurados. Assim, os modelos de aprendizagem para a Sociedade da Informação, em geral, e para o e-Learning, em particular, têm a ganhar ao usar-se uma abordagem construtivista.

Reconhece-se também que o conteúdo é nuclear em praticamente todas as soluções de e-Learning, mas o seu intercâmbio, pesquisa, reutilização e personalização têm sido

menosprezados. Convenhamos que e-cursos compostos de pdfs ou outros documentos similares de baixa granularidade não constituem a melhor estratégia de aprendizagem.

Nesta perspectiva, o modelo SCORM e a filosofia dos objectos de aprendizagem têm-se assumido como a forma mais adequada de desenvolver, partilhar e reutilizar conteúdos de e-Learning.

Não obstante, o uso de objectos de aprendizagem é muitas vezes conotado com o acto de folhear um livro, levando os alunos a ter a sensação de que estão perante conteúdos estáticos e que, consequentemente, provocam a sua desmotivação. Para além disso, a aprendizagem pode ocorrer sem objectos de aprendizagem, uma vez que a aprendizagem pode ocorrer quando os alunos colaboram/cooperam para resolver problemas.

O *Learning Design* assume-se como a forma de minimizar estas limitações já que coloca ênfase na personalização dos conteúdos e promove uma participação mais activa do aluno no processo de aprendizagem, através da modelação de unidades de aprendizagem. Actualmente, a IMS-LD é a especificação que mais se destaca para descrever cenários de aprendizagem com várias actividades e serviços associados, permitindo a utilização de vários tipos de modelos pedagógicos.

A IMS-LD pode ser vista como uma tecnologia de integração e pode usar ou ser ampliada com várias outras especificações, incluindo a norma SCORM. Por conseguinte, as vantagens na adopção desta especificação são claras, mesmo para os fabricantes de plataformas de e-Learning que adoptaram o SCORM há pouco tempo.

Finalmente, o recurso a mapas mentais, mapas de conceitos ou mapas de tópicos pode constituir uma excelente alternativa para a organização, representação e sintetização do conhecimento nos sistemas de e-Learning.

3- Camadas Estrutural e Sintáctica da Web Semântica

A Web está prestes a tornar-se na maior base de dados do mundo pesquisável semanticamente não só por humanos, mas também por máquinas – uma verdadeira biblioteca ou mediateca universal onde a informação se encontrará devidamente organizada e poderá ser recuperada com base no seu significado. A arquitectura para a Web Semântica deverá fornecer a sintaxe para representar metadados e os vocabulários para exprimir esses metadados, ou seja, as ontologias (Harmelen, 2004). O presente capítulo incide essencialmente sobre as tecnologias para os metadados. As ontologias serão discutidas no capítulo seguinte.

A descrição da evolução da rede Internet, e em particular da Web, bem como as suas principais limitações constituem a primeira secção deste capítulo.

Uma das primeiras tarefas num novo campo de investigação é definir o fenómeno com alguma precisão (Garrison, 1989). Por conseguinte, a clarificação e contextualização dos conceitos referentes à iniciativa Web Semântica são organizadas na secção seguinte.

A terceira secção apresenta a arquitectura para a Web Semântica e sua evolução e as secções seguintes têm como objectivo analisar individualmente as tecnologias desta arquitectura para a camada estrutural e para a camada sintáctica: XML para a estruturação da informação com base na sintaxe definida nos esquemas XML e os esquemas de metadados e as metalinguagens para descrever e atribuir significado a essa informação.

3.1- Rumo à Web Semântica

A Internet é uma rede de redes de computadores à escala mundial. Constitui a infraestrutura de suporte das denominadas auto-estradas da informação.

O seu sucesso é, por um lado, fruto de características tecnológicas: a independência da plataforma, a instalação e a actualização relativamente económicas, a confiabilidade e a facilidade de utilização e manutenção inerentes ao modelo de navegação hipermédia, a simplicidade das linguagens e dos protocolos, a capacidade de introdução de novas tecnologias;

e, por outro, de factores políticos e de gestão: rede aberta e democrática cuja propriedade, gestão e responsabilidade se encontram distribuídas entre diferentes países, instituições, empresas e grupos de indivíduos (Silva, 1999). Com o crescimento exponencial da Internet, nomeadamente com o aparecimento da Web, passou a ser possível aceder a mais informação e a mais pessoas. Actualmente, a Web assume-se como a melhor infra-estrutura para a comunicação, publicação, partilha ou disseminação de informação (conteúdos). Estes factores favoreceram o aparecimento de inúmeros sítios de informação (Websites), desde as páginas Web pessoais até aos mais complexos Sistemas de Comércio Electrónico e de Ensino a Distância.

3.1.1- Evolução da Web

A Web pode ser vista como uma rede de servidores que veio tornar a Internet mais divertida e amigável. Estes servidores proporcionam o acesso a um conjunto de documentos ou páginas ligadas (através de hiperligações) que podem conter texto, imagens, gráficos, som, animação ou vídeo.

Este serviço Internet, desenvolvido inicialmente no CERN (*Conseil Européen pour la Recherche Nucléaire*) em 1989, tornou-se popular em 1993 com o lançamento do *browser* Mosaic (Cliente da NCSA - *National Center for Supercomputing Applications*) e a partir desse momento a utilização da Internet cresceu exponencialmente. O aparecimento do primeiro *browser* (navegador ou visualizador) impulsionou o desenvolvimento da HTML (*HyperText Markup Language*). Em Outubro de 1994, foi criado o W3C – *World Wide Web Consortium*, um organismo responsável pela promoção e acreditação de normas relacionadas com este serviço, com o objectivo último de maximizar as potencialidades da Web.

Quando a Web se assumiu como um dos serviços disponibilizados pela rede Internet, os conteúdos distribuídos correspondiam a programas, dados ou informações do tipo textual. Os sistemas de hipertexto, que entretanto surgiram, revolucionaram a forma de comunicar. O texto foi, gradualmente, complementado por outros media (áudio, animação e vídeo), novas formas de os relacionar entre si (hiperligações) e connosco (interactividade), impulsionando a migração dos sistemas hipertextuais para sistemas hipermédia (Gonçalves, 2002).

O serviço Web baseia-se no protocolo HTTP (*HyperText Transfer Protocol*) que permite transportar documentos hipermédia (essencialmente páginas em formato HTML). As

páginas HTML são criadas manualmente através de editores de HTML e editores de texto ou através de ferramentas específicas de criação de páginas Web (por exemplo, *Microsoft FrontPage* ou *Macromedia DreamWeaver*, entre muitos outros). Estas páginas HTML designam-se páginas estáticas, uma vez que são criadas uma única vez. Por oposição, as páginas HTML geradas várias vezes, automaticamente e em tempo real por programas específicos (conversores ou *gateways*) designam-se páginas dinâmicas. A vantagem das páginas dinâmicas em relação às estáticas resulta essencialmente do acesso interactivo a Bases de Dados. Por conseguinte, a Web é um sistema mundial de hipermédia distribuído, baseado no modelo cliente-servidor.

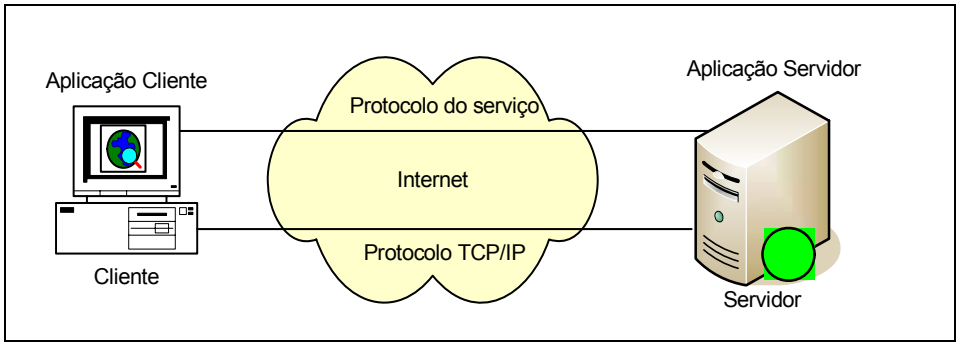


Figura 17 – Modelo Cliente-Servidor na Internet

A partir de 1995, o Java assumiu-se como a linguagem de construção de aplicações interactivas para a Web, favorecendo a interacção homem-máquina através de applets carregadas a partir de um servidor Web remoto e executadas por uma máquina virtual nos clientes Web. Estas aplicações impulsionaram a aplicação da filosofia da Web no seio das organizações para a partilha de informação, entre outros recursos, através de Intranets e Extranets. Uma Intranet consiste num ambiente computacional interno de uma organização (rede local), ou eventualmente de um grupo restrito de organizações, que utiliza a tecnologia existente na Internet. O *browser* é usado como interface para aceder a informações organizacionais que não estão disponíveis para o público em geral. Podemos caracterizar uma Intranet pelo fluxo de informações institucionais através de documentos digitais utilizando a interface Web. Uma Extranet é uma Intranet adoptada por empresas distribuídas geograficamente que não dispõem de redes próprias e que utilizam a Internet como meio de suporte comunicacional dos seus sistemas internos (Silva, 1999), permitindo acesso seguro a utilizadores ou empresas externas.

A Internet, e em particular a Web, constitui um macro-sistema de dados que circulam de um lado para o outro, as 24 horas do dia, os 7 dias da semana, os 365 dias do ano.

3.1.2- Principais Limitações da Web

A Web tornou-se numa biblioteca de documentos à escala mundial, constituindo actualmente o maior repositório de informação que disponibiliza conteúdos em páginas estáticas ou dinâmicas e oferece serviços para praticamente todos os sectores da sociedade.

Face à proliferação de conteúdos na Web, surgiram mecanismos de pesquisa, cuja principal função é facilitar a exploração e recuperação de informação. Os mecanismos de pesquisa podem classificar-se em:

a) *Directories* (listas, catálogos ou directórios, tais como: Yahoo! e Lycos), onde a informação se encontra catalogada de acordo com as descrições das páginas fornecidas por quem as submete;

b) *Search engines* (motores de pesquisa ou motores de busca de informação, tais como: Google e Altavista), que baseiam a busca de informação em palavras-chave, em *robots* que realizam o *download* das páginas Web, em indexadores que criam índices extraindo automaticamente os termos-chave das páginas ou em interfaces de consulta que comparam os termos recebidos com a base de termos indexados (Ramalho, 2002; Sullivan, 2002a).

Contudo, o constante aumento do número de servidores e de páginas Web faz com que a teia de informação assuma dimensões gigantescas, tornando difícil encontrar aquilo que se deseja ou necessita, mesmo recorrendo aos agentes dos motores de busca, pois o número de resultados devolvidos é exageradamente grande e impreciso. As bases de dados de alguns motores de pesquisa são construídas automaticamente com um tipo de programas ou agentes de software denominados rastejadores (*crawlers*) que analisam periodicamente a informação disponibilizada através de vários serviços da Internet e criam registos sobre o seu conteúdo. Nesta perspectiva, sem constituir uma solução mágica, as *meta tags* e o texto ALT da *tag image* têm sido úteis para apoiar os agentes de pesquisa. As *meta tags* de título (*title meta tag*), de classificação (*classification meta tag*), de descrição (*description meta tag*) e de palavras-chave (*keywords meta tag*) são inseridas no cabeçalho das páginas Web, sendo invisíveis para os utilizadores, mas bastante úteis para os *crawlers*, uma vez que incluem metadados. No entanto, como nem sempre as *meta tags* traduzem a realidade do conteúdo dos Websites, os motores de busca passaram a preocupar-se mais com a indexação do texto disponível no corpo da página (Sullivan, 2002b).

Uma outra forma de auxiliar os utilizadores a encontrarem informação sobre determinado assunto foi o aparecimento de portais generalistas e temáticos, munidos de directório e motor de pesquisa. Mas, nem mesmo os portais temáticos, que se tornaram ferramentas importantes para os cibernautas, conseguem lidar facilmente com a grande acumulação de informação.

Actualmente, os motores de busca, entre outras empresas com presença na Web, disponibilizam mecanismos de pesquisa personalizados para cada Website que, após a indexação dos conteúdos internos, facilitam a procura de informação no Website em questão, através de palavras-chave.

Contudo, à medida que cresce o volume de informação e as necessidades de actualização se repetem, os Websites vêem-se obrigados a recorrer a Bases de Dados. As Bases de Dados constituem uma das principais formas de armazenamento de grandes quantidades de informação de forma estruturada, de modo a que a localização de informação armazenada possa ser feita de forma rápida e eficiente. Para tal, esses Websites disponibilizam mecanismos, criados com recurso a linguagens como ASP, PHP ou JSP, com vista a facilitar a recuperação de informação dos conteúdos internos (informação armazenada na base de dados do Website), através de palavras-chave.

No entanto, os motores de busca externos ao Website não conseguem ter acesso aos conteúdos gerados a partir de uma base de dados, uma vez que não têm forma de perceber como os dados se encontram organizados, sendo difícil para um agente externo ao Website realizar inferências sobre as Bases de Dados.

Na perspectiva desse tipo de motores de busca externos ao Website, a grande maioria das Base de Dados, embora armazenem muita informação, é informação que é em grande parte inútil, porque está inacessível, profunda ou invisível. Tal como referido anteriormente, esta parte da Web tem vindo a ser designada de Web profunda ou Web invisível, uma vez que as páginas não estão armazenadas em nenhum local, são geradas dinamicamente apenas quando o utilizador as solicita directamente interagindo com a Base de Dados por intermédio de uma aplicação ou página específica. Vários autores referem que a Web possui algumas centenas de vezes mais informação do que aquela que pode ser acedida usando os actuais motores de busca (Singh, 2002; Bergman, 2001; Sherman, 2001).

Segundo um estudo sobre a Web profunda, intitulado “The Deep Web: Surfacing Hidden Value” (Bergman, 2001), existem centenas de milhares de milhões de páginas Web

escondidas em bases de dados a que os motores de busca convencionais não conseguem aceder, pois os *crawlers* não conseguem ir além das páginas que encontram e seguir os links nelas inseridos, pois essas Bases de Dados só produzem resultados se questionadas directamente.

Em suma, existe uma panóplia de sítios na Internet recheados de informação útil que dificilmente conseguimos localizar. Por um lado, porque parte dessa informação se encontra algures perdida no fundo do mar de informação onde só os humanos a conseguem entender (para as máquinas essa informação passa despercebida, é difícil de localizar ou, simplesmente, não a conseguem interpretar), e por outro, porque poderá estar armazenada em bases de dados, sendo complicado aceder a ela através dos motores de busca actuais. A constante proliferação de conteúdos na Web tornou cada vez mais difícil encontrar o que quer que seja, mesmo recorrendo a catálogos ou motores de busca, pois o número de resultados devolvidos é exagerado e impreciso. Isto acontece porque os resultados devolvidos correspondem a centenas ou mesmo milhares de páginas Web que contêm palavras iguais às palavras-chave submetidas, não havendo preocupação com a semântica ou significado dessas palavras. Por isso, não é de estranhar que 71% dos 561 inquiridos no estudo publicado pela *Roper Starch Worldwide* em Dezembro de 2000 ficassem frustrados com os resultados obtidos aquando da realização de pesquisas. Embora existam estudos bem mais recentes e animadores, a grande maioria aponta para percentagens entre os 55% e os 74%. Este estudo indicava também que, em média, os internautas começavam a demonstrar indícios de nervosismo após 12 minutos de pesquisa *on-line* improdutiva, pelo que 86% dos inquiridos achava que deveria ser encontrado um método mais eficiente de filtragem durante as pesquisas na Web (TEK, 2000).

Vários exemplos simples ilustram esta realidade. Imaginemos que pretendíamos descobrir “cursos de formação sobre informática” através de um dos motores de busca mais populares da actualidade: Google. Submetida a pesquisa usando as aspas (é pesquisada a frase exacta), curiosamente não nos devolveria qualquer resultado. Sem as aspas (todas as palavras-chave são pesquisadas), obteríamos cerca de 1.250.000 documentos Web. Se adicionássemos um parâmetro que incluísse algum nível de semântica, como por exemplo “em Dezembro de 2006” como critério de filtragem, esse número baixaria para cerca de cerca de 867.000 documentos Web. Se incluíssemos mais parâmetros para filtrar a pesquisa através do formulário de pesquisa avançada (país, localidade, data de actualização das páginas, língua, formato do documento, tipo de curso, etc), obviamente que conseguiríamos reduzir o número de páginas Web devolvidas. Porém, continuaríamos a obter um número excessivo de páginas

(cerca 350 páginas Web para a pesquisa ilustrada na Figura 18) e a grande maioria delas não corresponde minimamente ao que pretendemos.

Figura 18 – Uma pesquisa de cursos de formação no Google

Mas, para conseguirmos obter uma resposta à nossa questão ainda teríamos que interpretar o conteúdo de cada uma dessas páginas. Portanto, uma pesquisa baseada apenas na sintaxe, deixando a avaliação da semântica para o utilizador, não é a melhor estratégia para orientar as pesquisas numa Web que cresce exponencialmente.

Embora os resultados não sejam famosos, os motores de busca estão indiscutivelmente na ordem do dia. Tal como refere João Paulo Luz (director comercial do SAPO), o acesso a páginas de pesquisa ultrapassou em muito os acessos a outros sites (TEK, 2006). A Microsoft reconhece o potencial desta área e está a trabalhar para ser líder nas pesquisas, enfrentando os actuais líderes das tecnologias de pesquisa: Google e Yahoo!, tal como refere o seu CEO Steve Ballmer (Gutiérrez, 2006).

A Web deixou de ser apenas uma colecção de páginas para se transformar numa infra-estrutura que disponibiliza não só o acesso a páginas estáticas, como também a aplicações e serviços que lidam com a crescente abundância de informação. No entanto, embora a eficácia (atingir os objectivos) e a eficiência (atingir os objectivos optimizando os recursos) dessas aplicações num contexto de aplicação específico (organizações) sejam bastante satisfatórias, o mesmo já não se poderá dizer quando aplicadas a contextos mais gerais ou, em última instância, a toda a Web, uma vez que, embora possam ser eficazes, não conseguem ser eficientes.

Em suma, a evolução da Internet e, mais concretamente, da Web pode resumir-se em três gerações consequentes e cumulativas. Enquanto que a primeira geração da rede permitia apenas a troca de dados entre máquinas distintas sendo a Web resultado da criação manual de páginas em HTML (páginas estáticas), a segunda provocou uma revolução ao disponibilizar uma vasta gama de aplicações e informação para as pessoas (preocupação na apresentação de informação recorrendo a editores e utilizando modelos e folhas de estilo, geração automática de páginas Web baseadas em base de dados - páginas dinâmicas e interactivas - e interacção com o utilizador através de formulários e de mecanismos de busca), tornando também possível o ensino/aprendizagem e o comércio electrónicos. A próxima geração da rede tem vindo a ser designada Web Semântica (*Semantic Web*) e aposta na estruturação dos recursos de informação, na descrição dos mesmos através de metadados e no recurso a agentes inteligentes orientados por ontologias para eliminar, ou pelo menos atenuar, as limitações dos motores de busca e para apoiar o utilizador na realização de tarefas mais sofisticadas que as executadas actualmente.

As aplicações para a Web Semântica permitirão processar, reutilizar e partilhar a informação inteligentemente, com ou sem ajuda humana. A Web transformar-se-á assim numa enorme base de conhecimento compartilhado, constantemente lida e escrita não só por humanos, mas também por aplicações semânticas.

Diversos autores defendem que existe uma grande similaridade entre o cérebro humano e a Internet. Se todos podemos estar ligados à Internet, poderá esta assumir-se como o cérebro do nosso planeta? E o seu comportamento colectivo será inteligente, podendo constituir uma mente global? Poderá esta forma colectiva de inteligência comparar-se à de uma mente humana individual, ou de um grupo de mentes humanas?

A Internet pode não estar organizada exactamente da mesma forma que o cérebro humano, mas podemos dizer que é uma extensão dele. McLuhan afirmou que as tecnologias podiam ser vistas como extensões dos nossos próprios membros e sentidos. Enquanto extensões de nós próprios, qualquer tecnologia exterioriza, amplifica e modifica diversas funções cognitivas (McLuhan, 1964-1969). Consequentemente, podemos afirmar que a Web poderá vir a transformar-se numa extensão dos nossos sentidos e das nossas mentes, um cérebro global da sociedade em que vivemos. Tal como o cérebro inclui um sistema de memória onde pode armazenar, localizar e recordar informações, também a Internet disponibiliza espaços e protocolos para armazenar, partilhar e recuperar informações, favorecendo a cognição colaborativa de forma descentralizada. Este espaço de partilha de informação é como a memória colectiva da mente global.

Podemos, então, afirmar que a Internet (tecnologia/hardware) tem vindo a evoluir no sentido de um cérebro global distribuído e a sua actividade (agentes de software, seres humanos e dados) representa o processo cognitivo de uma mente global cada vez mais inteligente. Tal como o cérebro humano é composto por milhares de neurónios, também a Internet é uma vasta rede computacional distribuída composta por biliões de processadores em interacção paralela. Nestes processadores podemos incluir os programas de software, os próprios seres humanos e sistemas compostos de ambos, tais como as organizações (as instituições educativas como organizações do saber por excelência são um exemplo evidente disso mesmo). Tal como o poder de raciocínio do cérebro é maior do que o poder da soma das suas partes, também a Internet no seu todo é mais do que a soma de suas partes, beneficiando das características e potencialidades de uma rede de redes. O poder do sistema cresce de forma exponencial à medida que as partes e as relações ou ligações entre elas são adicionadas.

Esta abordagem baseia-se na possibilidade dos servidores poderem ser capazes de aprender o percurso que os cibernautas fazem com vista a que a rede possa funcionar como um cérebro global (Heylighen et al., 2001). Genericamente, esta ideia assume que as hiperligações na Web podem ser vistas como as associações no cérebro humano, ou seja,

como as sinapses que interligam os neurónios. Mais informação sobre esta visão pode ser encontrada no anexo B.

3.2- A Web Semântica

A Web foi criada com a visão de que seria um espaço onde a informação teria um significado bem definido, facilitando a cooperação e a comunicação entre as pessoas e os agentes de software (Berners-Lee et al., 2001). Passada mais de uma década do seu lançamento, essa visão ainda não foi alcançada. A Web continua organizada na perspectiva dos humanos (linguagem natural e ênfase na apresentação), o que torna a busca de informação uma tarefa árdua. Normalmente, os computadores preocupam-se com a apresentação da informação contida nas páginas Web, sendo incapazes de a entenderem sintáctica, semântica e morfologicamente. Para que a Web contemple a perspectiva das máquinas, há que encontrar soluções de estruturação, integração, troca e compreensão semântica da informação tanto na óptica dos humanos, como na óptica das máquinas através dos seus agentes de software. É neste contexto que surge a denominada Web Semântica (W3C, 2001) como um cenário desejável para a Web.

3.2.1- As Mudanças da Web Actual para a Web Semântica

Em primeira instância, a Web Semântica é uma iniciativa ou projecto, liderado pelo W3C, que pretende criar um meio universal para troca de informação, atribuindo significado (semântica) ao conteúdo dos documentos da Web, de modo a que esse significado seja compreendido não só por humanos, mas também por máquinas. Em última instância, a Web Semântica é uma extensão da Web actual que permitirá encontrar, partilhar e (re)combinar informação mais facilmente. Não se trata de uma revolução, mas sim de uma evolução.

Os objectivos principais desta iniciativa são (Miller, 2004):

- Construir uma estrutura comum de representação dos dados de modo a facilitar a integração de diversas fontes de informação para extrair novo conhecimento (migração de uma Web de documentos para uma Web de dados);
- Aumentar a utilidade da informação conectando-a aos seus conceitos e ao seu contexto;
- Descoberta e análise mais eficientes da informação.

A Figura 19, baseada na apresentação *Weaving Meaning: An Overview of The Semantic Web* de Eric Miller (2004), ilustra as diferenças entre a Web actual e a Web Semântica.

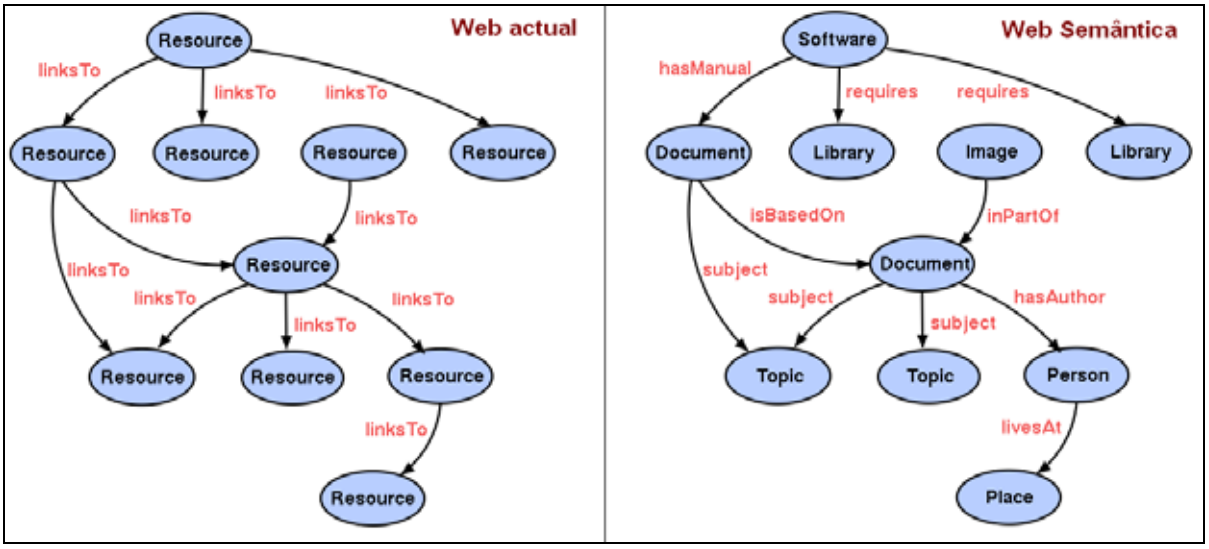


Figura 19 – Web Semântica como extensão da Web actual

A Web actual baseia-se essencialmente em links entre recursos, mas a sua semântica ou significado apenas pode ser capturada dentro do contexto e na perspectiva dos humanos. Ou seja, o utilizador lê e interpreta o significado do documento que encontra para decidir se corresponde ao que procura.

Com a Web Semântica passamos a tirar maior proveito dessas hiperligações, uma vez que os links passam a fornecer as relações entre os recursos de informação, permitindo que as máquinas tenham acesso a mais camadas de informação e, consequentemente, possam trabalhar e partilhar conhecimento com os humanos.

A tabela 1 resume as mudanças da Web actual ou tradicional para a Web Semântica (Raymond, 2001):

Característica	Web tradicional	Web Semântica
Destinatários	Seres humanos	Seres humanos e computadores (agentes de software)
Linguagem principal	HTML (*HyperText Markup Language*)	XML (*eXtensible Markup Language*)
Estrutura dos conteúdos	Documentos não estruturados	Documentos estruturados
Exibição dos conteúdos	Dados e sua apresentação no mesmo documento	Dados correspondem a um documento XML e apresentação corresponde a documentos XSL de acordo com as interfaces
Edição dos conteúdos	Documentos essencialmente estáticos	Documentos dinâmicos
Interactividade	Difusão unidireccional	Difusão bidireccional (Web editável)
Semântica dos conteúdos	Semântica implícita	Semântica explícita (metadados e ontologias)
Decomposição e recomposição	Websites independentes	Agregação num Website dinâmico de conteúdos de diversos Websites
Publicação	Centralizada (Servidor Web)	Descentralizada (P2P ou *Peer-to-Peer*)

Tabela 2 – Caracterização da Web

A missão do projecto Web Semântica pode traduzir-se na condução da Web rumo ao seu verdadeiro potencial (Miller, 2004). Neste contexto, os metadados e as ontologias, as metalinguagens e as ferramentas de anotação e criação de ontologias, os agentes inteligentes e sistemas de agentes móveis, entre outros avanços tecnológicos das Ciências da Computação e da Inteligência Artificial no âmbito da Gestão da Informação e do Conhecimento e dos Sistemas Distribuídos para a Web, constituem elementos essenciais para o desenvolvimento de soluções que a pouco e pouco contribuirão para a realidade da Web Semântica (WS).

A anotação de documentos HTML através das *meta tags* (<META></META>), ou mesmo o recurso simples a especificações de metadados, não é suficiente para que os motores de busca e outras aplicações disponham da semântica necessária para realizar inferências que permitam recuperar informação útil e relevante ou realizar outras operações mais avançadas.

Segundo a visão da WS, um qualquer utilizador, quer navegando num catálogo ou directório, quer utilizando um mecanismo de pesquisa, perante a indicação do assunto que pretende procurar, seria auxiliado por agentes inteligentes que, através de mecanismos e linguagens de inferência e orientados por ontologias articuladas, percorreriam páginas Web anotadas e repositórios de metadados com vista a recuperar os documentos ou recursos de informação sobre o assunto em questão (em vez de todos os documentos que contenham as palavras-chave indicadas). Ou seja, já não será a sintaxe a determinar o resultado do processo de recuperação de informação, mas sim a semântica.

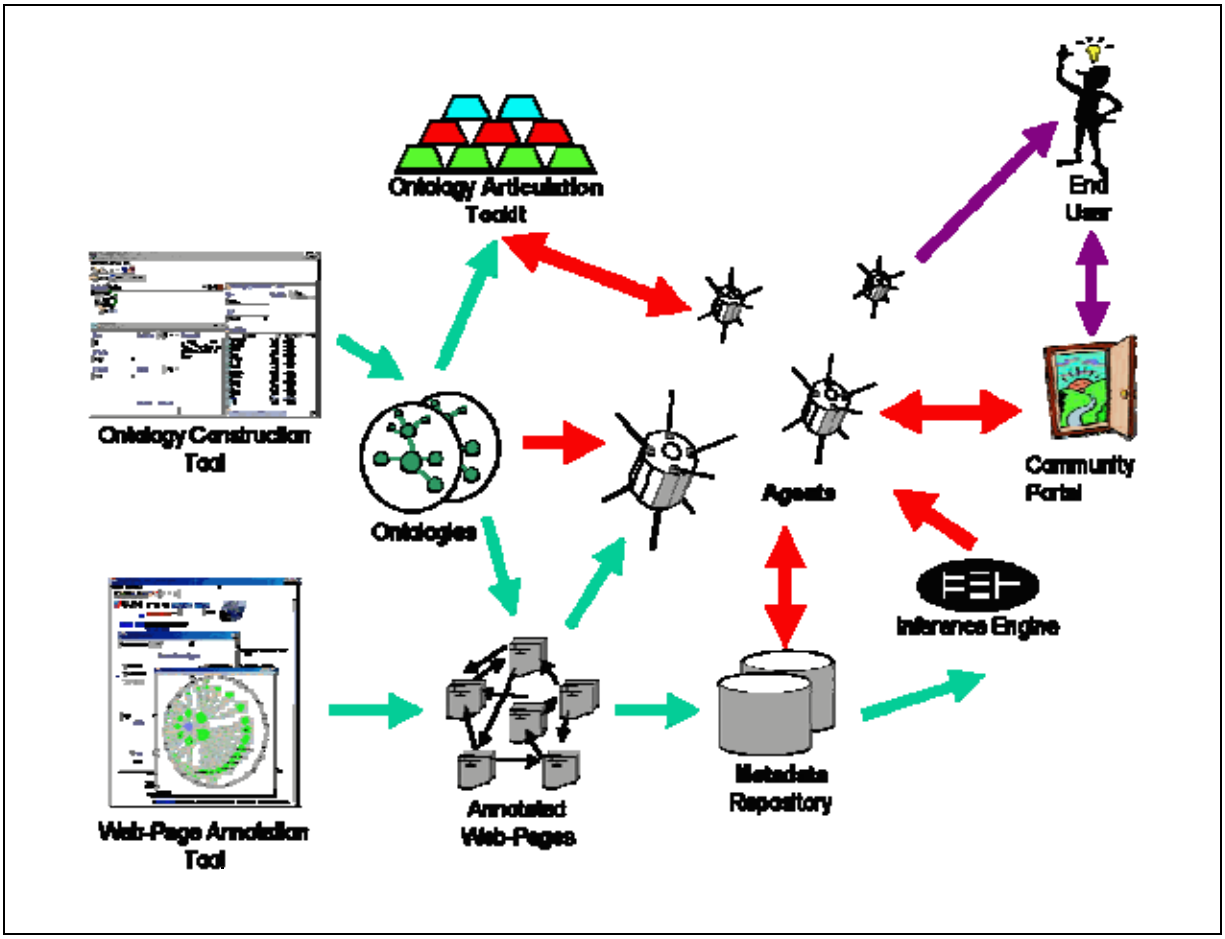


Figura 20 – Visão geral da Web Semântica

3.2.2- Arquitectura para a Web Semântica

A ideia da WS pode resumir-se à seguinte questão: como fazer com que os computadores entendam o conteúdo da Web? O primeiro passo será organizar e estruturar as

informações e o segundo será adicionar semântica às informações da Web, de tal forma que os agentes de software possam compreendê-las (Goñi et al., 2002).

A estruturação da informação para a WS assenta no uso intensivo de metadados, o que permitirá o acesso automático à informação por parte das máquinas, através do processamento semântico dos dados e de heurísticas (Berners-Lee et al., 2001). Com a evolução de uma Web baseada em documentos para uma Web baseada em dados, onde a informação é organizada em dados estruturados, passa a ser possível reutilizar dados ao nível aplicacional, organizacional ou entre os limites das organizações/comunidades, realizar pesquisas mais eficientes através de agentes de software, entre outras tarefas mais avançadas com vista a orientar, aconselhar e assistir o utilizador de acordo com os seus requisitos.

Em 2000, o W3C divulgou a primeira arquitectura para a Web Semântica baseada em camadas sobrepostas, em que cada camada ou tecnologia deveria obrigatoriamente ser complementar e compatível com as camadas inferiores, sem que tivesse de estar dependente de qualquer camada superior, tal como se pode verificar na Figura 21.

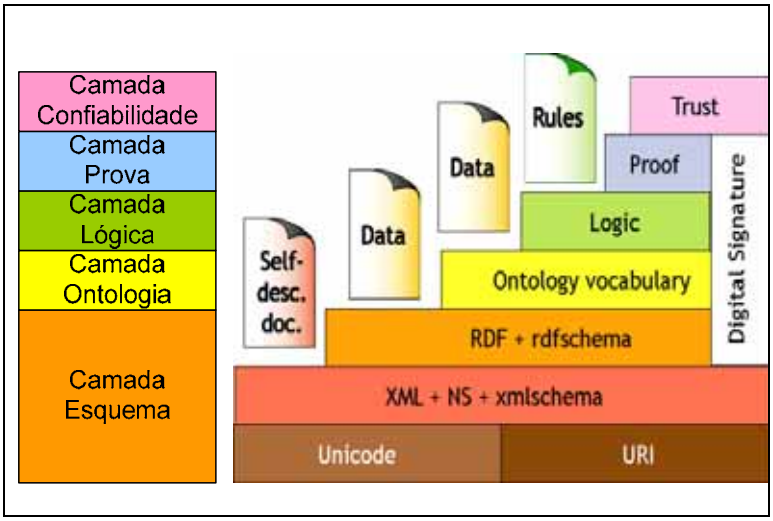


Figura 21 – Arquitectura para a Web Semântica (versão 2000)

As camadas mais baixas desta arquitectura ou infra-estrutura são cada vez mais usadas nos sistemas de informação para Web e podem ser agrupadas sob a designação de Camada Esquema (*Schema Layer*). Nesta proposta de arquitectura foram sugeridas apenas as tecnologias recomendadas para a camada esquema, resultantes dos estudos sobre as tecnologias XML e RDF (Miller, 1998; Berners-Lee et al., 1998; Lassila e Swick, 1999; Brickley e Guha, 2000).

O *Unicode* (Unicode, 2000) e o URI (Berners-Lee et al., 1998) constituem a base para a legibilidade e o endereçamento da WS, respectivamente.

A XML (Bray et al., 2000), o NS ou *namespace* (Bray et al., 1999) e o XMLS ou *xmlschema* (Fallside, 2001; Brown et al., 2001) formam a sub-camada sintáctica. A XML permite representar a estrutura dos dados e fornece um formato para intercâmbio de informação. Os XML *namespaces* são conjuntos de nomes, identificados por URIs, usados em documentos XML como nomes de atributos e tipos de elementos. Com eles podemos, por exemplo, identificar os vários significados que uma palavra pode ter. O XML *Schema* permite descrever que *tags* (rótulos ou etiquetas) são usadas e como é que são usadas (sintaxe).

A RDF (Lassila e Swick, 1999) e o RDFS ou *rdfschema* (Brickley e Guha, 2004) formam a sub-camada semântica. A RDF fornece uma forma de descrever recursos distribuídos, integrando os dados estruturados a partir de múltiplas fontes. A RDF descreve a informação de forma não ambígua, mas nada diz acerca das relações entre conceitos, o que limita significativamente a inferência. O RDFS fornece o vocabulário que permite que a informação seja vista como classes de instâncias, permitindo inferências sobre as mesmas.

Genericamente, a camada esquema é responsável por estruturar os dados e definir o seu significado para que se possa elaborar um raciocínio lógico. Para além da definição de que todos os documentos, objectos e eventos são recursos identificados por URIs (*Uniform Resource Identifiers*) e de que a sua leitura universal é garantida pela norma Unicode, XML(S) e RDF(S) constituem os principais elementos-chave desta camada no que diz respeito à estruturação e atribuição de significado aos dados. No âmbito desta tese usámos os termos XML(S) e RDF(S) para frisar que nos estamos a referir à tecnologia XML e ao seu esquema XML *Schema* e à tecnologia RDF e ao seu esquema RDF *Schema*, respectivamente.

A RDF e o RDF *Schema* fornecem um conjunto de primitivas que permite criar o vocabulário ontológico (*ontology vocabulary*) para representar o conhecimento de determinado domínio. A Camada Ontológica ou Camada Ontologia (*Ontology Layer*) é portanto constituída pelo vocabulário ontológico que fornece o significado dos termos e das teorias de determinada área do conhecimento compartilhados por todos os participantes do domínio, incrementando a expressividade semântica fornecida pela RDF e o seu esquema RDFS. Os modelos de raciocínio com dados semânticos são executados na Camada Lógica (*Logic Layer*). As camadas superiores da estrutura fornecem a troca das provas para reconhecer a verdade das inferências obtidas.

Os agentes devem ter capacidade para: compreender o significado e a relação entre os objectos, baseando-se no vocabulário ontológico; raciocinar sobre eles, utilizando as regras de inferência definidas na camada lógica; e ainda devem ser capazes de trocar dados, resultados do processamento de dados e provas de inferência (*proofs*) de forma eficiente e eficaz, uma vez que são eles que representam valor para os utilizadores da WS.

Finalmente, para que a Web seja de confiança e garanta a idoneidade da informação, deverá corresponder a um conjunto de documentos Web assinados digitalmente com determinadas chaves e conter indicações sobre essas chaves (Fensel et al., 2002).

Em 2002, o W3C apresentou algumas mudanças que, apesar de pouco significativas, permitiram clarificar a arquitectura para a Web Semântica (Berners-Lee, 2002). As alterações resumiram-se ao agrupamento de algumas tecnologias (as especificações XML e *xmlschema* foram agrupadas na camada XML que passou a incluir o XML *Schema*) e à subdivisão de outras (a camada RDF+*rdfschema* foi subdividida na camada RDF Modelo & Sintaxe e na camada RDF *Schema*, enquanto que o vocabulário ontológico foi subdividido em duas camadas: a camada ontologia e a camada regras). Foi também acrescentada transversalmente a criptografia para garantir, juntamente com a assinatura digital, a confidencialidade e a confiabilidade das informações na Web Semântica.

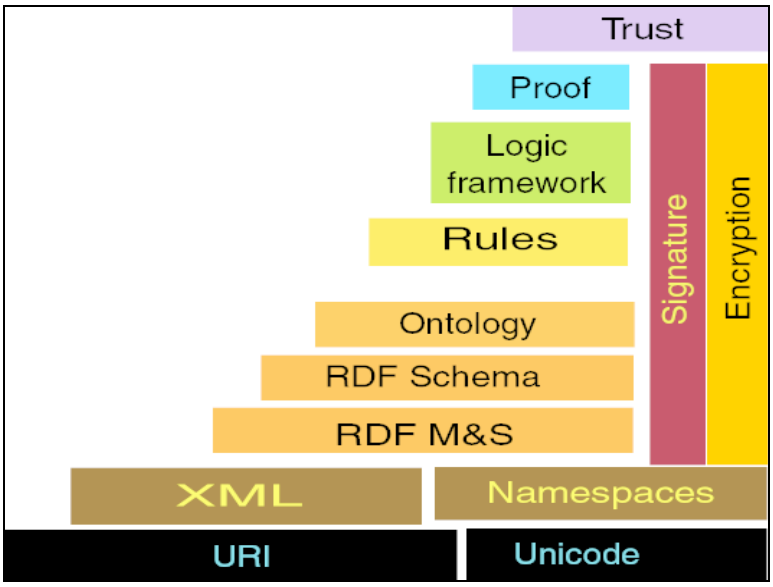


Figura 22 – Arquitectura para a Web Semântica (versão 2002)

A partir desta versão da arquitectura, os esforços concentraram-se essencialmente na implementação de linguagens e ferramentas para o desenvolvimento de ontologias. Consequentemente, em 2004, principalmente como resultado do grupo de trabalho *Web*

Ontology Working Group, foi publicada a *Web Ontology Language* (OWL) como linguagem recomendada pelo W3C para o desenvolvimento de ontologias.

Algumas dificuldades de integração e de compatibilidade da camada ontologia com as restantes camadas da arquitectura motivaram novas alterações à arquitectura para a Web Semântica. A Figura 23 corresponde à versão mais recente dessa arquitectura.

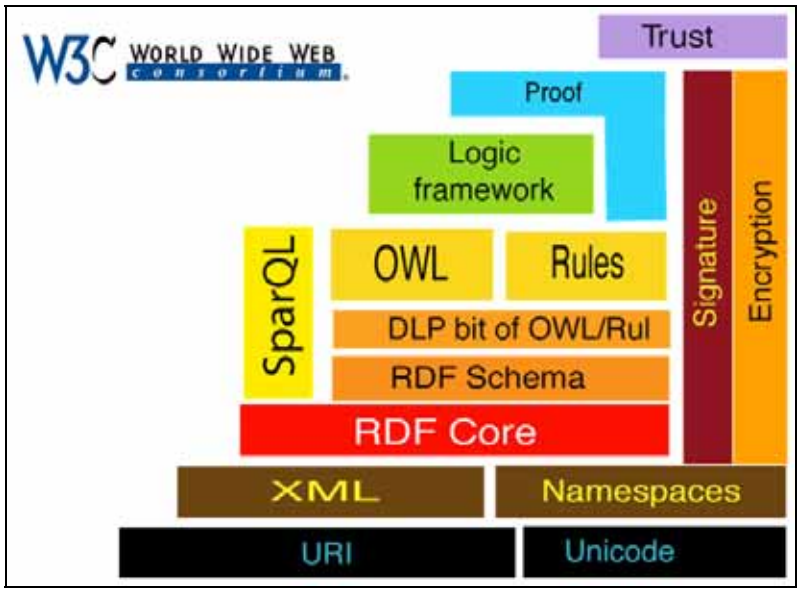


Figura 23 – Arquitectura para a Web Semântica (versão 2005)

Uma das últimas alterações diz respeito à introdução de uma tecnologia candidata a recomendação na camada DLP (*Description Logic Programs*). Esta tecnologia baseia-se na Lógica Descritiva (OWL DL) e na Programação Lógica (*F-Logic*) e tem como objectivo integrar as duas principais tecnologias para a WS (RDF e OWL), proporcionando uma estrutura de conhecimento mais flexível (Calvanese et al., 2006). Outra dessas alterações refere-se à camada de inferência SPARQL (*RDF Query Language and Protocol*) que permite realizar consultas sobre documentos RDF(S) ou OWL.

As principais tecnologias descritas sucintamente acima e ilustradas na Figura 23 constituem um dos principais pilares desta investigação e serão detalhadas na próxima secção e no capítulo seguinte. Com o intuito de compreender as funções e as características fundamentais das tecnologias, optámos por subdivir a arquitectura para a Web Semântica em quatro camadas principais:

- **Camada Estrutural:** tal como o próprio nome indica, permite estruturar os dados, identificar os recursos de forma única e fornecer os meios para a representação,

armazenamento e intercâmbio de informação, garantindo a integridade e confidencialidade dos dados;

- **Camada Sintáctica:** possibilita a estruturação dos dados associados a um recurso e a verificação da consistência desse recurso, de acordo com regras sintáticas formalmente descritas e previamente definidas e validadas;
- **Camada Semântica:** é responsável pela criação de vocabulários para a descrição semântica dos recursos e para a definição das relações existentes entre estes, de acordo com especificações formais, explícitas e compartilhadas de conceitos;
- **Camada Lógica:** permite declarar regras lógicas que podem ser executadas pelos agentes de software para realizar inferências automáticas e verificar o nível de coerência lógica dos recursos.

Aspectos tais como interoperabilidade e cooperação entre múltiplas fontes de informações, descrevendo diferenças sintácticas, semânticas e estruturais entre sistemas são igualmente importantes para a WS e não serão esquecidos ao longo desta dissertação.

As camadas *Logic*, *Proof* e *Trust*, bem como a *Signature* e a *Encryption*, não serão pormenorizadas no âmbito desta tese, uma vez que as tecnologias para implementar estas camadas ainda estão em discussão.

Assim, este capítulo incide sobre as tecnologias das camadas Estrutural e Sintáctica e o capítulo seguinte preocupa-se com as tecnologias das camadas Semântica e Lógica. Com o intuito de exemplificar as principais tecnologias para a Web Semântica introduzidas nesta secção e detalhadas nas secções e capítulo seguintes, o anexo C apresenta de forma prática através de um exemplo simples a evolução necessária da Web actual para a Web Semântica.

A integração das tecnologias ou linguagens para expressar esquemas de metadados, ontologias, agentes de software, entre outras, favorecerá o aparecimento de serviços Web que garantam a interoperabilidade e cooperação. Actualmente, a iniciativa W3C para a Web Semântica investe essencialmente nas seguintes frentes: SPARQL (*RDF Query Language and Protocol*), GRDDL (*Gleaning Resource Descriptions from Dialects of Languages*) e RIF (*Rule Interchange Format*) (W3C, 2006).

3.3- Tecnologias para as Camadas Estrutural e Sintáctica

O objectivo principal da WS não é, pelo menos para já, treinar as máquinas para que se comportem como pessoas, mas sim desenvolver tecnologias e linguagens que tornem a informação legível para as máquinas.

Nesta secção são abordadas, individualmente, cada uma das principais tecnologias para as camadas estrutural e sintáctica: a tecnologia XML(S) para fornecer a sintaxe e estruturar a informação, as tecnologias para os metadados para descrever essa informação e a tecnologia RDF(S) para descrever e atribuir significado a essa informação.

Para cada uma destas tecnologias, são referidas sucintamente as suas principais características e as linguagens, ferramentas e plataformas inerentes à sua construção e execução, para além de serem mencionadas as iniciativas e aplicações mais relevantes, nomeadamente no contexto educativo.

3.3.1- XML e XML Schema

A linguagem HTML (*HyperText Markup Language*) é uma linguagem básica de marcação derivada da *Standard Generalized Markup Language* (SGML) e utilizada na grande maioria das páginas Web. As principais limitações desta linguagem residem no facto de não admitir adicionar significado à informação e de não permitir estruturar os dados de um documento HTML, ou seja, não há distinção entre a codificação dos seus componentes básicos (conteúdo, estrutura e apresentação dos dados). As *tags* HTML não oferecem qualquer descrição acerca dos dados, apenas descrevem como uma página deve ser exibida (preocupação estética na apresentação da informação), e os dados são apresentados na linguagem natural.

A XML, sendo uma linguagem de marcação extensível, permite a definição de marcas e atributos. Podemos então dizer que XML é um conjunto de regras para definir etiquetas semânticas que dividem um documento em partes e identificam as diferentes partes desse documento (Harold, 1999). É a linguagem de marcação apropriada para a representação de dados, documentos e demais entidades, cuja essência se fundamenta na capacidade de agregar informação (Pimentel et al., 2000).

A tecnologia XML veio colmatar estas lacunas ao utilizar *tags* para descrever também a estrutura do documento e não apenas a sua apresentação ou aparência, para além de se assumir como a sintaxe recomendada para o intercâmbio de informação.

Com XML conseguimos não só garantir um melhor controlo em relação à interface, mas também uma maior flexibilidade na pesquisa, devido essencialmente à separação entre os rótulos de descrição e a apresentação do conteúdo.

Genericamente, os principais componentes da XML são: as declarações, os elementos, os atributos, as entidades, os comentários, as instruções de processamento e as regras definidas num DTD (*Document Type Definition*) ou num esquema XML ou *XML Schema*. Um exemplo básico de um documento XML pode ser visualizado na Figura 24.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<?xml-stylesheet type="text/xsl" href="http://www.meusite.pt/exemplo.xsl"?>
<!DOCTYPE BIBLIOTECA SYSTEM "http://www.meusite.pt/exemplo.dtd">
<BIBLIOTECA>
   <LIVRO>
      <CODIGO>001</CODIGO>
      <TITULO>Ensino a Distância & Tecnologias de Informação</TITULO>
      <VALOR>5 Euros</VALOR>
   </LIVRO>
   <LIVRO>
      <CODIGO>002</CODIGO>
      <TITULO>e-Learning e e-Conteúdos</TITULO>
      <VALOR>2 Euros</VALOR>
   </LIVRO>
</BIBLIOTECA>
```

Figura 24 – Exemplo básico de XML

A DTD e a XSD (*XML Schema Definition*) correspondem a duas tecnologias ou metalinguagens relacionadas com a XML, nomeadamente no que diz respeito aos conteúdos. Estas metalinguagens surgiram da necessidade de especificar as diversas utilizações da XML ou sub-linguagens (linguagens criadas a partir da XML). A estrutura de um documento em XML é, portanto, fruto da criação de um DTD ou de um esquema XML.

Um DTD não é mais do que um ficheiro que especifica um conjunto de regras para restringir e validar a estrutura do documento XML, ou seja, define a estrutura de um documento através da especificação dos seus elementos básicos, relacionamentos, atributos, entidades e o conjunto de *tags* associadas para marcar o documento. A utilização de um DTD facilita o processamento de documentos XML por aplicações, para além de controlar a coerência das marcações de XML. De acordo com o documento de XML visualizado anteriormente, o DTD correspondente está representado na Figura 25.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!ELEMENT BIBLIOTECA (LIVRO+)>
<!ELEMENT LIVRO (CODIGO,TITULO,VALOR)>
<!ELEMENT CODIGO (#PCDATA)>
<!ELEMENT TITULO (#PCDATA)>
<!ELEMENT VALOR (#PCDATA)>
```

Figura 25 – Exemplo básico de DTD

Uma vez que o DTD tem uma sintaxe particular, tentou encontrar-se uma forma de escrever em XML a estrutura dos documentos XML. O XML *Schema* é, portanto, uma alternativa ao DTD baseada em XML. Esta metalinguagem foi originalmente proposta pela Microsoft, mas em Maio de 2001 tornou-se uma recomendação oficial do W3C.

De acordo com o exemplo de XML apresentado, o esquema de XML poderia ser:

```
<?xml version="1.0"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
targetNamespace="http://www.meusite.pt"
 xmlns="http://www.meusite.pt"
 elementFormDefault="qualified">
   <xs:element name="BIBLIOTECA">
     <xs:complexType>
       <xs:sequence>
         <xs:element name="LIVRO">
           <xs:complexType>
             <xs:sequence>
               <xs:element name="CODIGO" type="xs:integer"/>
               <xs:element name="TITULO" type="xs:string"/>
               <xs:element name="VALOR" type="xs:string"/>
             </xs:sequence>
           </xs:complexType>
         </xs:element>
       </xs:sequence>
     </xs:complexType>
  </xs:element>
</xs:schema>
```

Figura 26 – Exemplo básico de XML Schema

Tal como os DTDs, os esquemas XML têm como objectivo definir os blocos de construção permitidos num documento XML:

- Elementos;
- Atributos;
- Elementos que são elementos filho;
- Ordem dos elementos filho;
- Número de elementos filho;
- Elementos que são vazios ou podem incluir texto;
- Tipos de dados para elementos e atributos;

- Valores por omissão e fixos para elementos e atributos.

Obviamente que haveria necessidade de alterar o documento XML para que, em vez de validar a estrutura do documento através do DTD, o fizesse através do esquema XML. Para tal, bastaria substituir a linha a) referente ao DTD pela linha b) inerente ao esquema XML:

```
Linha a) Referência ao DTD
<!DOCTYPE BIBLIOTECA SYSTEM "http://www.meusite.pt/exemplo.dtd">


Linha b) Referência ao XML Schema
<BIBLIOTECA
  xmlns="http://www.meusite.pt"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.meusite.pt biblioteca.xsd">
```

Finalmente, para apresentar e formatar o documento ao utilizador, recorreríamos a CSS (*Cascading Style Sheets*) ou XSL (*eXtensible Stylesheet Language*).

As folhas de estilo em cascata CSS já eram utilizadas com o HTML para separar a forma do conteúdo. Genericamente, as CSS também podem ser utilizadas com XML, tal como se utilizam em HTML.

A linguagem XSL tem o mesmo objectivo, mas é específica para XML. Esta linguagem não se limita a definir qual o estilo a aplicar a cada elemento do documento XML, já que também se podem utilizar pequenas instruções típicas das linguagens de programação. Para além disso, a saída não se limita a um documento HTML, podendo ser de outros tipos: documento escrito em WML (para WAP), documento de texto ou outro documento XML. Sendo uma linguagem muito mais potente que a CSS, recomenda-se a sua utilização. Uma das suas principais vantagens é permitir que um documento XML seja visualizado em vários dispositivos distintos. Para tal, por cada documento XML teríamos um documento XSL para cada dispositivo. Por exemplo, um para um navegador Mozilla Firefox, outro para MS Internet Explorer, outro para um telemóvel Motorola e outro para um telemóvel Nokia, etc. E se amanhã aparecer um novo dispositivo, bastará criar um novo documento XSL para que os nossos documentos XML possam ser visualizados nele.

Por conseguinte, recorrendo a diferentes folhas de estilo em linguagem XSL, os dados do documento XML do exemplo poderiam ser apresentados em vários suportes (monitor do computador, visor do telemóvel, ecrã de televisão, etc). Um exemplo do documento XSL para apresentação do XML no navegador Internet Explorer pode ser visualizado na Figura 27.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" version="1.0">
<xsl:output method="html"/>
<xsl:template match="/BIBLIOTECA">
<html>
  <head>
      <title>Biblioteca</title>
  </head>
   <body bgcolor="#FFFFFF">
      <h1>Revistas da ESEB</h1>
      <table border="1" cellspacing="0" cellpadding="5">
         <xsl:apply-templates/>
      </table>
   </body>
</html>
</xsl:template>
<xsl:template match="LIVRO">
   <tr>
      <xsl:apply-templates/>
   </tr>
</xsl:template>
<xsl:template match="CODIGO">
   <xsl:call-template name="COLUNA"/>
</xsl:template>
<xsl:template match="TITULO">
   <xsl:call-template name="COLUNA"/>
</xsl:template>
<xsl:template match="VALOR">
   <xsl:call-template name="COLUNA"/>
</xsl:template>
<xsl:template name="COLUNA">
   <td>
      <xsl:value-of select="."/>
   </td>
</xsl:template>
</xsl:stylesheet>
```

Figura 27 – Exemplo básico de XSL

No âmbito da WS, é aconselhável seguir um conjunto de recomendações ou boas práticas, de forma a evitar incompatibilidades inerentes à apresentação da informação (por exemplo: os *frames* deixaram de ser uma estratégia viável e a utilização de tabelas levanta algumas dúvidas).

A orientação da linguagem HTML para a apresentação implica o recurso a palavras-chave aquando da busca de informações, requisitando que a interpretação semântica seja efectuada pelo utilizador. Com a linguagem XML é dado o primeiro passo para que a interpretação semântica dos documentos seja efectuada também pelas máquinas, uma vez que os recursos podem ser relacionados e os conteúdos descritos de forma clara, facilitando a busca de informações ao recorrer a metadados.

A XML assume-se como uma tecnologia flexível que fornece a sintaxe para documentos estruturados, através da qual o significado pode ser comunicado independentemente da

plataforma, facultando a criação de uma estrutura arbitrária para os documentos, mas nada diz acerca do significado dessa estrutura. Esta tarefa é deixada para a linguagem RDF (ou tecnologias de metadados similares baseadas em XML) que permite descrever a informação de forma não ambígua para depois ser processada pelas máquinas (Thompson, 2004).

A linguagem RDF, bem como outras linguagens de representação da informação e do conhecimento, adoptaram a XML como sintaxe de representação devido essencialmente aos seguintes motivos:

- XML é legível tanto para as pessoas como para as máquinas, evitando ambiguidades;
- XML é a linguagem recomendada para a anotação de documentos;
- XML foi projectada para a representação de dados semi-estruturados;
- XML permite a sua apresentação em diversos formatos, de acordo com o dispositivo de saída;
- XML garante a interoperabilidade e intercâmbio de objectos de informação, independentemente da plataforma.

3.3.2- Metadados

Embora não sejam um conceito recente, os metadados têm vindo a assumir uma importância crescente no âmbito da gestão e recuperação de informação. É comummente aceite que metadados são “dados sobre dados” ou correspondem a “quaisquer indicações sobre recursos da informação” (Garshol, 2004). São um conjunto de dados estruturados que descrevem, explicam e localizam a informação, ou seja, tornam mais fácil recuperar, usar ou controlar um recurso da informação (DCMI, 2002).

O recurso informação pode ser um documento, uma página Web, um módulo de conteúdos reutilizável ou um objecto de informação de diferente estrutura e granularidade. Os metadados não se aplicam apenas a informação textual, mas também a outros recursos digitais tais como imagens, músicas ou vídeos. A adição de descrições textuais a esse tipo de recursos tem vindo a ser uma forma de os indexar e catalogar de modo a facilitar a sua localização, permitindo melhores resultados na recuperação de informação.

Com o crescimento da complexidade e volume da informação digital disseminada através da Internet, os metadados tornam-se vitais não só para facilitar o acesso à informação (especialmente no que diz respeito à precisão da procura), mas também para suportar o controlo e a gestão da informação armazenada (Oliveira, 2003).

Os metadados fornecem pistas, rotuladas pelas *tags* XML, para que as máquinas possam "compreender" a informação. Permitem que os computadores tenham acesso a camadas de conhecimento que anteriormente só poderiam existir ou ser utilizadas pelo cérebro humano através da interpretação do significado dos documentos. A adição de um maior nível semântico à camada de informação permitirá que os computadores possam realizar inferências com vista à geração de novo conhecimento ou à sua reutilização.

A criação dos metadados pode ocorrer aquando da criação do recurso de informação à qual se referem ou aquando do processo de classificação e indexação do documento. Face à falta de especialistas para classificar e indexar os recursos no contexto de um sistema de e-Learning, muitas vezes a concepção de um recurso digital e a criação dos metadados para esse recurso ocorrem ao mesmo tempo e o autor do recurso é também o autor dos metadados, ou seja, o professor.

Os metadados podem ser parte integrante do documento ao qual se referem ou estarem armazenados num documento diferente daquele que contém a informação propriamente dita. É desejável que os metadados e os dados do recurso sejam mantidos em documentos separados para permitir diferentes graus de acesso aos metadados (pode haver necessidade de disponibilizar uma maior quantidade de metadados a um professor do que a um aluno), para oferecer diferentes descrições consoante o contexto (aquando da reutilização de um recurso por diferentes autores, pode haver necessidade de proceder a alterações somente nos metadados, mantendo-se inalterável o recurso e vice-versa) ou mesmo para fornecer diferentes manifestações de um objecto conceptual (um documento textual em PDF e o mesmo documento numa versão áudio armazenada num ficheiro MP3 referidas na mesma descrição ou metadados). Esta estratégia pode não ser necessária se a linguagem para a descrição dos recursos permitir a riqueza semântica suficiente, como veremos mais adiante, nomeadamente na secção 3.3.3- *Resource Description Framework.*

Os metadados devem ser normalizados e, para tal, entidades oficiais responsáveis pela definição de normas, tais como *World Wide Web Consortium* (W3C), *International Standards Organization* (ISO), *American National Standards Institute* (ANSI), *Institute of Electrical*

and Electronics Engineers (IEEE), *Dublin Core Metadata Initiative* (DCMI), *Instructional Management Systems Global Learning Consortium* (IMS/GLC), ARIADNE (*ARIADNE Foundation*) e CanCore (*CanCore Metadata Initiative*), criaram grupos de trabalho com vista a definir protocolos e normas para especificação e gestão de metadados.

Os esforços desses grupos de trabalho, entre outros, resultaram em normas ou especificações e esquemas para metadados (conjuntos de elementos de metadados), das quais se destacam: MARC 21 *Concise Format for Bibliographic Data*, *Dublin Core Metadata* (DCM), IEEE - *Standard for Learning Object Metadata* (IEEE-LOM), *IMS Learning Resource Metadata* (IMS-LRM), *AGLS Metadata Element Set* (AGLS-MES), *Information technology - Metadata Registries* (MDR), *Geographic Information Metadata* (GIM), *Model for Metadata for Multimedia Information* (MMMI), *Meta Data Interchange Specification* (MDIS), ARIADNE, *Platform for Internet Content Selection* (PICS), *Multimedia Content Description Interface* (MPEG-7), entre outras. Mais recentemente, surgiu uma abordagem diferente, mas complementar, uma vez que permite expressar metadados, tais como os conjuntos de elementos de metadados DCM ou IEEE-LOM. Referimo-nos à metalinguagem para descrever recursos ou linguagem para expressar metadados *Resource Description Framework* (RDF). A utilização desta metalinguagem (ou dos metadados que expressa) adiciona camadas de conhecimento aos conteúdos, favorecendo uma verdadeira compreensão da informação publicada não só na perspectiva dos humanos, como na das máquinas.

O esquema de metadados mais popular é o DCM (não esquecendo o MARC 21 no âmbito das bibliotecas e centros de documentação), enquanto que o LOM é o mais utilizado no campo da Educação. Contudo, a RDF é a proposta que mais se destaca no contexto particular da WS, uma vez que é uma recomendação W3C e permite expressar os conjuntos de elementos de metadados dos esquemas DCM e LOM.

3.3.2.1- Dublin Core Metadata

As normas de metadados *NISO Z39.85-2001* e *ISO 15836-2003*, que dizem respeito à especificação formal do *Dublin Core Metadata Element Set* (DCMES), correspondem a um conjunto de especificações para a descrição dos recursos da informação de um qualquer domínio do conhecimento através de um vocabulário de 15 elementos. Um recurso de informação é “qualquer coisa que tem uma identidade” (*anything that has identity*), ou seja, é identificado por um URI (DCMI, 2002).

O DCMES usa quinze elementos (designados *Core Elements*) para descrever qualquer recurso de informação (DCMI, 2002):

- *Title* (título ou nome atribuído ao recurso);
- *Creator* (entidade responsável pela criação ou existência do recurso);
- *Subject* (assunto e palavras-chave que caracterizam o conteúdo do recurso);
- *Description* (descrição ou resumo do conteúdo do recurso);
- *Publisher* (entidade responsável por editar, publicar ou manter acessível o recurso);
- *Contributor* (entidade responsável por qualquer contribuição para o conteúdo do recurso);
- *Date* (data inerente à criação ou publicação do recurso);
- *Type* (tipo, função, natureza ou género do recurso);
- *Format* (formato do recurso: físico ou digital, tamanho ou duração);
- *Identifier* (referência para identificar o recurso num determinado contexto: URI, ISBN, etc.);
- *Source* (referência a um recurso de onde o recurso actual deriva);
- *Language* (língua do conteúdo intelectual do recurso: pt, en, fr);
- *Relation* (referência a um recurso relacionado);
- *Coverage* (extensão, alcance ou âmbito do recurso);
- *Rights* (gestão dos direitos inerentes ao recurso: direitos de propriedade intelectual, direitos de autor, entre outros).

Estes elementos são facultativos, podem ser repetidos sempre que necessário e podem ser agrupados em três grupos principais de acordo com o conteúdo, com os direitos reservados e com a instanciação (Oliveira, 2003), tal como se pode verificar na Tabela 3 e na Figura 28.

Conteúdo	Protecção de direitos	Instanciação
Title *Subject* *Description* *Type* *Source* *Relation* *Coverage*	*Creator* *Publisher* *Contributor* *Rights*	*Date* *Format* *Identifier* *Language*

Tabela 3 – Grupos de elementos DCMES

```
<?xml version="1.0" encoding="iso-8859-1"?>
<metadata xmlns:dc="http://purl.org/dc/elements/1.1/"
       xmlns:dcterms="http://purl.org/dc/terms/"
xmlns:dcmitype="http://purl.org/dc/dcmitype/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
       <dc:title>Exemplo de Pacote SCORM</dc:title>
       <dc:creator>Vitor Gonçalves</dc:creator>
       <dc:subject>SCORM</dc:subject>
       <dc:subject>Objectos de Aprendizagem</dc:subject>
       <dc:description>Um pacote é um conjunto de conteúdos ou páginas Web
empacotadas de acordo com as normas SCORM ou AICC para objectos de aprendizagem.
Estes pacotes podem incluir páginas Web de texto, gráficos, Javascript, Flash ou
qualquer outro conteúdo que possa ser visualizado em browsers Web.</dc:description>
       <dc:publisher>Vitor Gonçalves</dc:publisher>
       <dc:contributor>Francisco Gonçalves</dc:contributor>
       <dc:date>2006-10-08</dc:date>
       <dc:type>scorm</dc:type>
       <dc:format>text/html</dc:format>
       <dc:identifier>http://meusite/scorm/view.php?id=29</dc:identifier>
       <dc:source></dc:source>
       <dc:language>pt_utf8</dc:language>
       <dc:relation>http://meusite/view.php?id=3</dc:relation>
       <dc:coverage></dc:coverage>
       <dc:rights>Copyright 2006 - VGe-Learning - All rights reserved.</dc:rights>
</metadata>
```

Figura 28 – Esquema genérico de metadados DCMES em XML

A norma *Dublin Core Metadata* (DCM) inclui dois níveis para a descrição de recursos: simples (*simple*) e qualificado (*qualified*). Os esquemas de metadados que usam apenas os quinze elementos mencionados designam-se por ***Simple Dublin Core Metadata***. Cada um destes quinze elementos pode ser refinado com recurso a um conjunto limitado de qualificadores (*Dublin Core Qualifiers*). Existem duas classes de qualificadores: refinamento dos elementos (permitem que o significado de um elemento seja mais restrito ou específico) e esquemas de codificação (identificam esquemas que facilitam a interpretação do valor de um elemento). Os esquemas de metadados que, para além dos 15 elementos que constituem o DCMES, usam outros elementos (*Audience*, *Provenance*, *RightsHolder*, *accrualMethod*, *accrualPeriodicity*, *accrualPolicy*, *instructionalMethod*) e os elementos de refinamento designam-se por ***Qualified Dublin Core Metadata*** (Hillmann, 2005).

Tanto o *Simple DC* como o *Qualified DC* podem ser codificados em RDF, para além de serem expressos em XML, de acordo com as especificações do DCMI (*Dublin Core Metadata Initiative*): *Expressing Simple Dublin Core in RDF/XML* e *Expressing Qualified Dublin Core in RDF/XML*. Voltaremos a focar este assunto na secção 3.3.4- Interoperabilidade de esquemas de metadados.

Contudo, se um agente não entender um termo de refinamento específico para um determinado elemento, ignorá-lo-á e processará apenas o valor do metadado como se o elemento em questão não estivesse refinado. Mas, mesmo que um esquema de codificação não seja entendido por um agente, pelo menos o valor continuará a ser útil e legível para um ser humano.

Existem diversas aplicações da tecnologia DCM, nomeadamente utilitários, modelos para criação de metadados, ferramentas de edição de modelos, aplicações para extracção de metadados, exemplos de produção automática de metadados, exemplos de conversão entre formatos de metadados, ferramentas e exemplos de perfis de metadados, etc (DCMI, 2006).

Na geração automática dos metadados aplicável aos recursos digitais podem identificar-se dois métodos ou situações: a extracção de metadados e a recolha de metadados (Greenberg, 2005).

A extracção de metadados envolve a semântica dos recursos de informação e emprega técnicas sofisticadas de indexação automática para produzir metadados estruturados (anotações) para a representação do objecto de informação (por exemplo: algoritmos que geram palavras-chave). A recolha de metadados confia nas capacidades das aplicações para capturar os metadados previamente criados pelas pessoas, pelas máquinas, ou por ambos (por exemplo, os cabeçalhos de páginas Web contêm informações que podem ser capturadas). Destacamos os exemplos de aplicações referidos no Website da iniciativa DCMI (2006). Não menosprezando obviamente algumas iniciativas de investigação académicas, normalmente decorrentes de projectos de doutoramento, tais como os trabalhos de Ferreira e Kofuji (2006), Lourenso (2005), Shi e outros (2003), Han e outros (2003); Takasu (2003), Barbosa (2003), Rezende e Bax (2002), Preto e outros (1999), Jenkins e outros (1999), entre outros.

No contexto das ferramentas para o e-Learning, o editor eXe (*eLearning XHTML editor*) é uma das ferramentas que usa a especificação DCM para anotar os objectos de aprendizagem.

3.3.2.2- Learning Object Metadata

A Web pode ser vista também como uma mediateca de recursos educativos. No entanto, encontrar os objectos de aprendizagem que a compõem é cada vez mais uma missão impossível. Os metadados assumem-se como uma das formas de resolver este problema.

Analogamente à informação usada pelos bibliotecários para classificar, indexar e catalogar os livros e outros recursos educativos, os metadados podem ser aplicados a objectos de aprendizagem com o intuito de esclarecer quem os criou, que ferramentas usam, onde estão armazenados, etc. Nesta perspectiva, a norma *Learning Object Metadata* (LOM) tem como principal objectivo permitir que esses objectos de aprendizagem sejam geridos, encontrados e avaliados através dos correspondentes metadados.

No campo da Educação destacaram-se as especificações IEEE LOM, ARIADNE e IMS-LRM. Actualmente, IEEE LOM (*IEEE Learning Object Metadata*) é a norma mais comum. Por isso, têm sido publicadas recomendações para a tradução dos restantes esquemas de metadados para LOM. Por exemplo, *Alignment of IMS Learning Resource Meta-data with IEEE Learning Object Metadata*: *Guidelines for Using the IMS LRM v1.2.1 to IEEE LOM 1.0 Transform* (IMS, 2006) e *Mapping between ARIADNE and LOM* (Najjar, 2003; ARIADNE, 2004).

A norma IEEE LOM é um conjunto de especificações que permitem descrever e facilitar a localização e a recuperação de LOs. Os atributos que permitem a descrição de um objecto de aprendizagem podem ser agrupados em nove categorias (IEEE-LTSC, 2002):

- Geral (*General*) - refere-se a informação geral que descreve o LO como um todo (Elementos: *Identifier, Title, Catalog Entry, Catalog, Entry, Language, Description, Keywords, Coverage, Structure, Aggregation level*);
- Ciclo de vida (*Lifecycle*) - agrupa os atributos relacionados com o ciclo de vida de um recurso: a história de um LO e respectivos contributos para essa evolução, para além dos atributos inerentes ao estado actual desse objecto (Elementos: *Version, Status, Contribute, Role, Entity, Date*);
- Meta-metadados (*Metametadata*) - reúne informações sobre a própria instância dos metadados (em vez do LO que a instância do metadado descreve). Ou seja, agrupa os atributos que descrevem os metadados que indexam o LO (Elementos: *Identifier, Catalog, Entry, Contribute, Role, EntitytDate, Metadata Squema, Language*);
- Técnica (*Technical*) - agrupa os requisitos e características técnicas do LO (Elementos: *Format, Size, Location, Requirements, Type, Name, Minimum Version, Maximum version, Instalattion Remarks, Other Plataform requirements, Duration*);
- Educacional (*Educational*) - agrupa os atributos educativos e pedagógicos do LO (Elementos: *Interactivity Type, Learning Resource Type, Interactivity level, Semantic*

density, Intended End User Role, Context, Typical Age Range, Difficulty, Typical Learning Time, Description, Language);

- Direitos (*Rights*) - agrupa os direitos de propriedade intelectual e condições de utilização inerentes ao LO (Elementos: *Cost, Description, Copyright and others restrictions*);
- Relação (*Relation*) – incide nos atributos inerentes às relações entre recursos, ou seja, agrupa características que definem a relação entre o LO e outros objectos educacionais correlacionados (Elementos: *Kind, Resource, Identifier, Description, Catalog Entry*);
- Anotação (*Annotation*) - permite comentários sobre o uso educacional do objecto e fornece informação sobre quando e por quem foram criados os comentários (Elementos: *Person/Entity, Description, Date*);
- Classificação (*Classification*) - descreve a posição do LO em relação a um sistema particular de classificação, tal como UDC (Universal Decimal Classification) ou ARIADNE, etc. (Elementos: *Purpose, Taxon Path, Source, Taxon, Id, Entry, Description, Keywords*).

De salientar que o autor dos metadados deve recorrer, sempre que possível, a vocabulários existentes para descrever o LO. Por exemplo, é recomendável o recurso a taxonomias ou tesauros para adicionar assuntos ao elemento *Keywords* ou o recurso a RFC 2048 para adicionar os literais MIME (*Multipurpose Internet Mail Extension*) ao elemento Format. Para apoiar a anotação de alguns dos outros elementos LOM são sugeridas também listas de valores predefinidos (*tokens*). Referimo-nos mais concretamente aos elementos: *Structure, Aggregation Level, Status, Type, Name, Interactivity Type, Learning Resource Type, Interactivity Level, Semantic Density, Intended End User Role, Context, Difficulty, Cost, Copyrights And Other Restrictions, Kind, Purpose, General_Contribute_Role* e *Meta_Metadata_Contribute_Role*. Em relação à identificação de pessoas ou instituições, é recomendado o uso da representação *vCard*, tal como definido na especificação IETF RFC 2426:1998.

Mais detalhes sobre a sintaxe XML/LOM podem ser consultados directamente na norma IEEE P1484.12.x, nomeadamente IEEE P1484.12.3 *Standard for Extensible Markup Language* (XML) *Schema Definition Language Binding for Learning Object Metadata*.

```
<lom>

  <general>
        <!-- General Section -->
        <title>
                <string language="en">LOM Example: LO Metadata</string>
                <string language="pt">Exemplo LOM: Metadados para LO</string>
        </title>
          <language>en</language>
        <description>
          <string language="en">Metadata is information about an learning object, be it
physical or digital.</string>
<string language="pt">Metadado são informações sobre um objecto de aprendizagem, seja ele
físico ou digital.</string>
        </description>
        <keyword>
          <string language="en">learning object</string>
          <string language="pt">objecto de aprendizagem</string>
          <string language="fr">objet d'apprentissage</string>
        </keyword>
  </general>

  <lifeCycle>
        <!-- Lifecycle Section -->
  </lifeCycle>

  <metaMetadata>
        <!-- Metametadata Section -->
  </metaMetadata>

  <technical>
        <!-- Technical Section -->
  </technical>

  <educational>
        <!-- Educational Section -->
        <description>
          <string language="en">Comments on how this resource is to be used.</string>
          <string language="pt">Comentários de como este recurso deve ser usado.</string>
        </description>
  </educational>

  <rights>
        <!-- Rights Section-->
  </rights>

  <relation>
        <!-- Relations Section -->
  </relation>

  <annotation>
        <!-- Annotation Section -->
        <entity>vCard of utilizador ABC</entity>
        <date>
          <dateTime>2005-02-01</dateTime>
        </date>
        <description>
          <string language="en">Comments of user: pupil/teachear</string>
          <string language="pt">Comentários do utilizador: aluno/professor</string>
        </description>
    </annotation>

  - <classification>
       <!-- Classification Section -->
    </classification>

</lom>
```

Figura 29 – Esquema genérico de metadados LOM em XML

O recurso a metadados LOM por parte dos mecanismos de pesquisa (e dos agentes) na busca de recursos educativos favorecerá a devolução de resultados mais precisos. Não obstante, a avaliação dos mesmos continuará a depender em grande parte da opinião do utilizador. Neste sentido, o sistema de suporte deverá permitir que os utilizadores possam comentar os recursos de aprendizagem (por exemplo, através da categoria de atributos "anotação"). Só assim, por um lado, os alunos poderão consultar os comentários feitos por outros para estimar o valor de um recurso de aprendizagem, para obter orientação de como usá-lo correctamente ou de como relacioná-lo com outros e, por outro, os professores poderão usar esses comentários para perceber quais as vantagens e desvantagens dos recursos que publicaram a fim de os poderem alterar (SHEN et al., 2002).

Em suma, os principais objectivos do projecto LOM são (LOMWG, 2002):

- Permitir que alunos ou formandos e professores ou instrutores possam procurar, avaliar, aceder e utilizar objectos de aprendizagem;
- Permitir a partilha, intercâmbio e reutilização de objectos de aprendizagem através de qualquer tecnologia de suporte a sistemas de aprendizagem, assegurando a interacção e compatibilidade entre os diversos ambientes de ensino/aprendizagem;
- Favorecer o desenvolvimento de objectos de aprendizagem sob a forma de peças que possam ser combinadas ou decompostas de várias formas;
- Possibilitar que os agentes de software combinem automática e dinamicamente objectos de aprendizagem em lições personalizadas com base nos requisitos de cada aluno;
- Facultar combinações entre múltiplos objectos de aprendizagem favorecendo um ambiente de aprendizagem distribuído aberto.

No entanto, a norma LOM não tem sido tão utilizada quanto seria de esperar no campo educativo. A sua aplicação limitou-se à descrição de objectos de aprendizagem em alguns sistemas de e-Learning e correspondentes ferramentas de desenvolvimento de e-cursos ou em redes P2P (*peer to peer*), tais como Edutella (*http://edutella.jxta.org*). Estes dois tipos de utilização correspondiam a ambientes fechados, sendo inacessível o uso directo dos dados e metadados a partir da Web (Bourda e Bich-Liên, 2003). Com vista a garantir a interoperabilidade e reutilização dos recursos educativos exigia-se uma infra-estrutura mais aberta e suficientemente flexível para suportar diversos conjuntos de metadados (Nejdl et al., 2001). A infra-estrutura

distribuída do projecto Edutella não poderia oferecer apenas anotação ou descrição de recursos educativos baseada em LOM-IMS e ADL SCORM pelo que, para o intercâmbio desses conjuntos de metadados, passou também a usar a recomendação para metadados RDF(S) do W3C e um modelo de consulta baseado em Datalog e apropriado para o intercâmbio de consultas em toda a rede Edutella. A tradução entre os esquemas XML e RDF, inerentes a metadados LOM para objectos de aprendizagem, não é difícil, uma vez que estes esquemas definem basicamente os metadados estruturados hierarquicamente, sendo as diferenças entre RDF e XML pouco relevantes (Nejdl et al., 2002).

Para além do Edutella, têm vindo a surgir outros sistemas e projectos com o intuito de reunir, partilhar e reutilizar recursos de aprendizagem dispersos e de disponibilizar ao utilizador final uma interface uniforme que permita a procura, acesso e avaliação desses recursos, dos quais se destacam: *ARIADNE Knowledge Pool System* (*http://ariadne.cs.kuleuven.ac.be*); SMETE - *U.S.-based Science, Mathematics, Engineering and Technology Education Digital Library* (*http://www.smete.org*); GEM - *Gateway to Educational Materials digital library* (*http://www.thegateway.org*), EdNA - *Educational Network Australia* (*http://www.edna.edu.au*), etc. Mais recentemente, destaca-se o projecto AMG (*Automating Metadata Generation*) e a correspondente API SAmgI (*Simple AMG Interface*) no âmbito da geração automática de metadados e disponível através do URL *http://ariadne.cs.kuleuven.ac.be/amg/Intro.jsp* (Cardinaels e Meire, 2005).

No contexto das ferramentas para o e-Learning, o RELOAD editor, RELOAD LD editor e LomPad são algumas das ferramentas que usam a especificação LOM para anotar os objectos de aprendizagem.

3.3.2.3- Multimedia Content Description Interface

Genericamente, MPEG é uma família de normas usadas na codificação de informação audiovisual num formato digital de compressão. MPEG-1 é a norma base de codificação para produtos como o CD vídeo e o MP3. A MPEG-2 é a norma-base para produtos como a televisão digital e os DVDs. MPEG-4 é a norma para os conteúdos multimédia para a Web. MPEG-7 é a norma para a descrição e a procura de conteúdos audiovisuais. E, finalmente, a norma MPEG-21 preocupa-se com as questões de distribuição, transmissão e consumo de conteúdos multimédia, com vista a permitir o acesso a uma grande variedade de conteúdos, apostando na interoperabilidade e garantindo os direitos de propriedade.

A norma *Multimedia Content Description Interface*, ou simplesmente MPEG-7 foi desenvolvida pelo *Moving Picture Experts Group* (MPEG). É uma norma ISO/IEC (*International Standards Organization/International Electro-technical Commission*) para a descrição de conteúdos audiovisuais em ambientes multimédia ou hipermédia, com vista a suportar a interpretação do significado da informação. O MPEG-7 veio, então, possibilitar a compreensão de imagens e a localização, filtragem e recuperação de informação em filmes, vídeos e músicas (Staab, 2000).

A descrição deste tipo de conteúdos pode variar consoante o seu tipo (imagens animadas, gráficos, áudio ou vídeo), a sua aplicação e os utilizadores a que se destinam, pelo que o conjunto de atributos descritivos também pode variar. Estes atributos descritivos ou descritores (*Descriptors* - Ds) tanto podem ser parte integrante do documento audiovisual, ou seja, estar incluídos no mesmo sistema de armazenamento, como podem estar noutro sistema.

Os descritores agrupam-se em esquemas de descrição (*Description Schemes* - DSs) criados ou modificados através da Linguagem de Definição de Descrições (*Description Definition Language* - DDL). A DDL é uma linguagem baseada na linguagem XML e pode ser decomposta em: componentes estruturais XML *Schema*, componentes de tipos de dados XML *Schema* e extensões específicas MPEG-7 (Martínez, 2003).

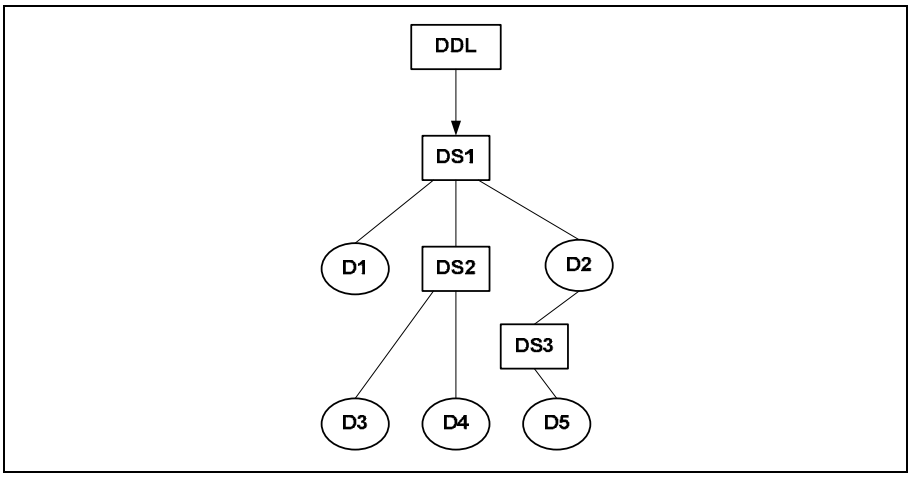


Figura 30 – Principais elementos MPEG-7

Um descritor não é mais do que a representação de uma característica, ou seja, define a sintaxe e a semântica da representação dessa característica. Os esquemas de descrição especificam estrutura e semântica (relações entre descritores e esquemas de descrição).

Os DSs não são mais do que estruturas de metadados para descrever e anotar conteúdo audiovisual. Ou seja, os DSs fornecem uma forma normalizada de descrever em XML os

conceitos importantes relacionados com os conteúdos audiovisuais e os conteúdos de gestão a fim de facilitar a procura, a indexação, a filtragem e o acesso. Não obstante, a forma como a descrição do conteúdo é codificada e armazenada ou a forma como a mesma é usada para responder ao utilizador cai fora dos limites desta norma. Ou seja, uma vez que pode ser usada por diversas aplicações, a norma MPEG-7 não define o processo de extracção dos descritores do conteúdo, nem a sua posterior aplicação ou utilização.

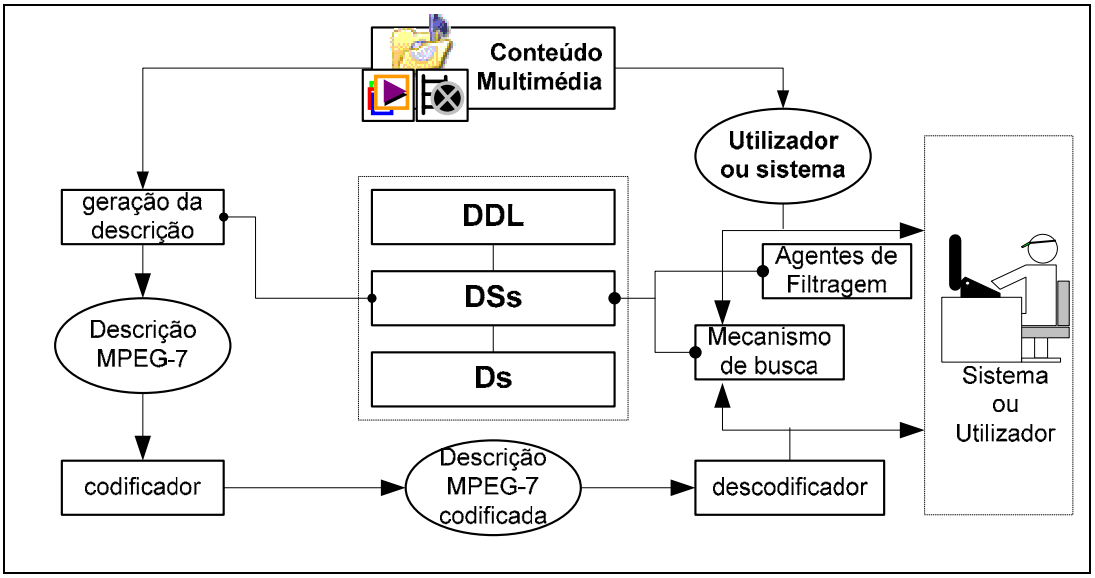


Figura 31 – Contexto da norma MPEG-7

A norma MPEG-7 divide-se em sete partes, as quais passamos a descrever sucintamente:

- ***MPEG-7 System***: corresponde às ferramentas do sistema necessárias à preparação das descrições MPEG-7 no que diz respeito ao seu transporte e armazenamento eficientes, à sua sincronização e à gestão da propriedade intelectual, com vista a definir a arquitectura final;
- ***MPEG-7 DDL***: refere-se à linguagem que permite definir a sintaxe das ferramentas de descrição MPEG-7, a criação de novos DSs e Ds e a modificação de DSs já existentes;
- ***MPEG-7 Visual***: corresponde às ferramentas de descrição que lidam exclusivamente com as descrições visuais;
- ***MPEG-7 Audio***: refere-se à parte da norma inerente às ferramentas de descrição que tratam somente das descrições de áudio;

- ***MPEG-7 MDS***: esta parte da norma (*Multimedia Description Schemes*) trata das características genéricas e das descrições multimédia. Estes esquemas de descrição multimédia incluem descrição de conteúdos (estrutura e semântica), gestão de conteúdos (informação sobre direitos de propriedade, autor, formato, preço, etc.), estrutura e organização dos conteúdos, bem como sobre a navegação e acesso aos mesmos.
- ***MPEG-7 Reference Software***: consiste na implementação das partes relevantes do MPEG-7. Para tal, existe o software XM (*eXperimentation Model*) enquanto plataforma de simulação de Ds, DSs, esquemas de codificação (*coding schemes*) e DLL;
- ***MPEG-7 Conformance***: esta parte da norma contém instruções e procedimentos para testar as implementações MPEG-7.

A anotação de documentos multimédia no contexto da WS não foi esquecida pelo W3C, tendo sido criado recentemente um novo grupo de trabalho para investigar os problemas inerentes à anotação semântica do multimédia e à integração dos metadados no multimédia para a Web Semântica (Stamou, 2006; Smith, 2006).

3.3.3- Resource Description Framework

A *Resource Description Framework* (RDF) é uma *framework*, tecnologia ou linguagem para representar informação na Web (Lassila e Swick, 1999). Sendo uma das aplicações da XML, usa a notação XML como sintaxe de codificação e de descrição dos metadados.

Os metadados são utilizados para descrever as características de um recurso e as suas relações. Assim, o objectivo principal da tecnologia ou metalinguagem RDF é definir um mecanismo para descrever recursos independentemente da área do conhecimento.

Inicialmente, a especificação RDF consistia essencialmente num modelo de dados e numa sintaxe para o esquema de metadados. O modelo de dados era responsável pela descrição dos recursos, enquanto que a sintaxe XML permitia criar e partilhar os metadados entre as aplicações. Esta especificação traduzia-se na recomendação *Resource Description Framework (RDF) Model and Syntax Specification W3C Recommendation* de 22 Fevereiro de

1999 (Lassila e Swick, 1999). Entretanto foram surgindo algumas propostas de revisão. Actualmente, a recomendação RDF consiste nos seguintes documentos:

- RDF *Primer* (Manola e Miller, 2004): introduz a linguagem RDF e descreve algumas das suas aplicações com o objectivo principal de ajudar os projectistas de sistemas da informação e os programadores de aplicações a compreender as características da RDF e a forma de a usar;
- RDF *Concepts and Abstract Syntax* (Klyne e Carroll, 2004): define a sintaxe abstracta na qual a RDF se baseia e da qual se serve para ligar a sua sintaxe concreta à sua semântica formal. Inclui também a discussão dos objectivos do projecto (*design*), dos conceitos-chave, do *datatyping*, da normalização dos caracteres e da manipulação das referências URI;
- RDF *Semantics* (Hayes, 2004): apresenta a especificação de uma semântica precisa e correspondentes sistemas de regras de inferência, tanto para a RDF como para o seu esquema RDFS (este esquema será detalhado no capítulo seguinte);
- RDF/XML *Syntax* (Beckett, 2004): define a sintaxe de XML para RDF, designada RDF/XML, em termos de *namespaces* em XML (Bray et al., 1999), XML *Information Set* (Cowan e Tobin, 2001) e XML *Base* (Marsh, 2001). A gramática formal para a sintaxe é anotada com as acções que geram triplas do grafo RDF, tal como definido em *RDF Concepts and Abstract Syntax*. As triplas são escritas usando o formato de serialização *N-Triples* (subconjunto da notação N3) do grafo RDF que permite uma captura mais precisa do mapa da informação num formato processável por máquinas;
- RDF *Vocabulary Description Language 1.0 - RDF Schema* (Brickley e Guha, 2004): assumindo a RDFS como uma extensão semântica da RDF, esta especificação explica como usar a linguagem RDF para descrever vocabulários de RDF e define o vocabulário para esse fim;
- RDF *Test Cases* (Grant e Beckett, 2004): descreve os exemplos distribuídos pelo grupo RDF *Core Working Group*, nomeadamente casos de teste de conformidade com a gramática formal RDF e outros requisitos técnicos.

Podemos desde já reafirmar que esta abordagem é significativamente diferente das anteriormente descritas, uma vez que não se limita a anotar recursos de informação através de metadados seguindo um conjunto de descritores predefinido. Muito pelo contrário, constitui

uma camada complementar que permite expressar os metadados, tais como DCM e LOM. A linguagem RDF contribui para a melhoria de vários cenários, dos quais se destacam:

- Catalogar os recursos de informação e as suas relações num sistema (Website, página Web ou biblioteca digital, etc) ou entre vários sistemas;
- Classificar o conteúdo e descrever os recursos de informação;
- Pesquisar informação de forma mais precisa, permitindo que os motores de busca ofereçam melhores resultados;
- Filtrar com maior precisão dados para obter sistemas de avaliação de conteúdo mais viáveis;
- Representar conjuntos de documentos como um único e grande documento lógico, quando apropriado;
- Facilitar o intercâmbio e a partilha de conhecimento através de agentes de software inteligentes;
- Estabelecer relações seguras entre documentos e computadores para facilitar a troca de ideias e de recursos;
- Descrever os direitos de propriedade intelectual nas páginas Web;
- Expressar as preferências de privacidade de um utilizador ou as políticas de privacidade de um Website.

A Web actual é um conjunto de recursos e hiperligações (*links*). Os recursos são identificados por URIs (*Uniform Resource Identifiers*). Estes identificadores de recursos asseguram que cada objecto na Web esteja relacionado com apenas um conceito. O URL (*Uniform Resource Locator*) é um exemplo típico de URI. Normalmente, o utilizador consegue ler ou interpretar uma hiperligação, identificando o sentido semântico dessa descrição. Contudo, uma máquina não consegue deduzir qual o significado dessa hiperligação. Por exemplo, dois cursos de e-Learning podem fazer referência a um determinado livro. Um deles lista o livro como leitura obrigatória e o outro lista-o como leitura opcional. Para os alunos ou formandos, o sentido é totalmente diferente, mas para a máquina não, uma vez que encara ambos como simples hiperligações. Neste contexto, uma relação entre dois recursos possui uma propriedade que permite atribuir significado à ligação. No exemplo referido, podemos atribuir a propriedade "está entre os livros obrigatórios" ao URL no primeiro e-curso e a propriedade "está entre os

livros opcionais" ao URL no segundo e-curso. Com a XML e a RDF o conhecimento passa a ser formalizado de uma forma estruturada.

A RDF permite a automatização do processamento de recursos, facilitando a troca de conhecimento via Web (Klyne e Carroll, 2004). Para tal, recorremos aos URIs, associando-os aos elementos da RDF, tal como se usa uma ligação numa página Web. Os recursos identificados por URIs possuem propriedades ou atributos. As propriedades que os descrevem correspondem a características, atributos, relações e respectivos valores.

Um recurso pode ser praticamente qualquer coisa (física ou lógica) desde que seja identificável por um URI: um Website ou documento composto por várias páginas Web, uma página Web simples, parte de uma página Web, uma tabela, uma base de dados, uma imagem, um vídeo ou outro qualquer objecto de informação acerca de um conceito bem definido. Portanto, cada recurso deve ser identificado por um identificador de recursos (URI) que assegura que os objectos na Web estejam relacionados a apenas um conceito.

A RDF é uma linguagem que permite a codificação, o intercâmbio e a reutilização de metadados estruturados (Daconta et al., 2003). Basicamente, a RDF fornece uma forma de associar propriedades a recursos, através de *statements*. Um *statement* (declaração) tem três componentes: *resource* (recurso), *property* (propriedade) e *value* (valor), através dos quais é possível descrever os recursos de informação disseminados na rede, definindo recursos e identificando as suas propriedades e valores.

Uma propriedade é qualquer característica que possa descrever um recurso. Mas uma propriedade pode representar também o relacionamento entre recursos. Para definir o significado, as características e as relações entre recursos recorremos ao Esquema RDF (*RDF Schema* ou RDFS), tal como veremos mais adiante.

Uma declaração corresponde a um determinado recurso, às suas propriedades e aos respectivos valores. Podemos resumir uma declaração como "o recurso (*subject*) possui a propriedade (*predicate*) com o valor (*object*). Por exemplo, a declaração <"http://www.vgportal.ipb.pt", "autor", "Vitor Gonçalves"> teria o significado: A página http://www.vgportal.ipb.pt (recurso) tem como autor (propriedade) Vitor Gonçalves (valor).

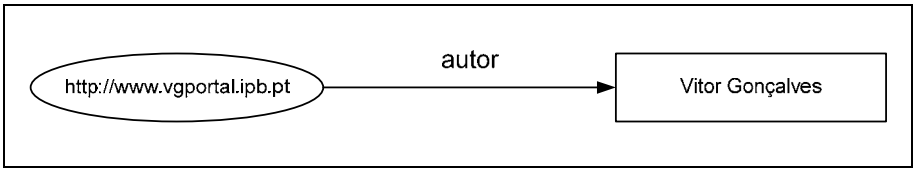


Figura 32 – Grafo de um Modelo de Dados RDF básico

As declarações representam um grafo que vai do nó sujeito para o nó objecto e o arco tem o nome da propriedade, tal como se pode verificar na representação gráfica da declaração inerente ao exemplo. Para apoiar o desenvolvimento, visualização e navegação gráfica em modelos RDF podemos recorrer a ferramentas, tais como DLG (*Directed Label Graphs*) ou IsaViz (ver Figura 33). O modelo VSW (*Visual Semantic Web*), que é uma extensão da metodologia OPM (*Object-Process Methodology*), permite representar o conhecimento estático e dinâmico, simultaneamente em modo gráfico e textual, através dos componentes OPDs (*Object-Process Diagrams*) e OPL (*Object-Process Language*), respectivamente (Dori, 2003).

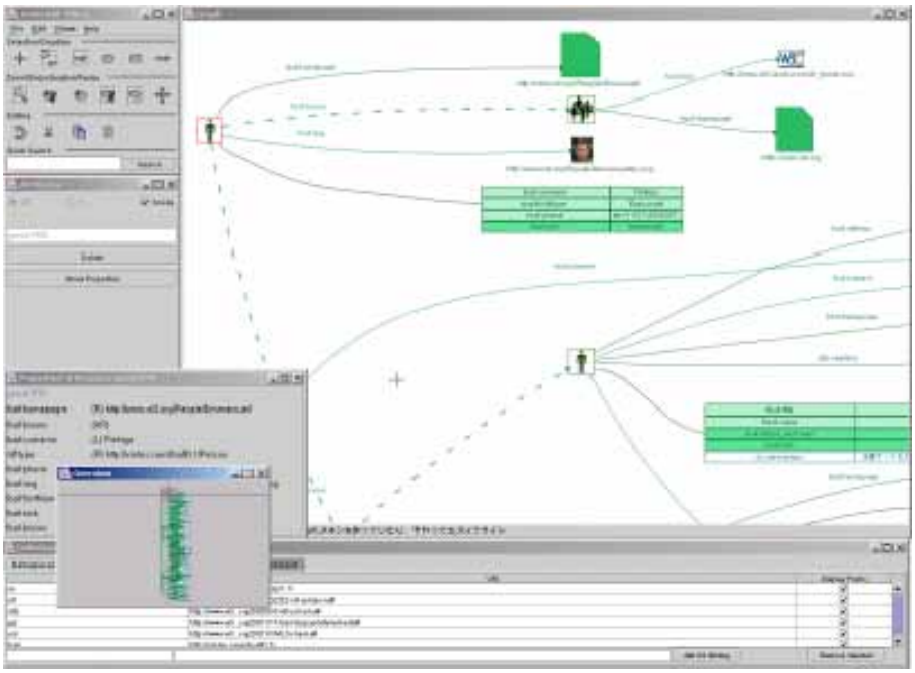


Figura 33 – Ambiente gráfico do IsaViz

Se aplicarmos a sintaxe RDF baseada em XML ao grafo da Figura 32 obtemos o seguinte código:

```
<?xml:namespace ns = http://www.w3.org/RDF/RDF/" prefix="rdf" ?>
<?xml:namespace ns = http://purl.org/dc/elements/1.1/" prefix="dc" ?>

<rdf: RDF>

     <rdf:Description about="http://www.vgportal.ipb.pt">
         <dc:Creator>Vitor Gonçalves</dc:Creator>
     </rdf:Description>

</rdf: RDF>
```

Figura 34 – Exemplo básico de RDF

O elemento *Description* permite agrupar diversos *statements* (declarações) para o mesmo recurso. O atributo *about* deste elemento indica o recurso ao qual todas essas declarações se aplicam. O elemento *dc:Creator* é um elemento da especificação para metadados *Dublin Core* e permite declarar o valor "Vitor Gonçalves". Caso pretendêssemos fornecer o título do URI deste exemplo, bastaria recorrer ao elemento *dc:Title* de metadados *Dublin Core*, tal como veremos na secção seguinte. Para além da sintaxe XML, o código acima necessita das facilidades oferecidas pelos XML *namespaces*, que mais não são do que conjuntos de nomes, identificados por URIs, para usar em documentos XML como nomes de atributos e tipos de elementos (Bray et al., 1999).

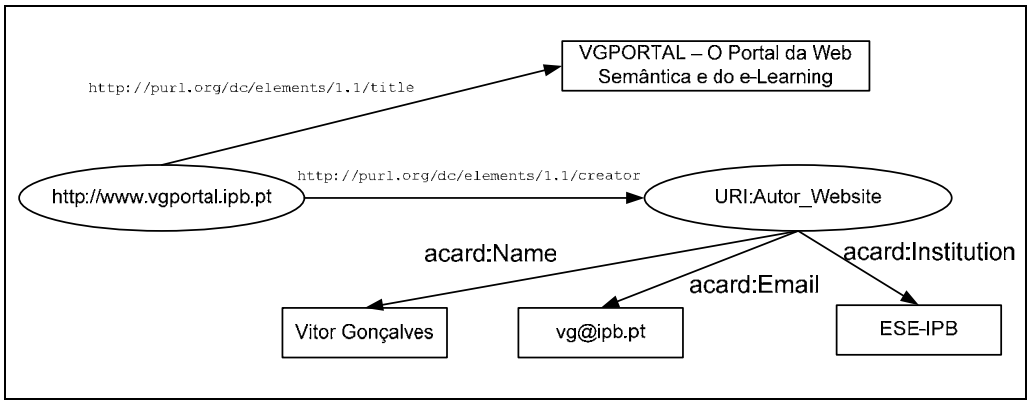


Figura 35 – Grafo de um Modelo de Dados RDF

Num diagrama de modelo de dados, um objecto pode corresponder a outro recurso identificado por um URI, uma *string* ou outro qualquer tipo de dados definido em XML. Por exemplo, se necessitarmos fornecer mais informação sobre o autor (e-mail e instituição), o valor "Vitor Gonçalves" teria que ser substituído por um recurso (identificado por um URI) com as propriedades (nome, e-mail e instituição). A diferença é que, no primeiro exemplo, temos uma declaração com uma única propriedade (o literal "Vitor Gonçalves") e no segundo, passou a existir uma propriedade estruturada que possui três propriedades (nome, email e instituição). Assim, o grafo do modelo de dados da Figura 35 expressa os metadados através da sintaxe de serialização RDF/XML da Figura 36.

Por conseguinte, as *tags* RDF de uma página Web contêm recursos que descrevem uma ou mais propriedades para cada recurso. Cada propriedade possui um valor ou aponta para outro recurso. Assim, as máquinas sabem como usar o valor armazenado em cada propriedade, uma vez que se encontra definido publicamente e armazenado através dos denominados *namespaces*. De referir que o prefixo *acard* corresponde a um *namespace* específico definido uma única vez na terceira linha do código RDF/XML.

```
<?xml:namespace ns = http://www.w3.org/RDF/RDF/" prefix="rdf" ?>
<?xml:namespace ns = http://purl.org/dc/elements/1.1/" prefix="dc" ?>
<?xml:namespace ns = "http://www.vgportal.ipb.pt/acard/" prefix="acard" ?>
<rdf:RDF>
     <rdf:Description about="http://www.vgportal.ipb.pt">
         <dc:Title>VGportal – O Portal da Web Semântica e do e-Learning</dc:Title>
         <dc:Creator>Vitor Gonçalves</dc:Creator>
     </rdf:Description>
     <rdf:Description ID = "http://www.vgportal.ipb.pt/vg">
         <acard:Name> Vitor Gonçalves </acard:Name>
         <acard:Email> vg@ipb.pt </acard:Email >
         <acard:Institution> ESE-IPB </acard:Institution>
     </rdf:Description>
</rdf:RDF>
```

Figura 36 – Exemplo de RDF

Mas, mesmo assim, há a necessidade de especificar que deverá ser indiferente usar como identificação "Gonçalves, Vitor", "Vitor Gonçalves" ou "Vitor Barrigão Gonçalves", pois o mecanismo de busca interpretá-los-á correctamente, já que cada propriedade corresponde a um tipo definido globalmente. Um exemplo similar é a confusão entre datas do tipo UK (dia/mês/ano) e do tipo US (mês/dia/ano) (Thompson, 2004).

A resolução das ambiguidades e a obtenção de maior precisão na recuperação de informação disponível na Web constituem duas das principais preocupações dos projectos da WS. A ambiguidade pode ser solucionada atribuindo-se um URI diferente a cada conceito de uma palavra. Assim, os agentes dos motores de busca poderão encontrar páginas que se refiram a conceitos específicos e não todas as páginas nas quais a palavra ambígua é utilizada. Mas durante o processo de busca, o agente também pode encontrar dois ou mais URIs diferentes para o mesmo conceito. Para resolver este problema, existem as ontologias que vão fornecer o vocabulário necessário para a comunicação entre os agentes computacionais e as páginas Web e mostrar as relações entre os conceitos, fornecendo mais significado que as descrições RDF/XML. Não obstante, podem ocorrer outros tipos de ambiguidades no conteúdo de documentos disponíveis na Web que interfiram na precisão da recuperação da informação.

A RDF não fornece a informação suficiente para que uma máquina (aplicação ou agente de software) perceba que o valor de uma propriedade que representa um autor é uma referência a uma pessoa e não a uma instituição, localidade ou veículo. Necessitamos de um esquema que valide as afirmações expressas em RDF. Necessitamos do esquema RDFS ou de outra proposta ontológica. Estas serão detalhadas no capítulo seguinte.

Genericamente, o RDF *Core* (termo utilizado para distinguir o núcleo RDF do esquema RDFS) define como descrever as propriedades e os valores dos recursos, enquanto que o RDF *Schema* define as propriedades que podem ser utilizadas para definir esquemas. Em suma, podemos afirmar que o XML e o RDF(S) está para os computadores assim como o HTML está para as pessoas.

Finalmente, destacamos algumas das aplicações mais relevantes da tecnologia RDF:

- Tecnologias para a distribuição de conteúdos de notícias *on-line*: RSS (*RDF Site Summary*); PRISM (*Publishing Requirements for Industry Standard Metadata*); e XMLNews-Meta (*XML and News packaging and RDF metadata format*);
- Tecnologias para redes sociais: FOAF (*Friend Of A Friend*) que permite criar páginas legíveis pelas máquinas (*machine-readable*) para descrever as pessoas, as ligações entre elas e as coisas que criam e fazem; FOAFCorp (*Corporate Friends of Friends*) que é uma extensão de FOAF para descrever com maior detalhe a estrutura de redes sociais e as relações entre entidades organizacionais; e *Client – Haystack* que permite a gestão de informação de uma forma personalizada através de um modelo RDF;
- Tecnologias para descrever colecções pessoais de música, vídeo, fotografias e agendas, tais como: *MusicBrains Metadata Initiative* (intercâmbio de áudio e vídeo através de metadados expressos em RDF); RDFPics (descrever e recuperar imagens usando RDF e http); e vCard (representação do formato *vCard* em RDF);
- Tecnologias para descrever documentos, ficheiros ou outros recursos de informação, tal como o XMP (*eXtensible Metadata Platform*) que nos permite adicionar metadados a ficheiros.

Com vista a facilitar a validação do código RDF, o W3C disponibiliza também um serviço *on-line* denominado RDF *Validation Service* (*http://www.w3.org/RDF/Validator/*).

Estes são apenas alguns exemplos que demonstram as potencialidades da tecnologia RDF. Contudo, a listagem de aplicações ao nível de projectos *open source* e proprietários cresce de dia para dia.

3.3.4- Interoperabilidade de Esquemas de Metadados

As metalinguagens semânticas permitem expressar, distribuir e partilhar formalmente o conhecimento necessário à interpretação e utilização inteligente da informação, independentemente dos receptores da informação serem humanos ou máquinas.

Normalmente, os repositórios de objectos de aprendizagem que entretanto foram criados não usavam os esquemas de metadados na sua totalidade ou usavam esquemas de metadados próprios. Os principais motivos eram a complexidade da utilização de todos os elementos que compõem um esquema e os requisitos inerentes à língua e à cultura da comunidade. Consoante as necessidades específicas de cada comunidade, eram escolhidos os elementos e respectivos valores por omissão que melhor se adequavam ao tipo de descrição que se pretendia fazer dos recursos de informação ou objectos de aprendizagem. Estes perfis de metadados (*metadata profiles*) são usados para adaptar especificações de metadados às exigências da comunidade local (Duval et al., 2002). Alguns exemplos desses perfis são: *CanCore Metadata Initiative* (CanCore, 2005), *Singapore eLearning Framework* (SingCore, 2001), *ARIADNE* (ARIADNE, 2004) e diversos *Dublin Core application profile(s)* (DCMI, 2002).

Uma vez que as normas ou especificações que mais se destacaram foram a DCM para os recursos de informação em geral e o LOM para os objectos de aprendizagem em particular, há necessidade de ligar esses perfis de metadados com a especificação que melhor se adeqúe. Com vista a aumentar a interoperabilidade entre os esquemas de metadados, surgiram algumas propostas. Uma vez que as abordagens existentes para mapear perfis de metadados em especificações são muito parecidas, destacamos a abordagem de Najjar et al. (2003) que usa XSLT (*XSL Transformations*) para transformar ARIADNE XML em IEEE LOM XML. Genericamente, a transformação XSL baseia-se numa parte do conjunto de especificações LOM: IEEE LOM P1484.12.3, tal como se pode verificar na Figura 37. Finalmente, é necessário validar as instâncias de LOM XML resultantes.

Neste últimos anos, as entidades oficiais mencionadas uniram esforços no sentido de garantir a compatibilidade e interoperabilidade entre DCM e LOM face aos requisitos da WS.

Os resultados destes esforços são importantes, já que o principal objectivo de uma arquitectura de metadados é representar e dar suporte a uma grande variedade de esquemas de metadados num ambiente distribuído, permitindo a interoperabilidade nos níveis sintáctico, estrutural e semântico (Cunha, 2002).

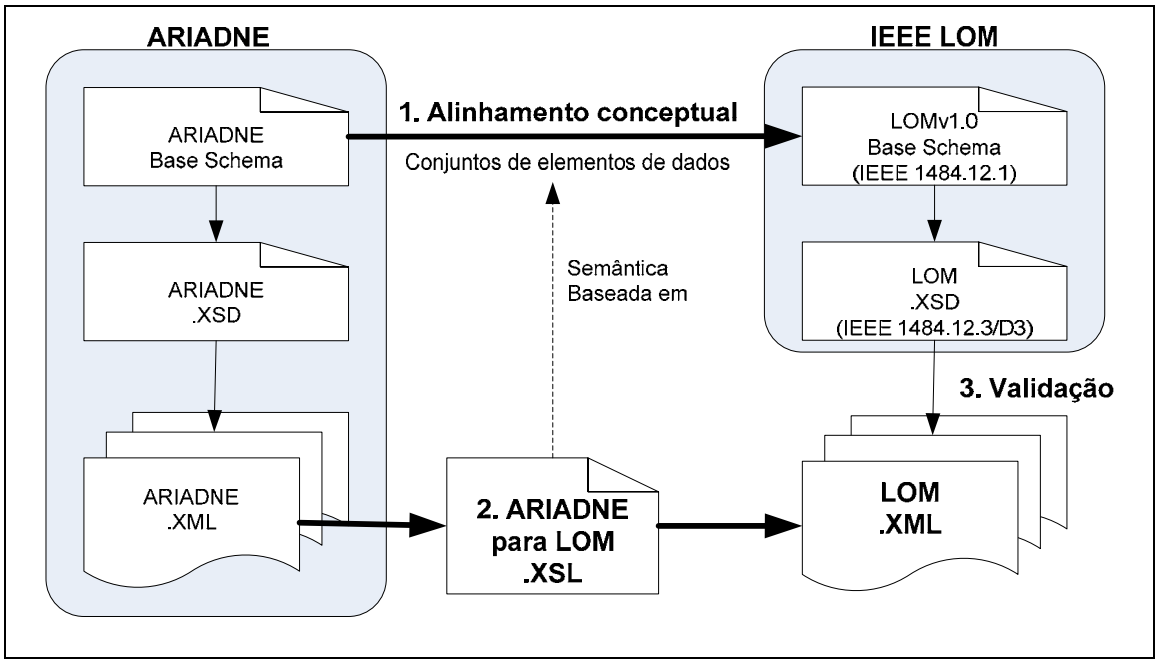


Figura 37 – Uma abordagem para gerar instâncias de metadados interoperáveis

Concluídas as especificações para expressar DCM e LOM em XML, nestes últimos anos têm vindo a desenvolver-se esforços no sentido de que as propostas para expressar DCM e LOM em RDF venham a constituir recomendações definitivas. Referimo-nos mais concretamente às propostas: *Draft IEEE Learning Object Metadata RDF binding* (IEEE LTSC P1484.12.4) proposta por Mikael Nilsson (LOMWG, 2002; Nilsson et al., 2003) e *Expressing Qualified Dublin Core in RDF/XML* proposta por Stefan Kokkelink e Roland Schwänzl (2002). De referir que a proposta *Expressing Simple Dublin Core in RDF/XML* submetida por Dave Beckett, Eric Miller, Dan Brickley (2002) é já uma recomendação definitiva.

Embora a normalização da tradução de DCM e LOM para XML tenha sido relativamente rápida e consensual e as propostas para expressar DCM e LOM em RDF estejam bem encaminhas, o mesmo não se pode dizer da ligação entre DCM e LOM.

Os elementos LOM não podem ser facilmente combinados com os elementos DCM, uma vez que os seus modelos são claramente diferentes. O modelo LOM, detalhado na especificação *LOM Data Model Standard* (IEEE LTSC P1484.12.1), é hierárquico e o modelo DCM, especificado na recomendação *DCMI Abstract Model*, é horizontal (*flat*). Portanto, a possibilidade mais viável de combinar os dois modelos será através das instâncias DC RDF e LOM RDF (Keynes, 2005; Johnston & Powell, 2006).

A crescente procura de definições de interoperabilidade impulsionou a criação do grupo *Joint DCMI/IEEE LTSC Taskforce* com vista a produzir uma recomendação para ligar os esquemas de metadados DCM e LOM (*Recommendation for using IEEE LOM Elements in Dublin Core Metadata*). Esta recomendação resolverá parte da situação descrevendo como usar IEEE LOM e Dublin Core conjuntamente em instâncias de metadados Dublin Core (DCMLTSCTaskforce, 2005).

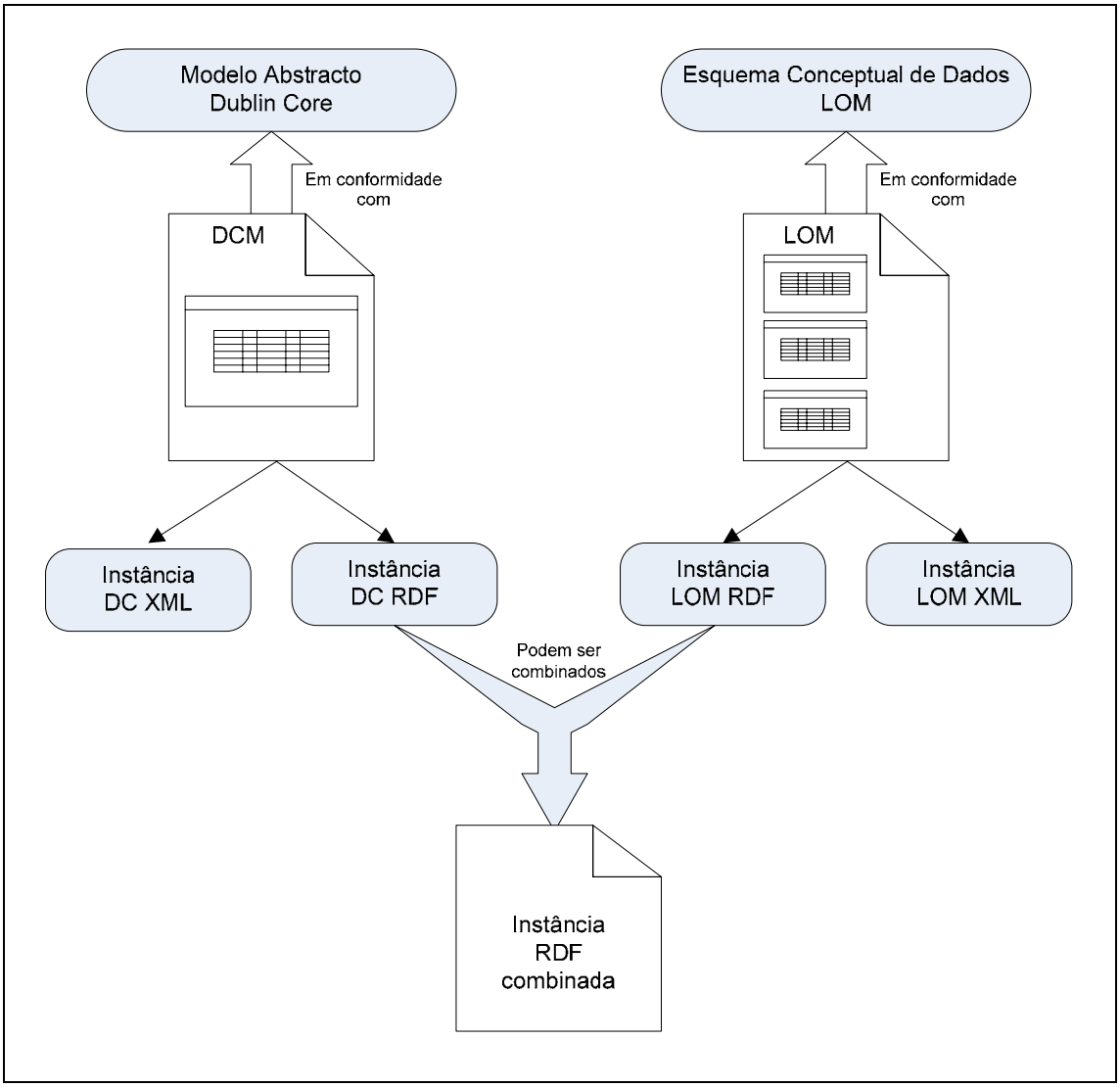


Figura 38 – Modelo para combinar DCM e LOM através de RDF

Consequentemente, o alinhamento das tecnologias para a Web Semântica com as tecnologias para os metadados aplicadas ao e-Learning impulsionará o aparecimento de novos modelos, arquitecturas, infra-estruturas e tecnologias para o e-Learning (Naeve et al., 2005).

3.4- Resumo

A primeira geração da Internet preocupou-se com a implementação da infra-estrutura tecnológica da rede e dos serviços aplicacionais (WWW, FTP, Email, etc.). A segunda incidiu essencialmente na construção de aplicações para a Web. Originalmente, a Web foi criada com o intuito de evoluir no sentido de um espaço onde pessoas e agentes de software pudessem comunicar e colaborar, interpretando informação e gerando conhecimento. Contudo, esta visão ainda não foi alcançada, já que os motores de busca actuais são incapazes de analisar semanticamente a informação e, consequentemente, as pessoas sentem uma crescente dificuldade em obter informação filtrada e de acordo com os seus requisitos. Assim, a terceira geração aposta na evolução de uma Web de documentos interligados através de *links* para uma Web de Dados que relaciona conceitos para permitir realizar inferências semânticas sobre os dados. Esta iniciativa foi denominada de Web Semântica.

A arquitectura para a Web Semântica pode resumir-se em quatro camadas principais: Camada Estrutural, Camada Sintáctica, Camada Semântica e Camada Lógica. Este capítulo incidiu sobre as tecnologias para implementar as primeiras duas camadas: XML(S) fornece a sintaxe e a estrutura que permitem estruturar os dados; os esquemas de metadados, tais como DCM e LOM, permitem descrever esses dados; e a RDF(S) permite expressar essas descrições ou metadados.

A HTML não oferece qualquer descrição semântica acerca dos dados nem das hiperligações entre eles, apenas descreve como os dados apresentados na linguagem natural devem ser exibidos. Com XML garantimos não só a separação entre os rótulos de descrição e de apresentação do conteúdo, mas também a sintaxe para o intercâmbio de informação.

Genericamente, os metadados são utilizados para descrever as características de recursos e as suas relações. A tecnologia RDF assume particular importância na arquitectura da WS uma vez que, usando a notação XML, permite a codificação, o intercâmbio e a reutilização de metadados, integrando as especificações de metadados existentes se conveniente.

Embora a RDF forneça um modelo para expressar metadados associando propriedades a recursos, isso não é suficiente para se entender o que um recurso representa, ou seja, necessitamos de associar mais significado à descrição RDF. Em suma, necessitamos de mais uma camada: a Camada Semântica.

4- Camadas Semântica e Lógica da Web Semântica

A tecnologia *Resource Description Framework* (RDF) constitui a camada fulcral da arquitectura para a Web Semântica. O capítulo III incidiu sobre esta tecnologia, bem como sobre as restantes tecnologias das camadas mais baixas dessa arquitectura. O presente capítulo aborda as tecnologias das camadas superiores, nomeadamente a Camada OWL e a Camada Regras que fornecem a semântica e lógica necessárias à realização de inferências.

Primeiramente, são apresentadas as estruturas convencionais de representação do conhecimento e as tecnologias para as ontologias como forma de perceber universalmente o significado da informação e incrementar a expressividade semântica fornecida pelas tecnologias para as camadas mais baixas da arquitectura para a WS.

Posteriormente, são apresentadas as regras e tecnologias para a inferência a fim de mostrar como é que os agentes inteligentes poderão entender e raciocinar sobre a informação, de forma a gerar conhecimento e satisfazer os requisitos dos utilizadores.

4.1- Representação do Conhecimento

O conhecimento resulta da interpretação realizada sobre a informação (conjunto de dados estruturados que transmitem algo que o sujeito não conhece). Mas, tal como foi referido no capítulo II, o conhecimento é construído através da interacção das estruturas cognitivas do sujeito (conhecimento que já detém) com a nova informação (quer se trate de dados estruturados em conteúdos ou informação obtida na comunicação com outros indivíduos). Portanto, o novo conhecimento deriva de um processo de interpretação semântico. Cada sujeito recorre às suas estruturas cognitivas para interpretar a nova informação e, consequentemente, para extrair novo conhecimento.

Para automatizar o processo ilustrado na Figura 39, pelo menos parcialmente, há que permitir que os computadores interpretem semanticamente a informação. Para tal, há que traduzir o modelo mental e as suas estruturas de conhecimento num formato útil para as

máquinas. Desde há muito tempo que se reconhece que a representação da informação e do conhecimento é importante nos sistemas de informação e nos respectivos sistemas informáticos, mas tornou-se vital para concretizar os objectivos da WS.

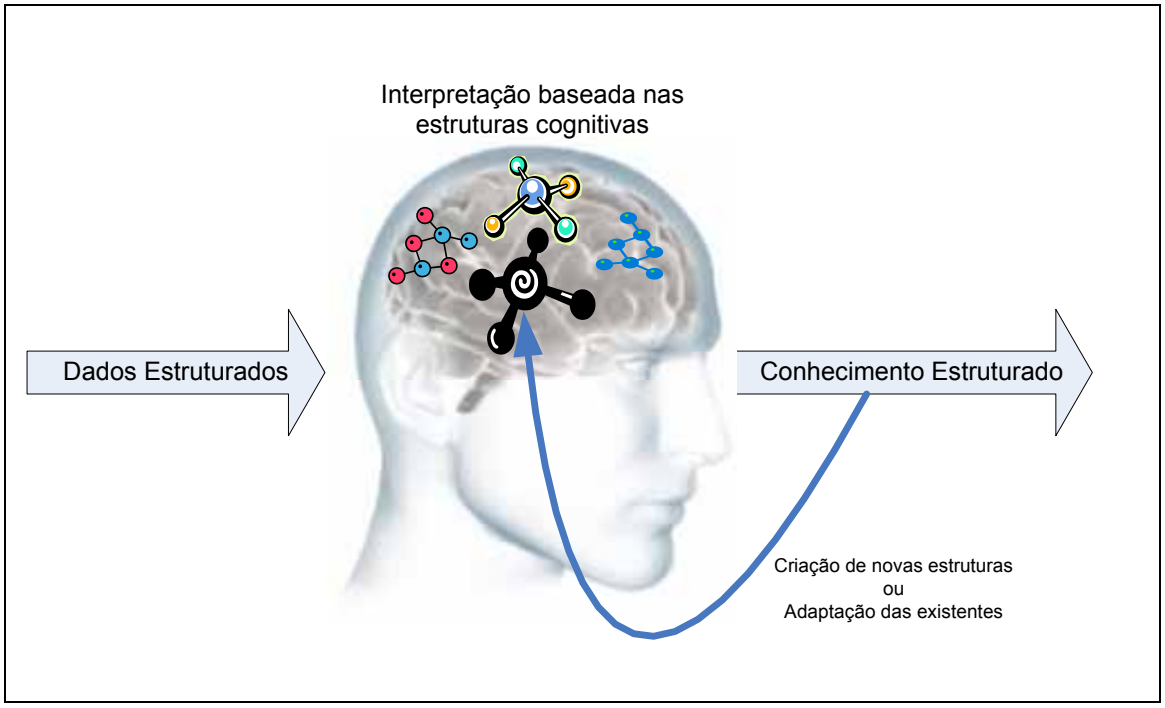


Figura 39 – Modelo mental genérico

Se a ideia é que as máquinas possam apoiar a difusão e recuperação de informação, então o primeiro passo é representar as estruturas cognitivas do modelo mental humano num formato que as máquinas possam manipular.

Ao longo das últimas décadas, diversos tipos de estruturas têm vindo a ser utilizados na estruturação do conhecimento existente num domínio. Os dicionários, glossários e índices constituem exemplos de estruturas que se organizam a partir da utilização de termos. As taxonomias (cabeçalhos de assunto ou esquemas de classificação) correspondem a estruturas que se organizam com a classificação e a criação de categorias. Os tesaurus e as redes semânticas são estruturas que se organizam a partir de conceitos e das suas relações. As ontologias incluem-se neste grupo (Almeida e Bax, 2003).

4.1.1- Tecnologias Convencionais

Tal como referido anteriormente, os metadados são tão úteis para a gestão dos conteúdos como para a descoberta destes após a publicação e, portanto, os metadados tendem, na prática, a ser usados para ambas as finalidades. Analisando os elementos de um esquema de metadados, Garshol (2004) concluiu que:

- O elemento *subject* ou *keyword* (assunto ou palavra-chave) é o elemento mais útil para apoiar a localização e recuperação de recursos de informação na perspectiva da grande maioria dos utilizadores. Note-se que o assunto já era usado com relevo nas fichas ou catálogos impressos (cabeçalho de assunto) inerentes à indexação e classificação de documentos de natureza monográfica;
- Os restantes elementos de metadados são mais úteis no âmbito da gestão e organização dos documentos e na ajuda ao utilizador para refinar a pesquisa ou decidir sobre se os resultados devolvidos na pesquisa correspondem aos seus requisitos;
- O elemento *subject* ou *keyword*, sendo apenas um campo de texto sem restrições (onde qualquer vocabulário é aceite), vê limitadas as suas potencialidades, uma vez que para descrever um determinado assunto podem existir diversas palavras-chave ou assuntos fortemente relacionados. Por exemplo: “mapas de tópicos”, “mapas de navegação em tópicos” (antigo nome para mapas de tópicos), “XML Topic Maps” (um formato para expressar mapas de tópicos em XML, frequentemente usado como sinónimo de mapas de tópicos), “XTM” (o acrónimo para a linguagem ou formato), etc.

Face à dificuldade em descrever precisamente o assunto dos objectos de informação, o uso de instrumentos de controlo terminológico para apoio à indexação e recuperação da informação torna-se imperativo. Ao longo do último século surgiram, evolutiva e cumulativamente, várias abordagens de classificação baseadas no assunto (*subject-based classification*): Vocabulário Controlado (*Controlled Vocabulary*), Taxonomia (*Taxonomy*), Tesauro (*Thesaurus*) e Classificação Facetada (*Faceted Classification*). Vários exemplos de aplicação destas abordagens podem ser encontrados em Steffen Staab e Rudi Studer (2004), nomeadamente no âmbito da categoria *Classification* da especificação LOM (Brase e Nejdl, 2004).

O objectivo destas abordagens não é descrever os objectos de informação (esse objectivo é dos metadados), mas sim descrever os assuntos (*subjects* ou *keywords*) que serão

usados nos metadados para os descrever. Genericamente, as classificações baseadas no assunto são formas de classificação dos conteúdos que agrupam os objectos de informação pelos assuntos que lhe são próximos ou que melhor os descrevem.

Embora não seja objectivo deste capítulo analisar detalhadamente os principais instrumentos para auxiliar a indexação e recuperação de informação, convém esclarecer sucintamente cada um deles para contextualizar as abordagens mais recentes: Ontologias (*Ontologies*) e Mapas de Tópicos (*Topic Maps*).

Embora diferentes, todas estas abordagens de classificação são similares nos seguintes aspectos (Librelotto, 2005):

- São abordagens para estruturar, classificar, modelar e representar conceitos e relações pertencentes a algum assunto de interesse significativo para uma determinada comunidade;
- Permitem que uma comunidade adopte e use o mesmo conjunto de termos de um modo uniforme;
- O significado dos termos é especificado de alguma maneira a um certo nível.

4.1.1.1- Dicionários e Índices

Genericamente, os **dicionários** são listas alfabéticas de termos ou conceitos e do vocabulário explicativo dos mesmos. Os **glossários** são um tipo de dicionários usados para fornecer a explicação de termos pouco conhecidos ou de termos técnicos.

Um dicionário também pode ser usado no âmbito do desenvolvimento de software para documentar as fases de identificação e especificação de requisitos com vista a definir e esclarecer os termos, melhorar a comunicação e evitar mal-entendidos.

Embora os dicionários tenham referências adicionais, tais como: “m.q. (mesmo que), “veja (*see*)” ou “veja também (*see also*)”; não têm apontadores para ocorrências de conceitos, tal como os índices.

Os **índices** são listas alfabéticas dos termos relevantes de um documento e da indicação dos locais onde esses termos aparecem. Por exemplo, num índice remissivo de um livro, cada conceito é associado com a indicação das páginas onde o mesmo foi referido.

4.1.1.2- Vocabulários Controlados

Os vocabulários controlados são constituídos por termos (nomes específicos para conceitos específicos) para usar na indexação ou assuntos para utilizar na classificação. Assim, evita-se que os autores definam termos de fraco significado, termos demasiado gerais ou termos demasiado restritos e impede-se que escolham formas ligeiramente diferentes do mesmo termo ou que dupliquem os termos. Por exemplo, evita-se que os autores usem "mapas de navegação em tópicos" ou "XTM" (ou mesmo termos incorrectamente definidos, tal como "mapas de topiocs"), forçando-os a usar "mapas de tópicos". Ou seja, um vocabulário controlado evita ambiguidades quanto à designação dos termos e conceitos. Alguns exemplos de vocabulários controlados são: LCSH (*Library of Congress Subject Headings*), MESH (*Medical Subject Headings*) e ERIC (*Education Resources Information Center*) *descriptors*.

Convém frisar que os vocabulários controlados (conjunto de termos para a indexação) nada têm a ver com os vocabulários de metadados (*tokens* ou propriedades sugeridas para descrever os objectos de informação) recomendados para alguns elementos dos esquemas LOM e DCM.

4.1.1.3- Taxonomias

As taxonomias (também designadas por taxionomias ou taxinomias) correspondem a classificações hierárquicas de termos. As taxonomias permitem hierarquizar os vocábulos ou termos dos vocabulários controlados, permitindo que os termos relacionados sejam agrupados e categorizados sistematicamente para que seja mais fácil encontrar o termo que melhor se adequa à descrição de um determinado objecto de informação. Uma das taxonomias mais conhecidas é a taxonomia dos organismos vivos. A Figura 40 ilustra parcialmente uma taxonomia que organiza hierarquicamente os conceitos que compõem o domínio dos seres vivos em diferentes classes.

As relações são definidas através de ligações entre os conceitos. Os conceitos superiores (classes) têm subclasses que herdam as suas propriedades. A partir desta taxonomia podemos afirmar que um humano é racional. Se é racional é mamífero e, se é mamífero é animal, que por sua vez é um ser vivo.

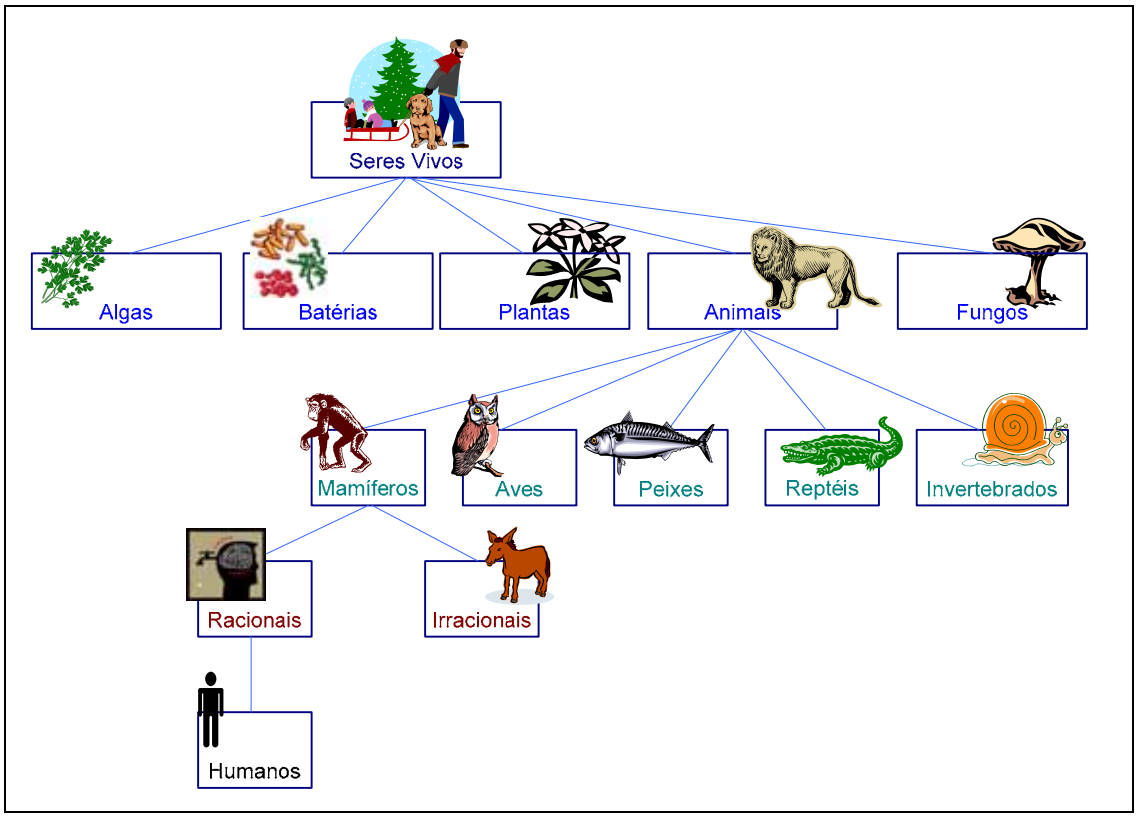


Figura 40 – Exemplo de uma taxonomia para os seres vivos

Para além das taxonomias dos seres vivos (classificação coerente dos animais e vegetais em hierarquias de grupos superiores e subordinados, ou seja, cada reino está subdividido em grupos ou taxa (plural de taxon) sucessivamente mais restritos: Filo, Classe, Ordem, Família, Género e Espécie), podem ser encontrados outros exemplos, tais como: NCBI (*National Center for Biotechnology Information*), *ACM Taxonomy*, Taxonomia dos Objectivos Educacionais (*Taxonomy of Educational Objectives*) ou Taxonomia de Bloom (*Bloom's Taxonomy*), etc.

A Figura 41 ilustra uma hipotética estrutura taxonómica para este capítulo. Esta estrutura permitiria perceber as categorias no âmbito da representação do conhecimento e descrever os assuntos que seriam usados na classificação do objecto de informação.

Contudo, embora a taxonomia seja útil para o autor aquando da criação dos metadados, na descoberta e recuperação de objectos de informação apenas a indicação "Mapas de tópicos" terá utilidade, uma vez que os restantes conceitos não são capturados. Por exemplo: os metadados não incluirão a linha que liga "Mapas de Tópicos" a "Representação do conhecimento", logo essa indicação nunca será considerada. Outras indicações inerentes aos conceitos expressos na taxonomia não capturados podem ser (Garshol, 2004):

- O facto de que “XML Topic Maps” é sinónimo de XTM;
- A diferença entre “XTM” e “Mapas de Tópicos”;
- O facto de que “mapas de navegação em tópicos” é sinónimo de “Mapas de Tópicos”, mas que actualmente já não é usado;
- A relação entre “Mapas de Tópicos” e a Web Semântica;
- A relação entre XTM, XML, HyTM e SGML;
- Entre muitas outras indicações que poderiam enriquecer a descrição do objecto de informação em questão.

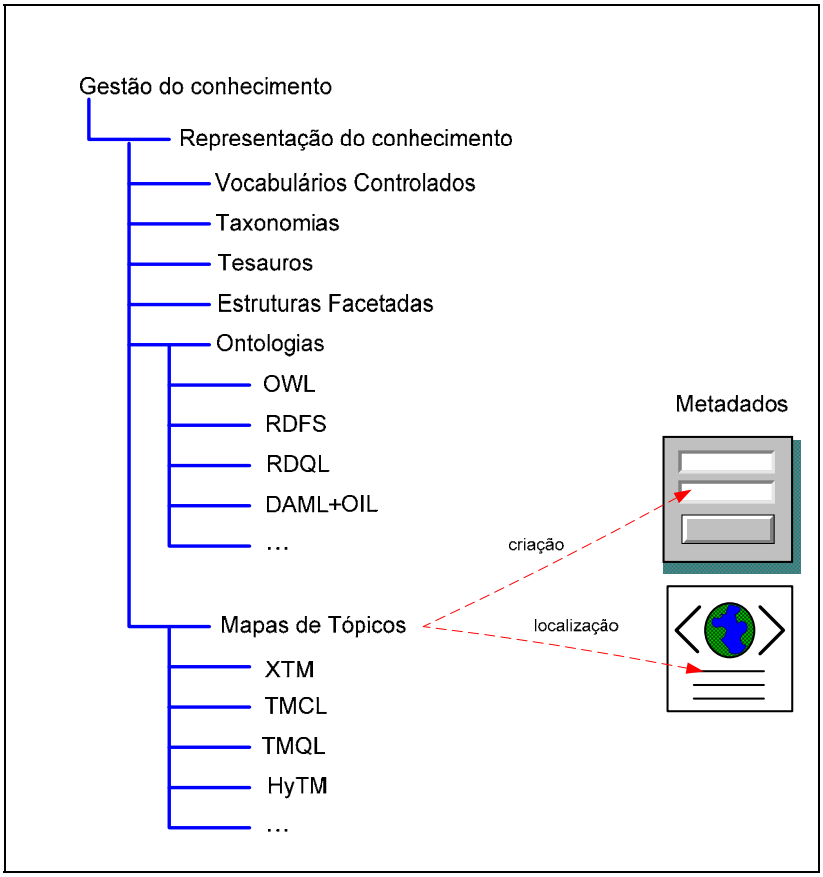


Figura 41 – Exemplo de uma taxonomia para a Gestão do Conhecimento

Em sentido lato e do ponto de vista dos metadados, podemos afirmar que não existe qualquer diferença entre um vocabulário controlado e uma taxonomia (Garshol, 2004).

4.1.1.4- Tesauros

Os tesauros ou *thesauri* são vocabulários controlados e dinâmicos de termos relacionados semântica e genericamente e cobrindo um domínio específico do conhecimento. Pode ser definido, também, segundo a sua função, como um dispositivo de controlo terminológico usado na tradução da linguagem natural dos documentos, dos indexadores ou dos utilizadores numa linguagem do sistema mais restrita (UNESCO, 1973). Existem duas normas para descrever estas estruturas: ISO2788 (tesauro monolingue) e ISO5964 (tesauro multilingue). Os componentes de um tesauro são: termos (descritores que representam conceitos); estrutura (relacionamento entre conceitos representados por termos) e conjuntos de remissivas. A notação usada nestas normas pode resumir-se a:

- Relações hierárquicas: TT (*Top Term*), BT (*Broader Term*) e NT (*Narrower Term*);
- Relações com outros termos: RT (*Related Term*);
- Termos preferenciais: USE // UF (usar preferencialmente um termo em vez de outro);
- Propriedade para esclarecer o contexto através de nota: SN (*Scope Note*).

Os tesauros são um instrumento melhor do que as taxonomias, uma vez que fornecem um vocabulário mais rico para descrever os termos. Por exemplo, na taxonomia da Figura 40 – Exemplo de uma taxonomia para os seres vivos, poderíamos indicar o uso do termo "pessoas" em vez do termo "humanos" através do termo preferencial "USE". Por conseguinte, muitos dos problemas identificados nas taxonomias podem ser resolvidos recorrendo a tesauros. Alguns exemplos de tesauros são: TEE (*Thesaurus* Europeu da Educação); TESE (*Thesaurus* Europeu dos Sistemas Educativos), Tesauro EUROVOC (*Thesaurus* das Comunidades Europeias), Tesauro UNESCO (*United Nations Educational, Scientific and Cultural Organisation Thesaurus*), Tesauro ETB (*European Treasury Browser Thesaurus*), TGN (*Thesaurus of Geographic Names*), *Webster's New World Thesaurus*, *Thesaurus of English Words & Phrases* ou *Roget's Thesaurus*, WordNet (*a lexical database for the English language - dictionary and thesaurus*), etc.

4.1.1.5- Classificações Facetadas

As classificações facetadas derivam da proposta inicial de Ranganathan (também conhecida como *Colon Classification*) que consistia em cinco facetas (*facets*): *Personality*

(Personalidade), *Matter* (Matéria), *Energy* (Energia), *Space* (Espaço) e *Time* (Tempo). Genericamente, as classificações facetadas são estruturas que identificam características comuns a várias categorias de um assunto, organizando-as em diversos aspectos ou facetas. Ou seja, a informação é organizada em compartimentos que associam um atributo a um valor atómico ou complexo. Logo, a descrição dos assuntos é realizada com maior especificidade e, consequentemente, existirá mais do que uma forma de descobrir e recuperar um determinado objecto de informação. Os exemplos mais relevantes de sistemas de classificação facetada são: DDC (*Dewey Decimal Classification*), UDC (*Universal Decimal Classification*), LCC (*Library of Congress Classification*) e NLMC (*National Library of Medicine Classification*).

4.1.2- Tecnologias para a Camada Semântica

As ontologias já são usadas no âmbito da Inteligência Artificial há alguns anos, geralmente associadas à inferência lógica e técnicas similares. Mais recentemente, começaram também a ser usadas para a localização e recuperação de informação no âmbito das Ciências da Computação e das Ciências da Informação (Garshol, 2004).

Em primeira instância, uma ontologia é um modelo de dados que representa um determinado domínio do conhecimento.

Os dicionários fornecem as definições dos conceitos, enquanto que os índices fornecem as indicações para os locais onde cada conceito é referenciado. A taxonomia clarifica os relacionamentos hierárquicos entre os conceitos, criando uma estrutura de classes/subclasses, enquanto que o tesauro amplia esta estrutura com algumas relações predefinidas. Finalmente, uma ontologia descreve os conceitos relevantes existentes num determinado domínio do conhecimento, definindo quaisquer relações binárias entre eles que se julgue interessante citar. Portanto, há uma forte conexão entre os diversos instrumentos de representação do conhecimento, tal como se pode verificar na Figura 42 (Librelotto, 2005).

As ontologias representam o culminar da exposição evolutiva e cumulativa que fizemos dos instrumentos convencionais para a representação do conhecimento. Ao fornecerem um vocabulário aberto para descrever objectos de informação, as ontologias assumem-se como fulcrais no âmbito da Web Semântica, uma vez que as restantes abordagens representam linguagens com vocabulários fixos.

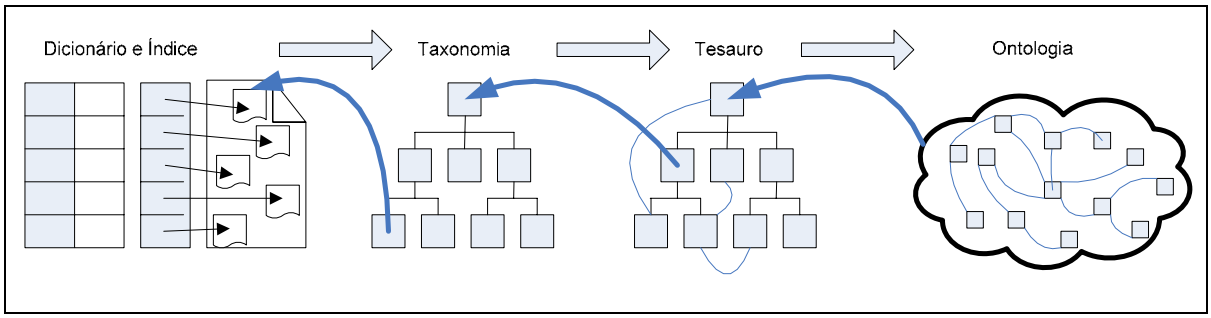


Figura 42 – Relação entre Dicionário, Índice, Taxonomia, Tesauro e Ontologia

Sendo finito o número de palavras, há um número infinito de contextos nos quais as palavras individuais podem aparecer (Bruner, 2000). Um dos conceitos em que assenta a WS está associado à utilização de URIs, cuja principal função é assegurar que os conceitos não sejam apenas palavras num documento, mas que estejam ligados a um significado específico. Podemos usar XML e RDF(S) para dizer, por exemplo, que o termo "aluno" e o termo "estudante" se referem ao mesmo conceito, mas essas linguagens não nos permitem estabelecer relações entre os recursos. Por exemplo, se um agente de software, durante o processo de busca, encontrar dois documentos que usam URIs diferentes para o mesmo conceito, desorientar-se-á. Para resolver este problema, recorremos às ontologias. Elas mostram as relações entre os conceitos e fornecem o vocabulário e estruturas de dados compartilhadas, indispensáveis para a comunicação entre os agentes de software. Uma vez que existem termos que têm significados diversos, as ontologias têm como função esclarecer o significado universalmente aceite para cada um deles. Ou seja, devem fornecer o vocabulário comum e o entendimento compartilhado de um determinado domínio (Berners-Lee et al., 2001). A exploração da WS requer que o foco esteja no conteúdo (ou no significado) e não nas palavras (WISE, 2002). Mas como explicar a uma máquina o significado de democracia se esta pode ser entendida de várias maneiras? Representar e relacionar todos e cada um dos conceitos existentes não é tarefa fácil, uma vez que podem variar de país para país e de contexto para contexto. Por exemplo, na nossa cultura, a cor preta é sinónimo de tristeza, enquanto que noutras culturas simboliza alegria.

Mas, não basta que as máquinas entendam os conteúdos textuais, também devem perceber o significado das imagens, sons, filmes ou outros conteúdos multimédia. Um quadro que seja considerado arte, por exemplo, mostrando mulheres nuas, não pode ser confundido com uma imagem pornográfica. Caso contrário, o material poderia vir a ser censurado sem ter em conta a sua qualidade artística.

Parafraseando Hans-Georg Stork (2002), a Web Semântica abarcará muito conhecimento, mas não todo. Portanto, embora as ontologias se limitem a áreas de interesse, serão numerosas e competirão entre si. Stork defende a multiplicidade de ontologias e assegura que é desejável a confrontação das mesmas.

As ontologias têm vindo a assumir-se cruciais em áreas tais como a representação do conhecimento, processamento da linguagem natural, recuperação de informação, bases de dados, gestão do conhecimento e sistemas multiagente, entre outros. Genericamente, podem ser usadas na marcação explícita de páginas Web e na construção de modelos de domínios (Silva, 2004). Nesta perspectiva, a WS pode ser vista como um grande número de pequenos componentes ontológicos que apontam entre si e que facilitarão o intercâmbio e processamento da informação entre aplicações Web (Hendler, 2001). No entanto, convém salientar que a semântica não reside apenas no conteúdo de um recurso, mas também na forma como este se relaciona com os demais recursos na Web. É essencial, portanto, que os recursos disponibilizados sejam suficientemente expressivos ao nível dos metadados para que as máquinas (ou agentes) possam ser capazes de processar e entender o real significado dos dados, de acordo com as necessidades do utilizador e as fontes de informações disponíveis.

Uma ontologia é uma especificação formal e explícita de uma conceptualização compartilhada (Gruber, 1993). É uma especificação formal e explícita porque a ontologia pode ser manipulada ou processada e o tipo dos conceitos usados e as restrições no seu uso são definidos explicitamente. Uma conceptualização é uma visão abstracta e simplificada do mundo que se deseja representar, independentemente do vocabulário utilizado. Ou seja, é um conjunto de conceitos e das relações entre esses conceitos usados para comunicar num dado domínio. E, por fim, a conceptualização é compartilhada porque o conhecimento que a ontologia reflecte deve ser consensual e amplamente aceite pelas comunidades.

Uma ontologia pode ser definida como um conjunto de conceitos e termos interligados que são usados para descrever uma determinada área do conhecimento ou construir uma representação para o conhecimento (Swartout e Tate, 1999).

Uma ontologia é uma forma de representar o conhecimento através de hierarquias elementares (Hendler, 2001). Pode ser vista como uma taxonomia formada por classes e subclasses de objectos, relacionadas entre si, à qual juntamos mais um conjunto de propriedades e regras de inferência. A declaração das propriedades de cada classe permite especificar o conhecimento inerente aos conceitos de um determinado domínio. A

especificação das regras através de ontologias permite representar explicitamente a semântica dos dados, o que impulsionará o desenvolvimento de uma rede de conhecimento humano, complementando o processamento das máquinas e melhorando qualitativamente os serviços Web. Esta taxonomia permite formar uma estrutura onde propriedades são atribuídas a determinadas classes e os objectos que pertencem a esta classe herdam as suas características.

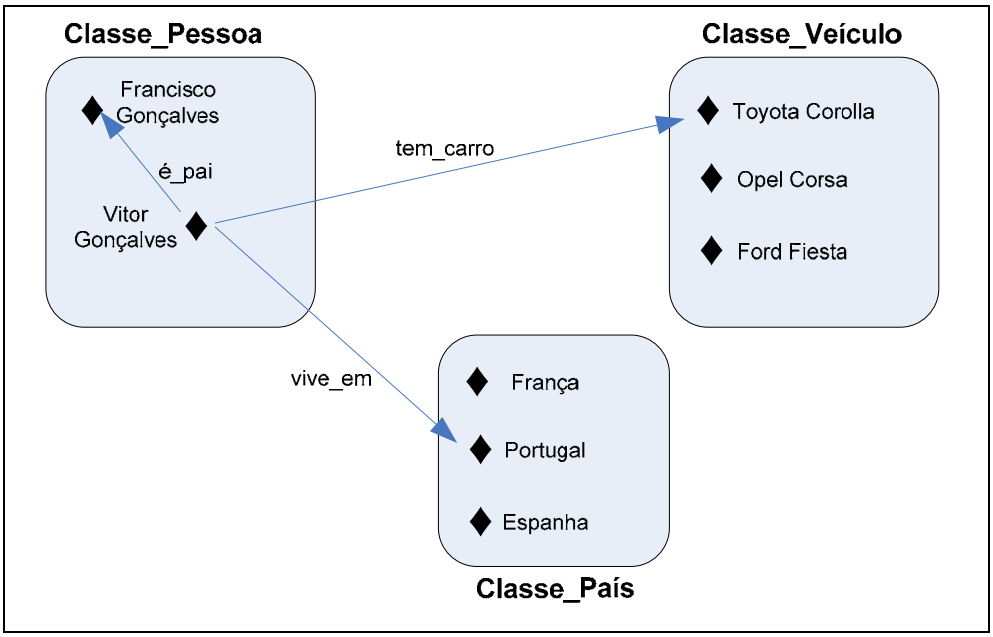


Figura 43 – Componentes básicos de uma ontologia

Genericamente, os componentes básicos de uma ontologia são:

- **Classes**: correspondem a grupos, conjuntos ou colecções de objectos, normalmente organizados numa taxonomia. Por exemplo: classe pessoa, veículo ou país;
- **Relações:** representam os tipos de interacções entre os conceitos de um domínio. Por exemplo: a relação "é_pai" entre duas pessoas;
- **Atributos**: são propriedades, características ou parâmetros que os objectos podem ter e partilhar. Por exemplo, o objecto "Toyota Corolla" da classe "Veículo" poderia ter como atributos: Nome – Toyota, Versão – Corola, Modelo – Hatchback, Numero_Portas – 3, Velocidades – 5, motor – 1.4 D4D, etc;
- **Instâncias**: correspondem às concretizações dos conceitos e relações estabelecidas na ontologia, ou seja, são utilizadas para representar elementos específicos ao nível mais baixo, ou seja, os próprios objectos. Por exemplo: "Toyota Corolla" é membro da classe "Veículo" e "Portugal" é membro da classe "País".

Finalmente, os **axiomas** (proposições lógicas usadas para modelar declarações sempre verdadeiras sobre os conceitos e relações) e as **regras de inferência** (representando a semântica que favorecerá a partilha de conhecimento com e entre os agentes) permitem definir as restrições (Noy e McGuinness, 2001; Almeida e Bax, 2003). Por exemplo, podemos definir que o custo de um curso corresponde à soma do preço dos módulos que o compõem ou que a idade de uma pessoa equivale à diferença entre a data actual e a sua data de nascimento.

A Figura 43 ilustra os principais componentes de uma ontologia destacando três classes. No entanto, cada uma dessas classes poderia por si só constituir uma ontologia. Por exemplo, poderíamos ter uma ontologia para veículos, tal como ilustrado parcialmente na Figura 44.

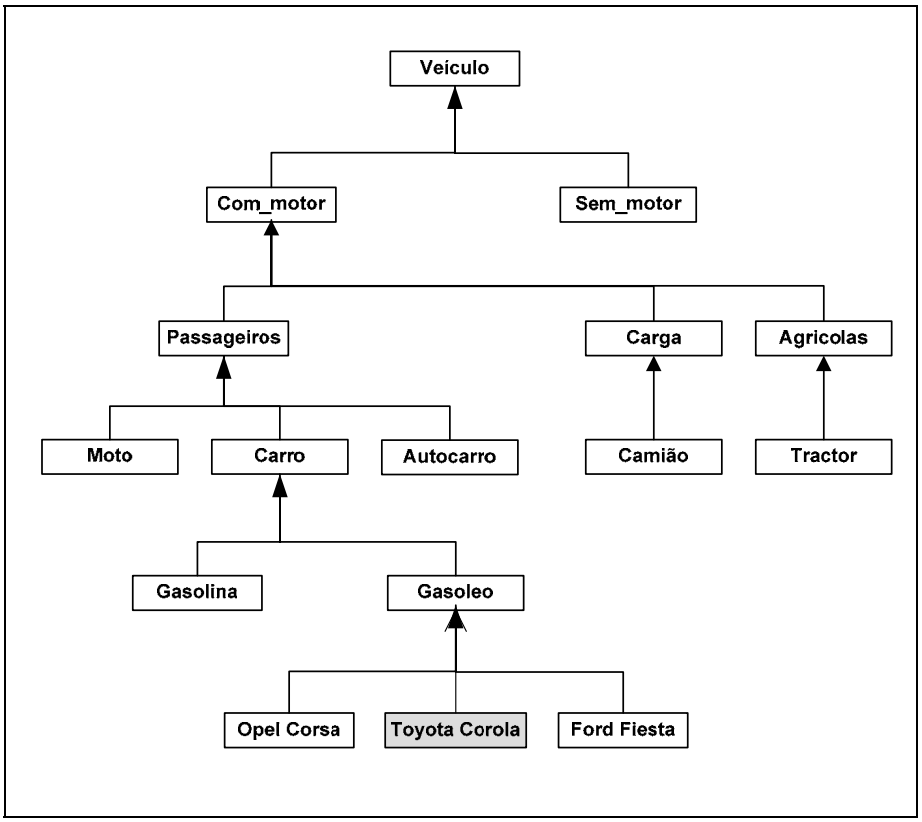


Figura 44 – Modelo conceptual de uma ontologia parcial

Tratando-se de uma representação hierárquica, facilmente se percebe que a relação mais relevante é *is_sub_class_of* (é_subclasse_de) ou, na forma inversa, *is_super_class_of* (é_superclasse_de).

Não existe uma forma correcta única de modelar um domínio (ou de criar uma ontologia), mas sim várias alternativas viáveis. O desenvolvimento de ontologias é necessariamente um processo interactivo (Noy e McGuinness, 2001).

O aparecimento de ontologias ligeiramente diferentes sobre o mesmo domínio poderá não ser prejudicial. Aliás, tal como referido, essa multiplicidade de ontologias e sua confrontação é desejável, uma vez que acaba por promover a integração entre elas (Goñi et al., 2002). Não obstante, parece ser evidente que o uso de ontologias permitirá, por um lado, pesquisas mais rápidas, uma vez que os agentes pesquisarão apenas os documentos que se referem à informação desejada, em vez de todas as páginas Web que contêm as palavras-chave ambíguas, tal como acontece com os mecanismos de busca actuais. E, por outro, auxiliará a comunicação e colaboração efectiva entre as pessoas e as aplicações computacionais.

Imaginemos que procurávamos informação sobre a "Família do Escritor José Saramago" usando, por exemplo, o motor de busca Google; o resultado da pesquisa traduzir-se-ia em alguns milhares de páginas Web (mesmo optimizando a pesquisa com os filtros disponíveis obteríamos algumas centenas de páginas), uma vez que o mecanismo de busca não efectua a procura com base na relação entre conceitos e no significado, mas sim em palavras-chave desprovidas de significado e contexto, tal como já referimos anteriormente.

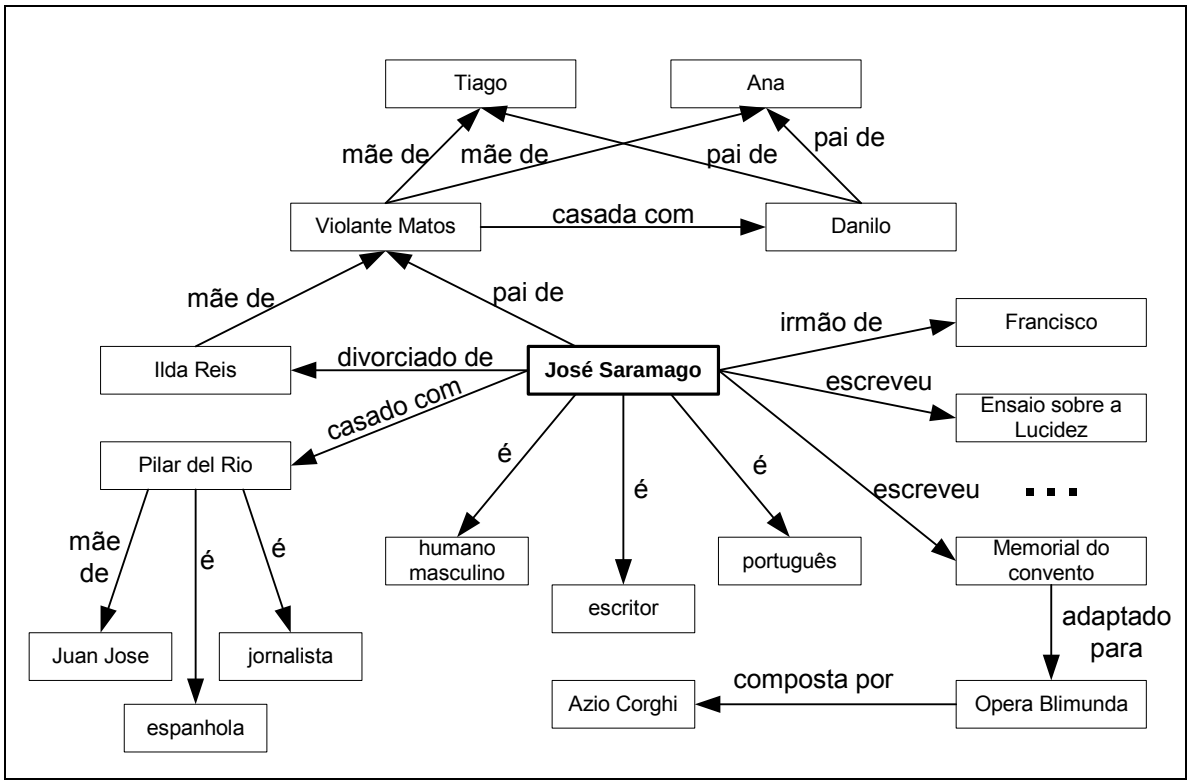


Figura 45 – Mapa conceptual de uma possível ontologia para José Saramago

Os motores de busca podem apresentar resultados satisfatórios na procura de um conceito. Mas, não lidam com a relação entre dois conceitos, não devolvem conhecimento

implícito e têm dificuldade em procurar informação em bases de dados. Ou seja, os agentes de software não distinguem os poetas dos atletas, as cores das flores ou os nomes comuns dos nomes próprios.

Um mecanismo de busca semântico é capaz de percorrer a informação representada em ontologias e responder mais precisamente a questões tais como: "Quem são os netos de José Saramago?"; "Quem é o genro de José Saramago?"; "Qual a nacionalidade da sua actual mulher?" ou mesmo "Qual a sua relação com o compositor italiano Azio Corghi?", uma vez que, após verificar que José Saramago pertence à classe dos humanos, o mecanismo de busca perceberá que a informação que procuramos incide nas suas relações com outros humanos.

Uma ontologia é então uma descrição ou especificação formal dos conceitos e dos relacionamentos que podem existir e que podem ser usados por um agente ou uma comunidade de agentes, e cuja finalidade principal é permitir a descoberta, a partilha e a reutilização do conhecimento (Staab e Studer, 2004). Portanto, uma ontologia relaciona-se com um vocabulário específico, uma linguagem específica e a conceptualização de um determinado domínio.

Nestes últimos anos, têm vindo a ser desenvolvidos diversos esforços no âmbito das metodologias, linguagens e ferramentas para a construção e integração de ontologias para a WS. Para além dos projectos no âmbito do W3C (nomeadamente os dos grupos: *WebOnt Working Group, Semantic Web Deployment Working Group, Semantic Web for Health Care and Life Sciences Interest Group e Semantic Web Interest Group*), destacam-se os contributos dos projectos DAML (DARPA - *Defense Advanced Research Projects Agency*) (DAML, 2002), On-To-Knowledge (*Information Society Technologies Program for Research*) (OTK, 2004), SUMO - *Suggested Upper Merged Ontology* (*IEEE Standard Upper Ontology Working Group*) (SUMO, 2005) e OntoWeb (projecto de intercâmbio de informação para a gestão do conhecimento e o comércio electrónico promovido por várias entidades agrupadas no *OntoWeb Consortium*) (Fernández-López, 2002), entre outros.

As tecnologias RDF (*Resource Description Framework*) e OWL (*Web Ontology Language*) são recomendadas pelo W3C para a WS com vista a fornecer uma estrutura para a gestão de recursos, para a integração de organizações e para a partilha e reutilização dos dados na Web (W3C, 2004).

A RDF fornece um modelo para expressar metadados que associa propriedades a recursos através da sintaxe da RDF/XML. Mas, para se entender o que um recurso representa,

esta informação não é suficiente, necessitamos de associar mais significado à descrição RDF, ou seja, necessitamos de ontologias. A RDF por si só não permite criar ontologias; no entanto, ao permitir descrever vocabulários que representam o conhecimento sob a perspectiva das redes semânticas, constitui a base para outras linguagens com essa finalidade, tais como *RDF Schema* (RDFS) e OWL que mais não são do que tecnologias que definem as primitivas para a criação de ontologias (Davies, 2003).

O modelo de dados RDF(S) pode ser comparado a um diagrama E-R (entidade-relação). Contudo, o modelo RDF não disponibiliza nenhuma forma de declarar as relações entre os recursos Web. Para definir o significado, as propriedades e as respectivas relações com os recursos recorremos ao esquema RDFS (extensão semântica da linguagem RDF). Assim o vocabulário RDF(S) fica definido através de dois *namespaces*: *rdf* e *rdfs*. Portanto, de acordo com as declarações expressas, no início dos documentos RDF deve ser incluído pelo menos um dos dois prefixos abaixo:

```
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
```

4.1.2.1- RDF Schema

Comparando os esquemas RDFS com os DTDs, conclui-se que, ao contrário de um DTD, que especifica um conjunto de regras para a estrutura de um documento XML, um RDFS fornece a informação sobre a interpretação das declarações (*statements*) apresentadas num modelo de dados RDF.

O RDFS define, portanto, as primitivas para a criação de ontologias simples, permitindo que estas possam ser representadas e, assim, possam ser efectuadas sobre as páginas Web inferências e consultas um pouco mais complexas.

As ontologias simples importam algum significado para as descrições RDF, declarando tipos e relacionando os termos utilizados. Contudo, não representam todos os tipos de relações entre os termos, uma vez que as relações importadas são extremamente simples e podem resumir-se à sintaxe:

- ***rdfs:Resource***: a um nível de abstracção mais alto, permite definir a classe genérica. Consideram-se 2 subclasses principais: *rdfs:Class* e *rdf:Property*.
- ***rdfs:Class***: permite definir uma classe;

- ***rdfs:subClassOf***: permite definir subclasses. A definição de subclasses resulta da relação "ser subclasse de" uma vez que é instância de *rdf:Property*.

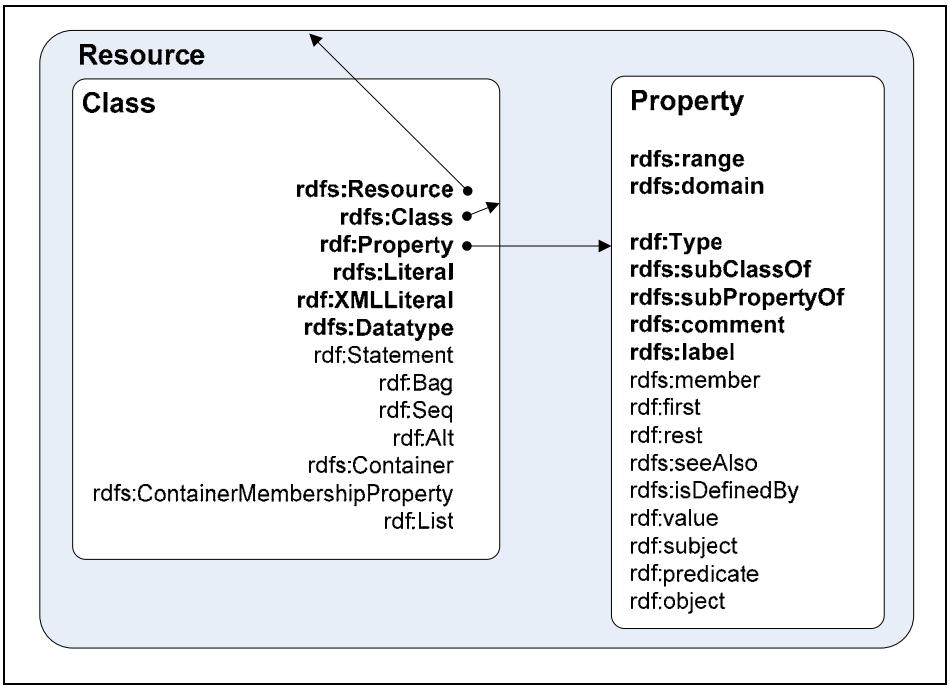


Figura 46 – Classes e propriedades do esquema RDFS

As funcionalidades básicas disponibilizadas por *rdfs:domain* e *rdfs:range* não fornecem nenhuma forma directa de definir restrições às propriedades intrínsecas de uma classe. No entanto, é possível declarar indirectamente essas restrições combinando o uso de *rdfs:domain* e *rdfs:range* com sub-propriedades (ver Figura 47 e Figura 48). O suporte directo para tais declarações é fornecido por linguagens ontológicas mais ricas, tais como OWL.

Em suma, o RDFS permite definir a hierarquia de classes e as suas propriedades, mas não representa o significado dos conceitos nem todas as relações possíveis entre eles. Por estas razões, é considerada uma linguagem para construção de ontologias simplificadas. Exigia-se uma linguagem que permitisse descrever explicitamente um domínio, através da definição:

- dos conceitos;
- das propriedades e atributos desses conceitos;
- das restrições sobre essas propriedades e atributos.

Sobre RDFS têm-se construído linguagens com primitivas mais expressivas e com capacidade de inferência, tais como DAML+OIL e OWL (ver secção seguinte).

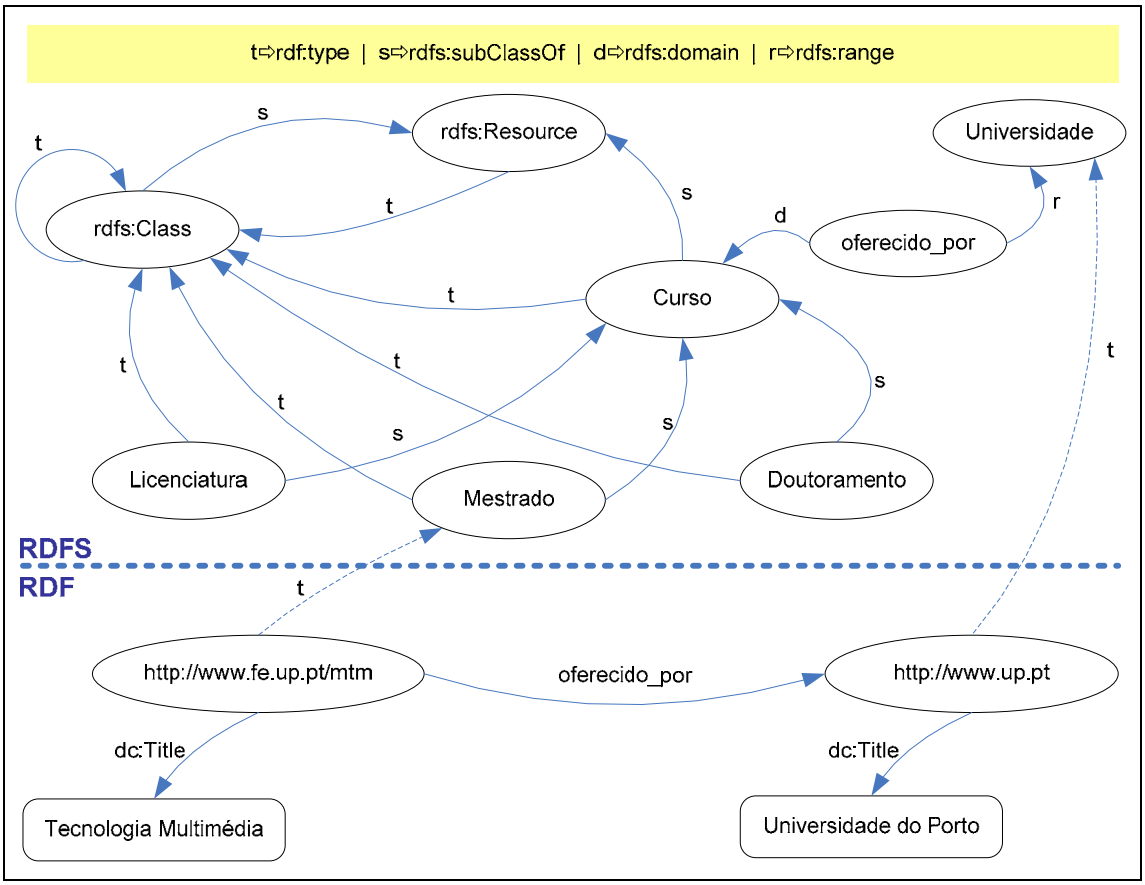


Figura 47 – Um exemplo de aplicação de RDF e RDFS

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22/rdf-syntax-ns#"
         xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#">
 <rdf:Description ID="Curso">
   <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
   <rdfs:subClassOf resource="http://www.w3.org/TR/rdf-schema#Resource"/>
 </rdf:Description>
 <rdf:Description ID="Licenciatura">
   <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
   <rdfs:subClassOf resource="#Curso"/>
 </rdf:Description>
 <rdf:Description ID="Mestrado">
   <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
   <rdfs:subClassOf resource="#Curso"/>
 </rdf:Description>
 <rdf:Description ID="Doutoramento">
   <rdf:type resource="http://www.w3.org/2000/01/rdf-schema#Class"/>
   <rdfs:subClassOf resource="#Curso"/>
 </rdf:Description>
 <rdf:Property rdf:ID="oferecido_por">
   <rdfs:domain rdf:resource="#Curso" />
   <rdfs:range rdf:resource="#Universidade" />
 </rdf:Property>
</rdf>
```

Figura 48 – Um exemplo de aplicação de RDFS

4.1.2.2- Web Ontology Language

OWL é uma linguagem de criação de ontologias para a Web baseada em RDF(S) e DAML+OIL que inclui mais algum vocabulário para descrever propriedades, classes e relações entre classes, entre outros elementos. DAML+OIL é uma linguagem semântica de marcação e de criação e integração de ontologias para os recursos da Web baseada nas características das linguagens DAML (*DARPA Agent Markup Language*) e OIL (*Ontology Inference Layer* ou *Ontology Interchange Language*). DAML (*http://www.daml.org*) é uma linguagem baseada em linguagens lógicas que é usada na especificação de ontologias. A linguagem DAML fornece uma infra-estrutura básica que permite que uma máquina faça inferências simples tal como os seres humanos (DAML, 2002). OIL (*http://www.ontoknowledge.org/oil/*) é uma linguagem que combina as características das linguagens baseadas em *frames* com a semântica formal e os serviços de inferência da lógica descritiva, com o intuito de obter ontologias específicas para a Web compatíveis com RDF(S). De salientar que a proposta DAML+OIL para a camada ontológica da Web permite também o intercâmbio de ontologias.

Um documento OWL inicia-se da seguinte forma:

```
<owl:Ontology rdf:about="uri">
<rdfs:comment>Exemplo de Ontologia</rdfs:comment>
<owl:imports rdf:resource="http://protege.stanford.edu/plugins/owl"/>
<owl:imports rdf:resource="http://purl.org/dc/elements/1.1"/>
<rdfs:label>Ontologia do Sistema de e-Learning da ESEB</rdfs:label>
</owl:Ontology>
```

Figura 49 – Exemplo básico de início de documento OWL

OWL é baseada na lógica descritiva para fornecer uma semântica formal, permite a representação de hierarquias e facilita um maior nível de interpretação dos conteúdos Web por parte da máquina do que aquele que é proporcionado por XML e RDF (ou mesmo RDFS).

As classes e subclasses OWL são definidas através dos elementos *owl:Class* e *rdfs:subClassOf*. No caso do exemplo RDFS da Figura 47, a declaração de uma classe e subclasse em OWL seria:

```
<owl:Class rdf:ID="Curso" />
<owl:Class rdf:ID="Mestrado">
<rdfs:subClassOf rdf:resource="#Curso"/>
...
</owl:Class>
```

Figura 50 – Exemplo de declaração de classes e subclasses em OWL

A declaração das propriedades das classes e subclasses realiza-se com o elemento *owl:ObjectProperty*. Para especificar a que classe pertence determinada propriedade recorre-se conjuntamente ao elemento *rdfs:domain*, tal como podemos observar na Figura 51.

```
<owl:ObjectProperty rdf:ID="tem_nome" />
<rdfs:domain rdf:resource="#Curso"/>
</owl:ObjectProperty>
```

Figura 51 – Exemplo de declaração de propriedades em OWL

O esquema RDFS apenas dispõe do *rdfs:domain* e do *rdfs:range* ao nível das restrições das propriedades, enquanto que OWL disponibiliza (Campos e Lima, 2006):

- Simetria: se A "tem_propriedade" B, então B "tem_propriedade" A (por exemplo: "é_casado_com");
- Funcional: no máximo um valor por instância (por exemplo: "tem_pai");
- Transitiva: se A "tem_propriedade" B e B "tem_propriedade" C, então A "tem_propriedade" C (por exemplo: "maior_que");
- Inversa: uma propriedade é inversa de outra (por exemplo: "possui" e "é_possuído");
- Equivalência: uma propriedade é equivalente à outra (por exemplo: "designação" e "título");
- Expressões booleanas sobre as classes:
  - Disjunção (carro OR bicicleta),
  - Conjunção (veículo AND símbolo_de_status),
  - Negação (NOT animal);
- Classes definidas:
  - Para condições necessárias:
    Exemplo: "Todos os leões comem carne" ou "`Leão ⇒ comer carne`"),
  - Para condições suficientes:
    Exemplo: "Todas as pessoas que não comem carne nem peixe são vegetarianas!" ou "`vegetariana ⇔ pessoa come (NOT (carne OR peixe)`".

Por conseguinte, quanto maior for o grau de expressividade fornecido pela linguagem ontológica, maior poderá ser o poder das inferências realizadas (Pollock e Hodgson, 2004). Hodgson (2004) dispôs as várias linguagens de representação do conhecimento em quadrantes semânticos para ilustrar o grau de expressividade e inferência que cada uma delas proporciona.

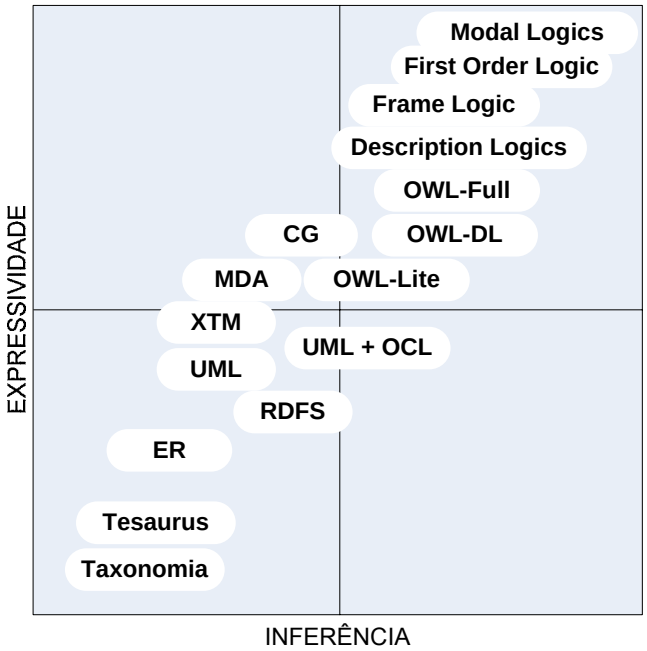


Figura 52 – Quadrantes Semânticos

Numa perspectiva de expressividade incremental, OWL apresenta uma abordagem em camadas (McGuinness e Harmelen, 2004):

- ***OWL Lite*** – sub-linguagem que fornece um menor grau de expressividade. Fornece o suporte para aqueles que necessitam primeiramente de uma hierarquia de classificação com restrições simples. Por exemplo, suporta restrições de cardinalidade, porém só permite valores de cardinalidade 0 ou 1;
- ***OWL DL*** ou ***OWL Description Logic*** – sub-linguagem que inclui todos os construtores OWL para fornecer um maior grau de expressividade onde todas as conclusões são completa e garantidamente computáveis (*computational completeness*) e todos os processamentos terminam num tempo finito (*decidability*);
- ***OWL Full*** – sub-linguagem que fornece o máximo de expressividade e liberdade sintáctica, permitindo que o vocabulário predefinido de um RDF(S) ou OWL seja alargado ou combinado, mas sem garantia computacional. Por exemplo, uma classe pode ser tratada simultaneamente como uma colecção de indivíduos ou como um

único indivíduo. Assim, os mecanismos de inferência poderão não derivar conclusões ou poderão não processar num determinado tempo finito.

Cada uma destas sub-linguagens é uma extensão da sua predecessora. Inicialmente poderemos optar por *OWL Lite*, mas à medida que necessitamos de um maior nível de expressividade, teremos que passar para *OWL DL* ou, eventualmente, para *OWL Full*.

Embora a linguagem OWL seja a norma recomendada pelo W3C no âmbito da WS, convém referir que, qualquer que seja a linguagem ontológica utilizada, a sua sintaxe deverá seguir as recomendações do W3C inerentes à representação do conhecimento. Em primeiro lugar, a linguagem ontológica deve ter uma sintaxe XML bem definida baseada num DTD ou num XML *Schema*. Em segundo, a linguagem ontológica deve assumir-se como uma extensão da linguagem RDF e do seu esquema RDFS (Horrocks et al., 2003).

Podemos então afirmar que as linguagens ontológicas permitem ampliar as linguagens XML e RDF(S), fornecendo a semântica formal que torna os recursos Web legíveis e interpretáveis para os agentes inteligentes. Por exemplo, "o grau de parentesco pai é um relacionamento mais geral do que o grau de parentesco mãe" e "Maria é a mãe de Francisco" são duas expressões que juntas permitem que um sistema possa concluir que "Maria é um dos pais de Francisco". Por conseguinte, se um utilizador perguntasse a um sistema de busca semântico "Quem são os pais de Francisco?", a aplicação correspondente poderia dar como resposta “Maria é um dos pais de Francisco”, mesmo que esse facto não estivesse mencionado em lado nenhum.

A utilização de ontologias permite lidar com conceitos, representando-os formalmente com vista a estruturar objectivamente um determinado domínio do conhecimento; evita problemas inerentes ao vocabulário da linguagem natural (homonímia, sinonímia, metonímia, etc); possibilita o intercâmbio e interoperabilidade do conhecimento entre diferentes domínios; favorece a reutilização do conhecimento e facilita a formulação de consultas para capturar significado dos documentos tendo em conta a semântica do domínio, entre outras características.

As ontologias assumem-se então como fulcrais no contexto das Ciências da Computação e da Informação e, em particular, no âmbito da Web Semântica. As principais áreas de trabalho envolvendo ontologias são (Campos e Lima, 2006):

- Metodologias para o projecto e desenvolvimento de ontologias;

- Linguagens ou formalismos para representação de ontologias;
- Ferramentas para criação e edição de ontologias;
- Métodos, técnicas e ferramentas para integração de ontologias;
- Modelos para geração (semi)automática de ontologias;
- Aprendizagem de ontologias baseada em recursos textuais;

Para além de áreas de aplicação, tais como: Ontologias para melhorar a semântica de descritores (grupos de investigação no âmbito do DCMI e *Stanford Medical Informatics at the Stanford University School of Medicine* no âmbito da ferramenta Protégé); para apoiar a recuperação de informação (OntoSeek, WebKB2, C-Web - *Community Web*, SEAL - *Semantic Portal*); para a gestão do conhecimento (CoMMA - *Corporate Memory Management through Agents*, ACACIA - *Knowledge acquisition for aided design through agent interaction*, PlanetOnto ou Kmi Planet, entre muitos outros projectos no âmbito da utilização de ontologias médicas); para o comércio electrónico (MKBEEM – *Multilingual Knowledge Based European Electronic Marketplace*, SMART-EC - *Smart-EC Support for Mediation and Brokering for Electronic Commerce*); para o processamento de linguagens naturais (Oncoterm, Gazelle, Ontogeneration); Grid Semântico (*Semantic Grid community portal*); para melhorar a semântica em bases de dados e para arquitecturas de agentes baseadas em ontologias, não esquecendo o contexto educativo onde surgiram timidamente alguns projectos de aplicação de ontologias (SchoolOnto, 2004; Edutella, 2006; ELENA, 2006).

Para além da RDFS e OWL, existem outras linguagens de criação de ontologias e as correspondentes ferramentas de construção e edição de ontologias e de anotação de páginas Web. Umas foram desenvolvidas no âmbito de iniciativas paralelas ao projecto da WS promovido pelo W3C e contribuíram decisivamente para as recomendações RDFS e OWL, e outras já existiam e têm vindo a adaptar-se à visão da WS.

As linguagens ou formalismos para representação de ontologias podem ser agrupadas em linguagens baseadas em lógica de primeira ordem ou em *frames* (KIF, Ontolingua, Loom, etc.) e em linguagens baseadas em XML (SHOE, XOL, OML, RDFS, OIL, DAML+OIL, OWL, etc). Polikoff e Allemang (2003) classificaram estas tecnologias do conhecimento em tecnologias semânticas, tecnologias de representação do conhecimento (campo da Inteligência Artificial), linguagens de gestão de conteúdo, linguagens de processamento do conhecimento e linguagens de modelação (campo da Engenharia de Software). Mais informação sobre estas

tecnologias de representação do conhecimento, bem como sobre alguns repositórios de ontologias, pode ser consultada no anexo D.

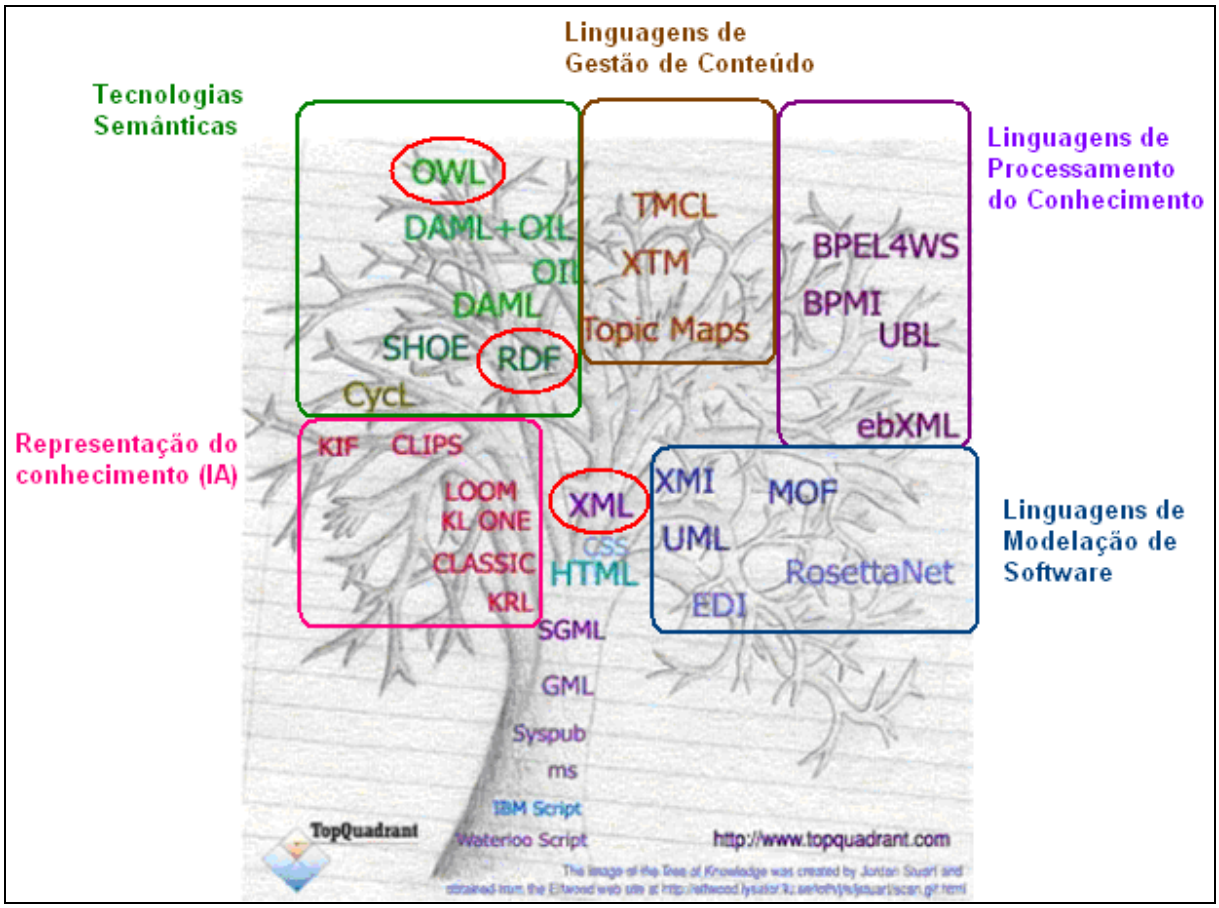


Figura 53 – Árvore das Tecnologias do Conhecimento

As tecnologias semânticas têm as seguintes principais áreas de actuação (Hodgson, 2006): a gestão de conteúdos (descrição de objectos de informação com metadados, apresentação de informação personalizada ao utilizador, distribuição de conteúdos de notícias); a gestão de conhecimento (pesquisa baseada em conceitos, localização e recuperação baseadas no contexto, agentes inteligentes de localização de informação, plataformas de agentes colaborativos); a interoperabilidade semântica (integração de dados, inferência, *Web Services* semânticos); e as interfaces e assistentes (interfaces dinâmicas, assistentes de anotação semântica, outros assistentes ou agentes semânticos).

As ontologias podem diferir quanto à sua estrutura, mas existem características e componentes básicos comuns à maioria delas, pelo que é possível classificá-las genericamente (Almeida et al., 2005):

- Quanto à função: ontologias de domínio, de tarefa e gerais (Mizoguchi et al., 1995);

- Quanto ao grau de formalismo: ontologias informais, semi-informais, semi-formais e formais (Uschold e Gruninger, 1996);
- Quanto à aplicação: ontologias de autoria neutra, de especificação e de acesso comum à informação (Jasper e Uschold, 1999);
- Quanto à estrutura: ontologias de alto nível, de domínio e de tarefa (Haav e Lubi, 2001);
- Quanto ao conteúdo: ontologias terminológicas, de informação, de modelação do conhecimento, de aplicação, de domínio, de representação e genéricas (Van-Heijist et al., 1997).

Face ao seu sucesso no âmbito da Inteligência Artificial, não poderíamos deixar de referir também as redes semânticas como um caso particular de ontologias. As redes semânticas representam os objectos como nós de um grafo e as relações entre eles como arcos que possuem um significado particular. Segundo Hodges (1997), tudo o que pode ser expresso em lógica de primeira ordem, também poderá ser expresso como uma rede semântica. O exemplo mais conhecido de uma rede semântica é a WordNet (Fellbaum, 1999).

Para apoiar a construção de ontologias surgiram diversas ferramentas, normalmente associadas a cada um dos tipos de linguagens ontológicas, tais como: Protégé, SemanticWorks, SWOOP, SMORE, etc (Breitman, 2005). Mais informação sobre estas ferramentas pode ser consultada no anexo E.

Uma vez que a OWL é a linguagem recomendada pelo W3C para a criação de ontologias adequadas à WS, a grande maioria das linguagens e ferramentas têm vindo a fornecer suporte (por exemplo através de *plug-ins*) na migração das ontologias para a versão OWL.

Não obstante, há que ter consciência de que ainda estamos no início de uma nova vaga e organizar a Web ontologicamente esbarra em problemas diversos, tais como:

- Alteração das páginas Web publicadas: dificilmente haverá migração da grande maioria das páginas Web publicadas ou a haver, só quando a arquitectura para a Web Semântica atingir um nível de maturidade suficiente para que os agentes de software processem exclusiva e logicamente ontologias para obter informação e conhecimento;
- Complexidade na especificação de ontologias por parte dos utilizadores Web comuns, bem como na instanciação das suas páginas Web, ainda mais numa arquitectura em camadas (XML – RDF – OWL);

- Dificuldade do cibernauta comum na formulação de consultas através de linguagens de consulta para a Web Semântica, envolvendo ontologias e regras lógicas, já que ainda são raras as interfaces amigáveis para a pesquisa semântica;
- Incertezas decorrentes da veracidade e exactidão das descrições contidas nas páginas que requerem camadas para verificação da autenticidade e da qualidade da informação, nomeadamente no que diz respeito à autoria (publicação de informação acerca da instituição e do autor do conteúdo), à acuidade (correcção, precisão, fidelidade ou exactidão do conteúdo), à actualidade (data de criação e/ou actualização do conteúdo), à abrangência (percepção da cobertura inerente à estrutura e organização do conteúdo) ou mesmo à objectividade (identificação clara do objectivo principal e nível de detalhe do conteúdo);
- A ambiguidade semântica, tal como acontece nas dúvidas de interpretação intrínsecas à linguagem natural, uma vez que algumas páginas Web possuem conceitos vagos e ambíguos, quer devido ao tipo de conteúdo (como, por exemplo, no caso da poesia), quer devido ao contexto (como, por exemplo, o vocabulário usado inerente à cultura geográfica);
- Morosidade na adopção de normas. Dificilmente uma norma é adoptada pela "rede da liberdade" num curto espaço de tempo. Embora tenham fornecido um importante contributo para as recomendações actuais, as diversas propostas de metalinguagens e de linguagens ontológicas, entretanto usadas no desenvolvimento de aplicações para a Web Semântica, acabarão por ter que migrar para as normas recomendadas. Não obstante, uma rápida adopção das normas poderia originar problemas de escalabilidade, tanto das ontologias como da indexação ontológica;
- As partes não obtêm o mesmo resultado que o todo. Dificilmente cada uma das linguagens e tecnologias para a Web Semântica por si só permitirá resolver as limitações actuais da Web. A descrição e classificação de objectos de informação e a sua recuperação e extracção devem ser suportadas por uma infra-estrutura para a Web Semântica que integre as diversas camadas de linguagens e tecnologias. Aliás, os objectivos da arquitectura para a Web Semântica confirmam a necessidade de solucionar, ou pelo menos minimizar, os problemas apresentados.

4.1.2.3- Interoperabilidade de Ontologias

Desenvolver ontologias é sem dúvida um trabalho importante. Contudo, diferentes grupos de pessoas usarão ontologias diferentes para conteúdos de um mesmo domínio. Entretanto, para serem verdadeiramente úteis, as ontologias devem ser compartilhadas de modo a que haja uma compreensão comum entre os produtores de conteúdos no que diz respeito ao significado dos termos. Logo, a combinação, a intersecção, a integração e o alinhamento das ontologias, enfim a interoperabilidade de ontologias, são também exigências cada vez mais importantes. A partilha de ontologias poderá assim impulsionar a descoberta e reutilização inteligentes dos conteúdos, em geral, e dos objectos de aprendizagem, em particular.

Neste contexto destacam-se as seguintes abordagens para a interoperabilidade de ontologias:

- Algoritmo para descoberta de mapeamentos semânticos, cruzando classificações hierárquicas, baseado na coordenação semântica (Bouquet et al., 2003);
- Sistema GLUE que usa estratégias de aprendizagem múltiplas para encontrar os mapeamentos semânticos entre duas ontologias (Doan et al., 2003);
- Conjunto de ferramentas PROMPT para manipular diversas e extensas ontologias. Dentre as ferramentas deste conjunto, destacamos: PROMPTfactor - ferramenta para extracção de partes de ontologias; iPROMPT - ferramenta interactiva para combinação de ontologias; anchorPROMPT - uma ferramenta automática baseada em grafos para alinhamento de ontologias e PROMPTdiff - ferramenta para identificação de diferenças entre duas versões da mesma ontologia (Noy e Musen, 2001).
- Sistema *on-line Articulation Service* que realiza o mapeamento de duas ontologias. A primeira ontologia é a chamada ontologia de assunto e a segunda ontologia objecto. O mapeamento é realizado de forma assimétrica, da ontologia de assunto para a ontologia objecto (Articulation, 2005).
- Sistema *on-line* OntoMerge (*Ontology Translation by Merging Ontologies*) que realiza a combinação de ontologias pela união dos seus axiomas. É um serviço semi-automático que dá suporte a humanos e a agentes de software na tarefa de encontrar diferenças de notação entre duas ontologias de áreas de sobreposição.

No âmbito do projecto OntoWeb (Fernández-López, 2002) foram analisadas algumas das principais abordagens para a interoperabilidade de ontologias, pelo que remetemos para esse projecto e para os estudos de outros autores (Hoog, 1998; Denny, 2004) com vista a complementar e comparar as abordagens aqui apresentadas.

A aplicação de metadados e ontologias aos sistemas de e-Learning não é propriamente uma novidade motivada pela WS. No entanto, a aplicação de metadados e ontologias segundo a perspectiva da WS pode traduzir-se numa maior precisão na inferência e na personalização dos serviços de aprendizagem em função dos requisitos do utilizador. Neste sentido, RDF/RDFS, OWL, DAML+OIL, Ontolingua, entre outras linguagens desenvolvidas para a reutilização e intercâmbio de conhecimento, tais como KIF (*Knowledge Interchange Format*) e UPML (*Unified Problem-solving Method description Language*), têm sido as linguagens mais referenciadas na bibliografia da área de investigação em e-Learning.

Em suma, o sucesso da WS e do e-Learning depende significativamente da proliferação de ontologias, pelo que se requer uma construção rápida das mesmas (Maedche e Staab, 2001). Actualmente, existem algumas propostas para desenvolver métodos de criação e manutenção automática de ontologias. No entanto, há ainda muito trabalho por fazer, nomeadamente ao nível das linguagens de definição de ontologias, das metodologias e das ferramentas de construção e manipulação de ontologias, da integração de ontologias, da aprendizagem de ontologias, das ferramentas de anotação de páginas, dos raciocinadores ou motores de inferência, ou mesmo dos agentes inteligentes e dos Web Services, entre outras ferramentas ou *frameworks* para o desenvolvimento de aplicações para a WS.

4.1.3- Mapas de Tópicos

Os mapas de tópicos, introduzidos no capítulo II como uma forma de tradução dos mapas de conceitos num formato XML para promover a representação e recuperação dos objectos de informação de um curso e a interoperabilidade semântica, constituem uma alternativa viável às tecnologias para as ontologias, desde que sejam criados como estruturas ontológicas para suportar a representação e recuperação de objectos de aprendizagem.

Um mapa de tópicos expressa assuntos ou temas através de tópicos dependendo do domínio de aplicação desse mapa (identificando os temas que são relevantes em determinado

domínio do conhecimento para o fim a que se destina o mapa), do utilizador (identificando os requisitos) e do autor (reflectindo a opinião do criador sobre o que os tópicos são) (Librelotto, 2005).

Os objectivos principais dos Mapas de Tópicos são (Rath, 2003):

- Classificar e qualificar o conteúdo dos recursos de informação como tópicos para habilitar ferramentas de navegação tais como: índices, referências cruzadas, glossários, dicionários, navegação por conceitos;
- Ligar os tópicos com vista a permitir a navegação entre eles;
- Estruturar recursos de informação não estruturados;
- Facilitar a localização e recuperação de informação;
- Filtrar a informação de acordo com o perfil do utilizador ou com o objectivo específico.

Recorrendo à analogia usada por Charles Goldfarb (2001), um mapa de tópicos pode ser comparado a um GPS aplicado ao mundo da informação e do conhecimento.

A parte principal de um mapa de tópicos é conhecida por TAO e corresponde aos seguintes elementos (Pepper, 2000):

- **T (*Topic* ou Tópico)** – pode ser qualquer coisa: uma pessoa, um nome próprio, uma entidade, uma localidade, um assunto, um conceito, um termo, uma ideia, um exemplo ou qualquer outra representação da realidade. Um tópico é um objecto (*object*) que reifica um assunto (subject). Ou seja, é um objecto que serve de intermediário (*proxy*) para algum assunto ou termo que representa uma ideia ou um conceito abstracto da realidade (por exemplo: “Vitor Gonçalves”, “Linguagens de Programação”, “Redes e Serviços Telemáticos”, “CET”, “ESEB”, etc). Embora um mapa de tópicos possa conter vários tópicos reificando o mesmo assunto, após o processamento haverá apenas um tópico para cada assunto. De salientar que os tipos de tópicos são considerados também tópicos. De acordo com os 5 tópicos do exemplo, poderíamos acrescentar os seguintes tipos de tópicos: “Professor”, “Disciplina”, “Curso” e “Escola”. Um tópico pode ter vários nomes (rótulos): “acção de formação” ou “curso de formação” ou “*training course*”, já que ambos representam o mesmo tópico;

- **A (*Association* ou Associação)** – corresponde à relação entre dois ou mais tópicos (por exemplo: escrito_por, representado_por, é_exemplo_de, é_um, etc.). De acordo com os 5 tópicos referidos no exemplo acima, poderíamos ter as seguintes associações: "Vitor lecciona Linguagens de Programação", "Linguagens de Programação é leccionada *por* Vitor", "Linguagens de Programação pertence ao CET", "CET é ministrado na ESEB", etc. De referir que os tipos de associações também são definidos como tópicos;

- **O (*Occurrence* ou Ocorrência)** – é a ligação entre o tópico e um ou mais objectos de informação considerados relevantes para o tópico (por exemplo: página do *curriculum vitae* do docente xlink:href="www.vgportal.ipb.pt/vg/index.php?page=vita"). Portanto, um tópico aponta para uma ou mais ocorrências, tal como podemos demonstrar no nosso exemplo: "documento do *curriculum vitae* detalhado do professor", "link para a Webpage do professor", "documento do programa da disciplina", "comentário sobre a disciplina", "guia do curso CET", "vídeo de apresentação da ESEB", "link para a Webpage da escola", etc).

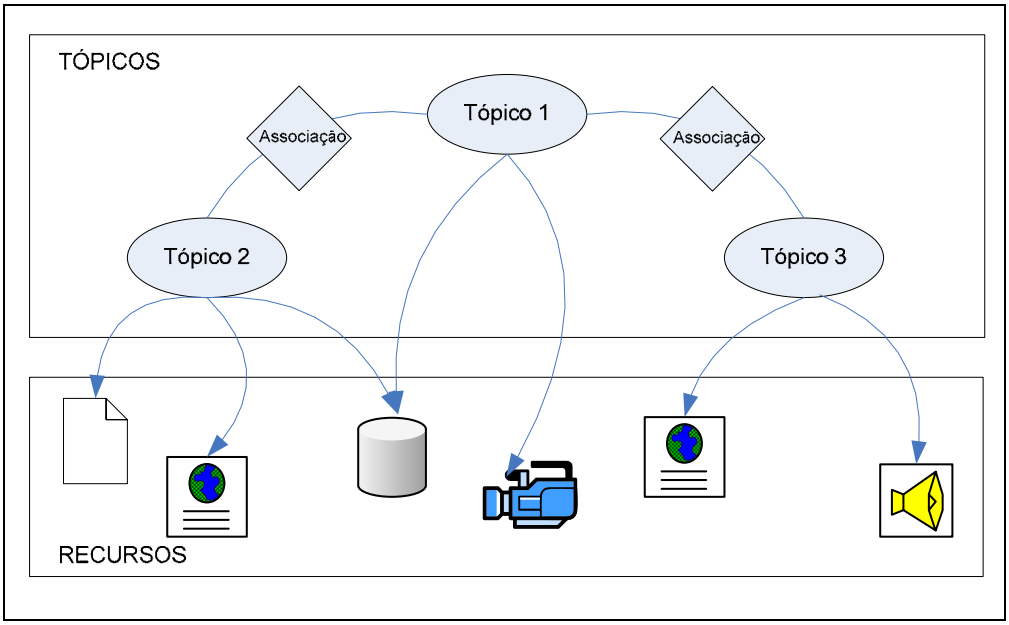


Figura 54 – Independência entre mapas de tópicos e recursos de informação

Estes elementos são suficientemente intuitivos para criar um mapa de informação, mantendo os recursos de informação na sua forma original. Ou seja, a independência entre os mapas de tópicos e os recursos ou objectos de informação é preservada. Os objectos de informação mantêm a sua forma original, não são modificados, o que permite o seu uso como ocorrências noutros mapas de tópicos, impulsionando assim a reutilização desses objectos de informação.

4.1.3.1- Norma Topic Maps

Neste últimos anos, com vista a normalizar a construção de mapas de tópicos e encontrar uma notação universal para os descrever, surgiram algumas propostas e linguagens: HyTM - *HyTime Topic Maps* (Newcomb, 2003), AsTMa= (Barta, 2005), LTM – *Linear Topic Maps* (Garshol, 2002) e XTM – *XML Topic Maps* (Pepper e Moore, 2001). Esta última acabaria por constituir a sintaxe recomendada para a norma *Topic Maps* (ISO/IEC 13250), substituindo a sintaxe inicialmente adoptada, HyTM (ISO/IEC/10744).

A norma *Topic Maps* ISO 13250 2nd Edition (Biezunski et al., 2002) está organizada no seguinte conjunto de normas: ISO 13250-2 *Data Model*; ISO 13250-5 *Reference Model*; e nas seguintes normas candidatas *Topic Map Query Language* e *Topic Map Constraint Language*.

Este conjunto de normas permite representar estruturas de conhecimento, ou seja, estruturas de recursos de informação usadas para definir tópicos e as correspondentes inter-relações. XTM é o formalismo ou linguagem baseada em XML que permite a descrição, manipulação e intercâmbio de mapas de tópicos no âmbito da Internet, de acordo com a sintaxe definida formalmente por um DTD (Pepper e Moore, 2001). É a linguagem recomendada no âmbito da norma ISO/IEC 13250:2 *Topic Maps* para descrever estruturas de conhecimento e associá-las aos recursos de informação correspondentes. A Figura 55 ilustra o principal vocabulário XTM para expressar mapas de tópicos.

```
<?xml version="1.0" encoding="UTF-8" ?>
 <topicMap id="12345" xmlns="http://www.topicmaps.org/xtm/1.0/"
                     xmlns:xlink="http://www.w3.org/1999/xlink">

  <topic id="12345_1">
   ...
     <occurrence>
        <instanceOf>
           <topicRef xlink:href="#Website_mtm">
        </instanceOf>
        <resourceRef xlink:href="http://www.fe.up.pt/mtm" />
     </occurrence>
   ...
  </topic>

   ...

  <association id="assoc_12345_1">
   ...
  </association>

   ...

</topicMap>
```

Figura 55 – Exemplo básico de XTM

Por conseguinte, um Mapa de Tópicos é basicamente um documento XML onde diferentes elementos são usados para representar tópicos, ocorrências de tópicos e relações (ou associações) entre tópicos.

Os tipos de tópicos são definidos declarando uma relação classe/instância, que pode ser vista como uma relação abstracta super-classe/subclasse. Recorrendo novamente ao exemplo da Figura 47 aplicado aos mapas de tópicos, podemos declarar os tipos de tópicos como está representado na Figura 56.

```
<topic id="curso"/>
<topic id="licenciatura">
  <instanceOf>
    <topicRef xlink:href="#curso"/>
  </instanceOf>
</topic>
<topic id="mestrado">
  <instanceOf>
    <topicRef xlink:href="#curso"/>
  </instanceOf>
</topic>
<topic id="LEI">
  <instanceOf>
    <topicRef xlink:href="#licenciatura"/>
  </instanceOf>
</topic>
<topic id="MTM">
  <instanceOf>
    <topicRef xlink:href="#mestrado"/>
 </instanceOf>
</topic>
```

Figura 56 – Definição dos tipos de tópicos em XTM

Aos tópicos podem ser associadas várias características. Para validar essas características existe o contexto (*scope*). Ou seja, só no contexto é que a atribuição de características é válida. Uma correcta aplicação do contexto permite eliminar ambiguidades, para além de permitir alterar a apresentação de informação ao utilizador consoante o seu perfil. O contexto é representado por um conjunto de tópicos, tal como se pode verificar na Figura 57.

```
<topic>
   <baseName>
      <baseNameString>Tecnologia Multimédia</baseNameString>
   </baseName>
   <baseName>
      <scope>
         <topicRef xlink:href="#deec"/>
         <topicRef xlink:href="#feup"/>
         <topicRef xlink:href="#pt"/>
         <topicRef xlink:href="#mtm"/>
      </scope>
   <baseNameString>MTM</baseNameString>
   </baseName>
</topic>
```

Figura 57 – Definição do contexto em XTM

Uma vez que a estrutura do documento XTM é definida por um DTD, a validação sintáctica do mapa de tópicos pode ser realizada por qualquer processador ou *parser* de XML.

A sintaxe da linguagem XTM 1.0 é constituída por 19 elementos:

- ***<topicRef>***: referência URI para um tópico;
- ***<subjectIndicatorRef>***: referência URI para um recurso que indica um assunto;
- ***<scope>***: referência a tópicos (*<topicRef>*), recursos (*<resourceRef>*) ou indicadores de assuntos (*<subjectIndicatorRef>*) compreendidos no contexto. A união dos assuntos que correspondem a estes elementos especifica o contexto no qual a atribuição da característica do tópico é considerada válida;
- ***<instanceOf>***: instância de um tópico ou apontador para um tópico que representa uma classe;
- ***<topicMap>***: documento mapa de tópicos. É o elemento pai de todos os elementos *<topic>*, *<association>* e *<mergeMap>* de um mapa de tópicos;
- ***<topic>***: nome e ocorrência de um tópico;
- ***<subjectIdentity>***: assunto que é representado por um tópico;
- ***<baseName>***: nome base de um tópico;
- ***<baseNameString>***: *string* que representa o nome base do tópico anterior pai de um tópico;
- ***<variant>***: formas alternativas do nome base;
- ***<variantName>***: forma alternativa a ser usada como variante do nome base;
- ***<parameters>***: processador de contexto para as variantes; a união dos assuntos que correspondem aos elementos *<topicRef>* ou *<subjectIndicatorRef>* especifica um contexto adicional no qual os nomes das variantes são considerados apropriados;
- ***<association>***: associação entre tópicos;
- ***<member>***: membro de uma associação;
- ***<roleSpec>***: papel desempenhado por um membro ou tópico numa associação;
- ***<occurrence>***: recurso considerado como uma ocorrência, ou seja, recurso que fornece informação relevante a um tópico;

- <***resourceRef***>: referência URI a um recurso;
- <***resourceData***>: informação sobre uma ocorrência de um tópico ou uma variante do nome base;
- <***mergeMap***>: referência URI relativa a um mapa de tópicos externo. O *mergeMap*, tal como o próprio nome indica, permite a fusão de mapas de tópicos através da união dos tópicos cujo identificador seja igual; no mapa de tópicos resultante esses tópicos iguais são representados num único tópico com todas as características dos tópicos originais.

A Figura 58 ilustra os elementos que compõem a sintaxe da XTM (Pepper e Moore, 2001). Para obter mais informação, recomenda-se a consulta da norma *Topic Maps* que se divide em cinco partes: Parte 1: *Overview and Basic Concepts*; Parte 2: *Data Model*; Parte 3: *XML Syntax*; Parte 4: *Canonicalization*; e Parte 5: *Reference Model*.

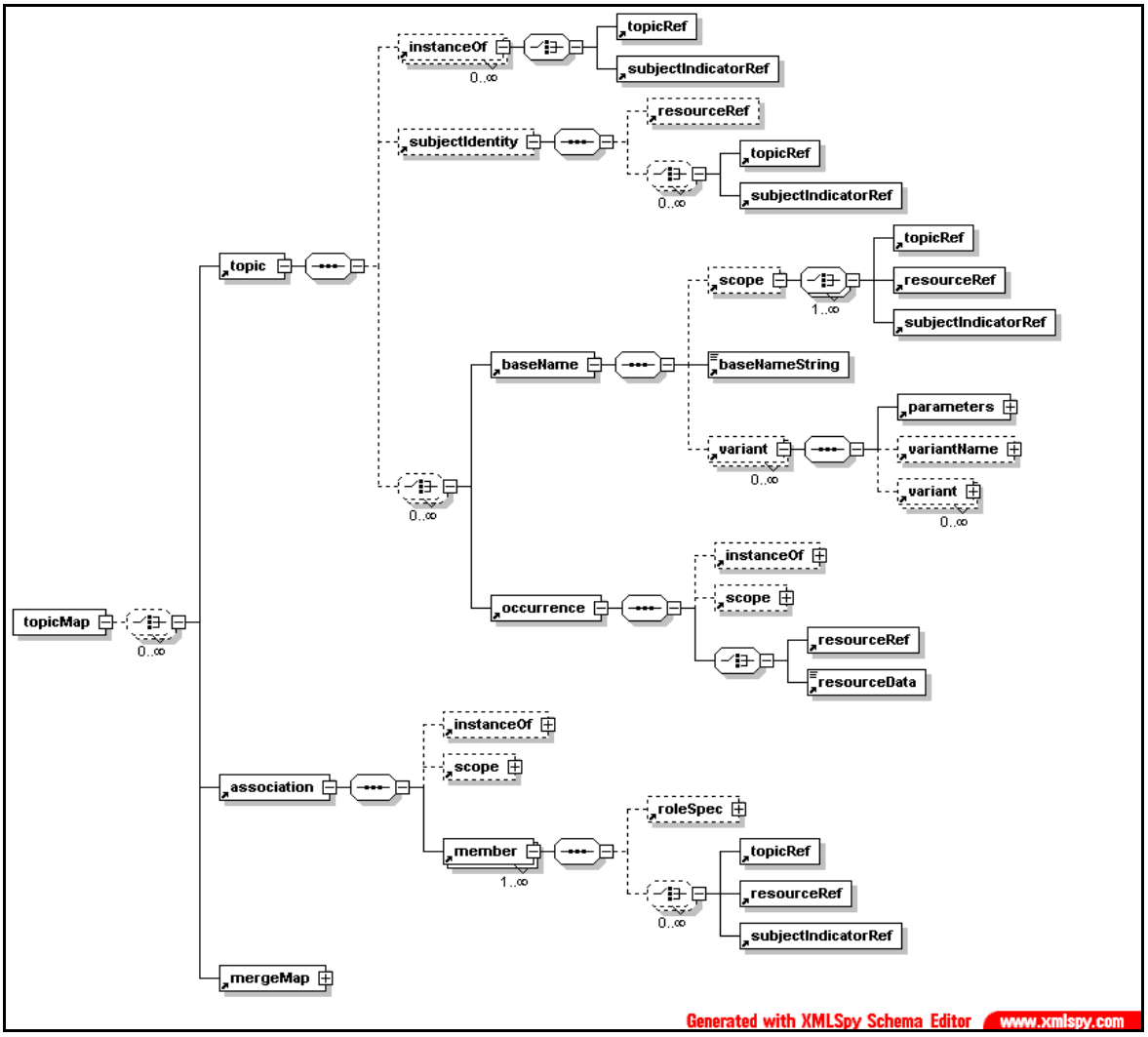


Figura 58 – Sintaxe da XTM

Tal como referido, existem algumas ferramentas para a criação e navegação em mapas de conceitos, mas muito poucas permitem exportar os mapas no formato XML. Uma dessas ferramentas é a *CmapTools* que permite a importação/exportação dos mapas de conceitos no formato normalizado para mapas de tópicos XTM, bem como no formato proprietário e específico para mapas de conceitos XCM (XML *Concept Maps*), não esquecendo a possibilidade de exportar para o formato HTML que permite a visualização da imagem do mapa de conceitos através de *browsers* Web (Cañas, 2004). No entanto, não possui qualquer mecanismo para realizar inferências sobre os mapas de conceitos ou os mapas de tópicos.

As duas soluções mais difundidas para criar e manipular mapas de tópicos, bem como realizar inferências lógicas sobre os mesmos, são *Ontopia Knowledge Suite* (OKS) (Ontopia, 2005a) e *Topic Maps for Java* (TM4J) (TM4J, 2005). OKS é a solução comercial mais difundida, enquanto que TM4J corresponde à solução *open source* mais usada.

Os mapas de tópicos descritos em XTM podem ser visualizados e navegados usando ferramentas ou motores de mapas de tópicos, tais como:

- *Ontopia Omnigator*: aplicação que permite carregar e navegar em mapas de tópicos, usando um *browser* Web. Este motor faz parte do *Ontopia Knowledge Suite* (OKS). A versão gratuita desta ferramenta limita os mapas de tópicos a 5000 tópicos, associações e ocorrências (Ontopia, 2005b).
- *TM4J*: aplicação composta por um conjunto de *packages* Java que fornecem as interfaces e implementações para a importação, navegação, manipulação e exportação de mapas do tópico descritos em conformidade com DTD XTM.
- *Ulisses*: módulo *open source* gerador de navegadores conceptuais, desenvolvido no contexto do sistema *Metamorphosis*, mas que pode ser usado isoladamente.
- *SemanText*: aplicação protótipo desenvolvida para demonstrar como a norma *Topic Maps* ISO/IEC 13250:2000 pode ser usada para representar redes semânticas. É possível adicionar nova informação à base de conhecimento e ao respectivo mapa de tópicos para inferir novo conhecimento.

Para além destas aplicações, existem mais algumas aplicações e protótipos para criação, visualização e navegação em mapas de tópicos, tais como *Topic Map Designer*, *TMTab plugin for Protégé*, *TMNav* e *xSiteable*, entre outras.

Em suma, embora com finalidades diferentes, os mapas de conceitos assemelham-se aos mapas de tópicos, sendo possível a tradução de uns nos outros desde que a estrutura (conceitos, relações e *links* para os recursos) dos mapas conceptuais possam ser armazenados em base de dados para, posteriormente, permitir a construção dos mapas de tópicos através de XTM (Paz et al., 2005).

4.1.3.2- Comparação entre RDF e Topic Maps

Embora definidas por diferentes famílias de normas e com objectivos distintos, RDF e *Topic Maps* têm diversas similaridades como tivemos oportunidade de abordar nas secções anteriores sobre estas duas tecnologias.

Em suma, as principais semelhanças são (Daconta et al., 2003; Librelloto, 2005):

- Ambas foram desenvolvidas como técnicas de representação do conhecimento para a gestão da informação;
- Ambas definem modelos abstractos;
- Ambas possuem linguagens concretas baseadas em XML;
- Ambas oferecem modelos simples, mas extremamente poderosos: no modelo *Topic Maps* os conceitos são representados como tópicos, enquanto que no modelo RDF são representados como recursos.

Contudo, também há uma série de diferenças significativas:

- Foram desenvolvidas por diferentes comunidades e para diferentes objectivos;
- Nos *Topic Maps* a representação do conhecimento é realizada na perspectiva dos humanos, enquanto que nos documentos RDF é efectuada na perspectiva das máquinas ou agentes de software;
- Embora tópicos e recursos possam ser considerados sinónimos, os mapas de tópicos são centrados nos temas ou assuntos, enquanto que os documentos RDF são centrados nos recursos;
- A nível semântico, podemos afirmar que a expressividade dos documentos RDF é inferior à dos mapas de tópicos. Mas, com RDFS, a expressividade desses documentos passa a ser superior;

- O mapeamento ou alinhamento entre RDF e mapas de tópicos é possível, mas o alto nível semântico é perdido na ausência de um esquema para os *Topic Maps*.

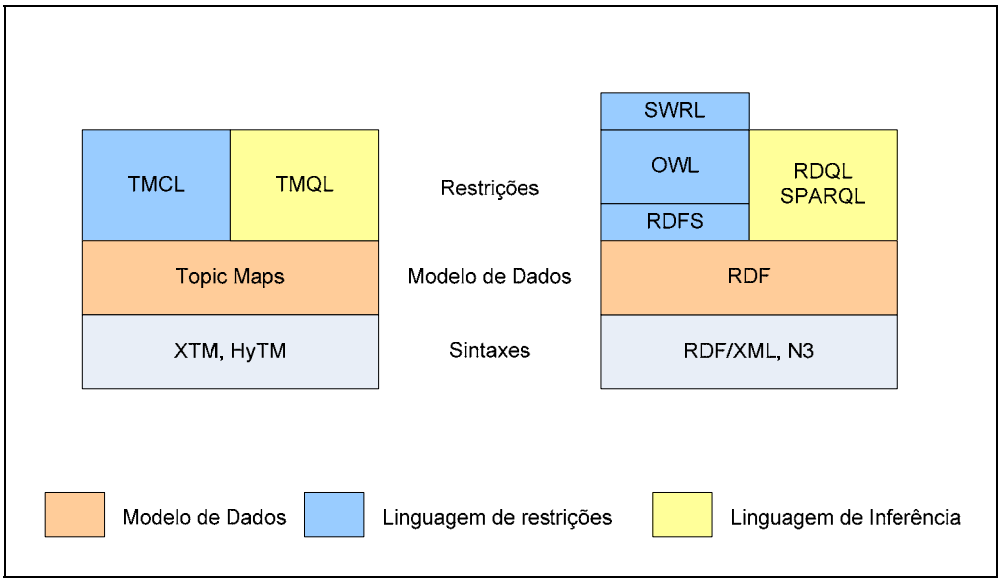


Figura 59 – As famílias de normas W3C RDF e ISO Topic Maps

RDF possui o RDFS (que pode ser considerado um metamodelo para o modelo de objectos RDF), mas a norma *Topic Maps* não possui nenhum modelo similar (Daconta et al., 2003). Com respeito a esta especificidade, nestes últimos anos a comunidade desenvolveu esforços no sentido de desenvolver um modelo abstracto (*reference model*) e as correspondentes linguagens.

Para além da *Topic Map Constraint Language* (TMCL) (Moore et al., 2005), neste último ano emergiu uma nova proposta para a recuperação de recursos de informação articulados em mapas de tópicos baseada na linguagem *Topic Map Query Language* (TMQL) (Garshol e Barta, 2005).

As tecnologias e linguagens de inferência no âmbito da família de normas W3C também estão numa fase de discussão desde 2004. Referimo-nos às propostas RDQL (*RDF Data Query Language*) (Seaborne, 2004) e SPARQL *Query Language for RDF* (Prud'hommeaux e Seaborne, 2004). Os principais conceitos destas linguagens de inferência podem ser encontrados na secção 4.2.2 – Linguagens para os mecanismos de inferência.

A liberdade de representação de um domínio proporcionada pelos mapas de tópicos, nomeadamente devido ao facto da noção de tópico ser ampla, é uma das suas características mais relevantes. Os mapas de tópicos podem portanto ser vistos como "ontologias de alto nível" (XTM) com vista a representar o significado de um determinado domínio numa

perspectiva de abstracção mais elevada. Consequentemente, para descrever os domínios mais específicos numa perspectiva de abstracção mais baixa poderíamos usar "ontologias de mais baixo nível" (RDFS e OWL).

4.2- Tecnologias para a Camada Lógica

Enquanto as páginas possuírem apenas informação lexical os agentes e/ou *robots* de pesquisa, mesmo os mais avançados, encontram um ambiente hostil para a realização das suas tarefas, porque tanto o conteúdo das páginas como o relacionamento entre elas é difícil de ser compreendido pelas aplicações computacionais, por se encontrarem geralmente em linguagem humana ou natural. Mesmo que os conteúdos sejam gerados a partir de uma base de dados, o significado dessa estrutura não é suficientemente claro para que um agente computacional possa realizar inferências sobre os dados.

O objectivo principal das tecnologias de inferência, de consulta ou de interrogação é encontrar um recurso de informação ou um conjunto restrito de recursos a partir de um vasto conjunto de recursos de informação.

A WS é uma Web que saberá responder às nossas questões através de metadados, ontologias e agentes inteligentes. Tal como afirma James Hendler, a Web está a ser transformada num conjunto de recursos que poderão ser entendidos e activados por agentes inteligentes e não apenas por humanos.

Nesta secção são apresentadas as principais linguagens de regras e tecnologias de inferência que constituem uma camada lógica básica para que seja possível inferir conhecimento da Web.

4.2.1- Linguagens para as Regras de Inferência

Com a descrição da estrutura dos dados em XML, a representação da semântica simples desses dados em RDF(S) e a representação formal comummente aceite sobre o que significam esses dados em ontologias mais expressivas como OWL estão criadas as condições para que os agentes possam raciocinar sobre dados com significado bem definido. Não obstante, para que os agentes de software possam raciocinar sobre as estruturas de dados, tirando partido das relações entre os objectos dessas estruturas esclarecidas através de ontologias, pode ser necessário definir previamente regras para os mecanismos de inferência.

Embora as ontologias OWL forneçam construtores suficientes para exprimir a semântica necessária para a realização de inferências, apresentam algumas limitações, nomeadamente ao nível dos construtores de composição de propriedades. Tal como se pode constatar nos exemplos ilustrados na Figura 60, a linguagem OWL não permite expressar relações entre propriedades de ramos diferentes da hierarquia. Ou seja, não existe forma de declarar os relacionamentos entre duas relações compostas (por exemplo, através da indicação de que o tio é a composição do irmão e do pai).

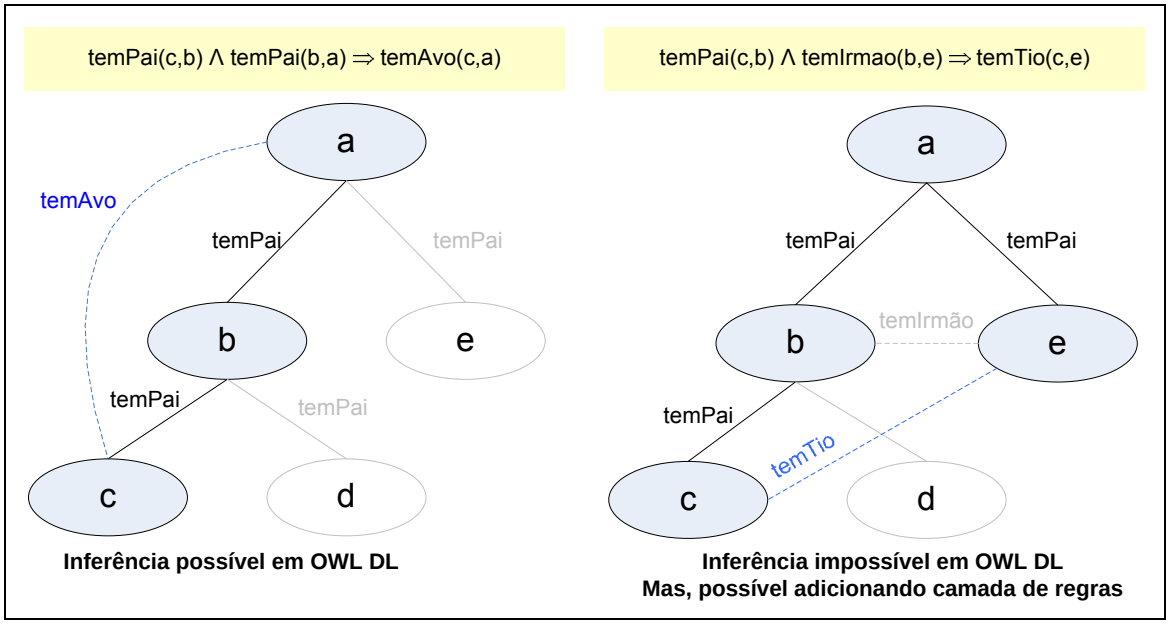


Figura 60 – Uma limitação da OWL

Por conseguinte, em algumas situações haverá necessidade de recorrer a linguagens de regras para amplificar os axiomas OWL embutidos na ontologia, aumentando a capacidade de inferência e permitindo inferir informações implícitas.

Tipicamente, em cada sistema de representação do conhecimento existe uma distinção clara entre os dados e as regras de inferência. As regras controlam as etapas de inferência que o mecanismo de inferência realiza sobre os dados (Berners-Lee, 2000).

As regras de inferência são também um elemento chave para a visão da WS, uma vez que permitem que os dados provenientes de diversas fontes sejam integrados, derivados e transformados de uma forma distribuída, transparente e escalável.

A iniciativa RuleML (*Rule Markup Initiative*) tem colaborado com o W3C na normalização de regras para a Web, nomeadamente no âmbito dos Sistemas Distribuídos, Web Semântica, Computação Autónoma e Arquitecturas Orientadas a Serviços (*Service Oriented Architecture*) activados semanticamente (RuleML, 2006).

O objectivo desta iniciativa é desenvolver a RuleML (*Rule Markup Language*) como uma linguagem canónica para regras baseada na XML, na semântica formal e em implementações eficientes. A RuleML cobre todo o espectro das regras: regras derivadas, regras transformadas e regras reactivas. Com esta linguagem é possível realizar consultas e inferências em ontologias Web, alinhamentos de ontologias, etc (Boley et al., 2005a).

Recentemente, têm vindo a ser desenvolvidas linguagens baseadas em RuleML. As linguagens de regras para a Web propostas para normalização que mais se destacam actualmente são: SWRL - *Semantic Web Rule Language* (Horrocks et al., 2004), WRL - *Web Rule Language* (Angele et al., 2005) e FOL RuleML - *First-Order Logic RuleML* (Boley et al., 2005b) enquadrada na proposta SWRL FOL (*Semantic Web Rule Language First-Order Logic*). No âmbito de outros projectos surgiram também linguagens para regras de inferência, tais como: DAML-L (*DAML-Logic*); N3 (*Notation 3*); XRML (*eXtensible Rule Markup Language*); RFML (*Relational-Functional Markup Language*); URML (*Universal Rule Markup Language*); AORML (*Agent-Object-Relationship Markup Language*); CBML (*Case Based Markup Language*); Metalog, KIF; ISO *Common Logic*; ISO Prolog; entre outras. Existem também algumas linguagens que têm aplicações específicas em dados domínios tais como: políticas de *Web Services* (WS-Policy, WSPL, Policy RuleML, SWSL, WSML), autorização e controle de acesso (XACML, EPAL, P3P/APPEL), regras para o comércio ou negócio (BRML - *Business Rules Markup Language*, SRML - *Simple Rule Markup Language*), entre outras áreas. Os esforços de normalização têm vindo também a incidir sobre a modelação de regras (*OMG's Business Semantics for Business Rules RFP* e *Production Rules Representation RFP*) e motores de execução de regras (JSR 94 - *Java API for Rules Engine*).

As linguagens de regras para a Web que mais se têm destacado nestes últimos dois anos foram: SWRL e WRL. SWRL é uma linguagem baseada nas cláusulas de Horn e na combinação das sub-linguagens OWL DL e OWL Lite com as sub-linguagens *Datalog* unárias/binárias da linguagem RuleML. Enquanto que WRL é uma linguagem baseada na programação lógica e com um nível de expressividade muito próximo da *Description Logic*. Uma comparação sucinta entre estas duas linguagens pode ser visualizada na Tabela 4 (Moura, 2006).

Parâmetros	SWRL	WRL
Baseada em	Cláusulas de Horn	Programação Lógica
Relação com OWL	Extensão da OWL DL	Possui um subconjunto em comum com a OWL DL
Origem	OWL DL e RuleML	WSML
Negação por falha	Sim, inclusive no cabeçalho da regra	Apenas na WRL-Flight e WRL-Full (excepto no cabeçalho da regra)
Quantificadores	Sim, inclusive no cabeçalho da regra	Apenas na WRL-Full (excepto no cabeçalho da regra)
Semântica	Ambiente aberto	Ambiente fechado

Tabela 4 – Comparação entre SWRL e WRL

No âmbito desta dissertação destacamos a linguagem SWRL, uma vez que pode ser vista como uma extensão da linguagem OWL recomendada pelo W3C, assumindo a semântica segundo uma visão mais coerente com o ambiente da Web Semântica.

Tal como vimos anteriormente, a sintaxe abstracta de uma ontologia OWL contém uma sequência de axiomas e de factos. A SWRL complementa estes axiomas com regras. Recordando a regra temTio da Figura 60, a sintaxe abstracta da SWRL trata as regras como axiomas e apresenta-as num formato legível para as pessoas (ver Figura 61).

```
axiom ::= rule

antecedente ⇒ consequente

      rule ::= 'Implies(' [ URIreference ] { annotation } antecedent consequent ')'
      antecedent ::= 'Antecedent(' { atom } ')'
      consequent ::= 'Consequent(' { atom } ')'
```

Figura 61 – Sintaxe abstracta da SWRL

Para expressar estas regras abstractas num formato legível para os agentes de software usamos SWRL/XML. Assim, no caso do exemplo da Figura 60 a regra temTio poderia ser declarada tal como se pode observar na Figura 62.

Independentemente da linguagem usada, a definição de regras de inferência deve ter em atenção os requisitos dos utilizadores e as necessidades dos fornecedores da tecnologia de regras de inferência.

```
temPai(c,b) ∧ temIrmao(b,e) ⇒ temTio(c,e)
```

```
<ruleml:imp>
  <ruleml:_rlab ruleml:href="#exemplo"/>
  <ruleml:_body>
    <swrlx:individualPropertyAtom  swrlx:property="temPai">
      <ruleml:var> c </ruleml:var>
      <ruleml:var> b </ruleml:var>
    </swrlx:individualPropertyAtom>
    <swrlx:individualPropertyAtom  swrlx:property="temIrmao">
      <ruleml:var> b </ruleml:var>
      <ruleml:var> e </ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_body>
  <ruleml:_head>
    <swrlx:individualPropertyAtom  swrlx:property="temTio">
      <ruleml:var> c </ruleml:var>
      <ruleml:var> e </ruleml:var>
    </swrlx:individualPropertyAtom>
  </ruleml:_head>
</ruleml:imp>
```

temPai(c,b)

temIrmao(b,e)

temTio(c,e)

Figura 62 – Um exemplo com regras em SWRL

4.2.2- Linguagens para os Mecanismos de Inferência

Quanto às linguagens de inferência, as recomendações candidatas recaem sobre RDQL (*RDF Data Query Language*) (Seaborne, 2004) e SPARQL *Query Language for RDF* (Prud'hommeaux e Seaborne, 2006).

A arquitectura genérica para um sistema ou aplicação de pesquisa semântica baseado no motor de inferência SPARQL é ilustrada na Figura 63.

Com base nas palavras-chave submetidas pelo utilizador, a aplicação envia a consulta em formato SPARQL para o motor de inferência. De acordo com os parâmetros dessa consulta, o motor de inferência pode interrogar directamente os documentos em formato RDF(S) ou através de GRDDL (forma de indicar uma transformação, normalmente em

formato XSLT, de um documento XHTML ou XML para RDF), extrair dados RDF dos documentos XHTML e XML ou ainda, através de *SQL-SPARQL Bridge*, consultar a informação da base de dados do sistema ou aplicação.

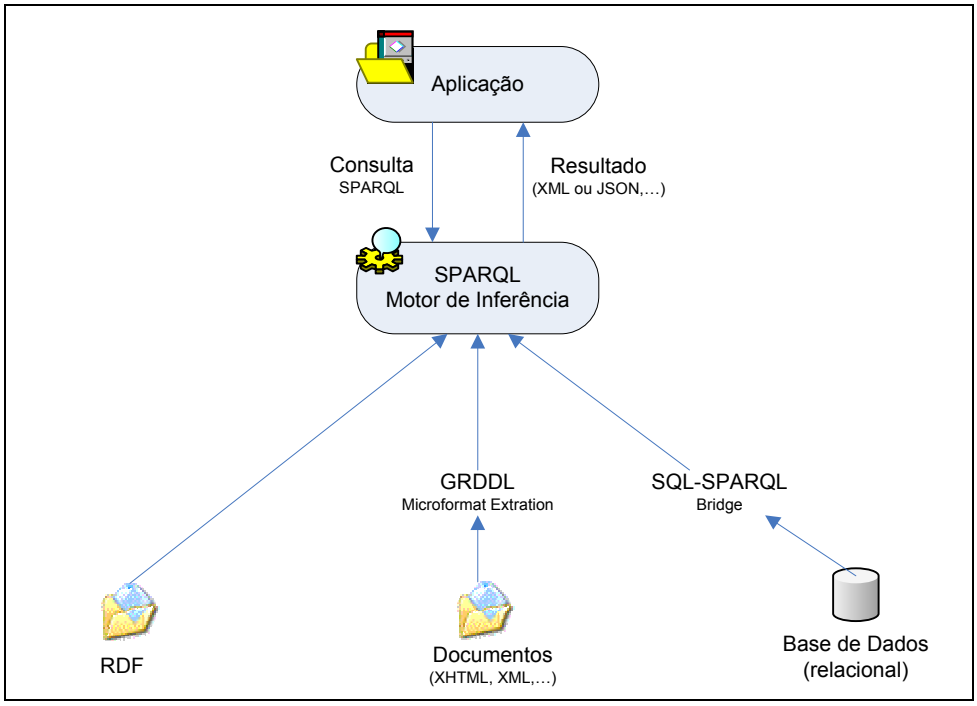


Figura 63 – Arquitectura genérica de uma aplicação com motor de inferência SPARQL

Um exemplo simples de como inferir um documento RDF(S) apresenta-se a seguir:

A Consulta (Query):

```
PREFIX  dc:  <http://purl.org/dc/elements/1.1/>
SELECT  ?title
WHERE   { ?x dc:title ?title
          FILTER regex(?title, "SPARQL")
        }
```

O Resultado da Consulta (Query Result):

```
TITULO:
"Manual das Tecnologias para a Web Semântica"
```

No que diz respeito aos raciocinadores ou *reasonners* que permitem a concretização das inferências, destacamos os que suportam OWL: RACERPro (Haarslev e Moller, 2003); Pellet (Sirin et al., 2006); Fact++ (Tsarkov e Horrocks, 2004); e Bossam (Jang, 2004). Este último inclui suporte para a SWRL.

De referir que a *framework* Jena (McBride, 2002), ferramenta para o desenvolvimento de aplicações para a WS, suporta as linguagens RDF(S) e OWL e inclui alguns raciocinadores ou motores de inferência pré-construídos (*RDFS rule reasoner*, *OWL FB reasoner*, *DAML micro reasoner*, *Transitive reasoner*, *Generic rule reasoner*), para além de permitir a criação de novos motores de inferência.

As regras de inferência e os raciocinadores assumem cada vez mais um papel relevante, nomeadamente nos sistemas baseados em conhecimento, sistemas de interface inteligente e agentes inteligentes para a Web (ou mais especificamente para a WS). As plataformas para o desenvolvimento e gestão de agentes são abordadas detalhadamente na secção seguinte.

4.3- Resumo

A finalidade da Web Semântica passa pelo desenvolvimento de um modelo tecnológico que permita a partilha global de conhecimento assistida por máquinas. Nesta perspectiva, após mais de cinco anos de investigação em torno da visão da Web Semântica, nomeadamente através dos grupos *RDF Core Working Group*, *Web Ontology Working Group*, *Semantic Web Advanced Development*, a 10 de Fevereiro de 2004, o consórcio W3C publicou como recomendações definitivas duas tecnologias-chave para a WS: *Resource Description Framework* (RDF) e *Web Ontology Language* (OWL).

RDF fornece um modelo para expressar os metadados, mas essa informação sobre os recursos de informação é insuficiente para perceber o seu significado. Torna-se necessário associar mais semântica à descrição RDF, ou seja, necessitamos de ontologias.

Este capítulo incidiu essencialmente sobre as tecnologias para as Camadas Semântica e Lógica. A semântica é expressa através de ontologias, enquanto que a lógica é definida através de regras.

Ao contrário das abordagens mais convencionais de representação do conhecimento, as ontologias fornecem um vocabulário aberto para descrever objectos de informação de uma determinada área do conhecimento, pelo que se assumem como fulcrais no âmbito da Web

Semântica. Não obstante, os mapas de tópicos têm vindo também a ser apresentados como uma alternativa viável, desde que usados numa perspectiva ontológica.

RDF por si só não permite criar ontologias; no entanto, ao permitir descrever vocabulários que representam o conhecimento sob a perspectiva das redes semânticas, constitui a base para outras linguagens com essa finalidade, tais como RDFS e OWL.

RDFS define as primitivas para a criação de ontologias simples. Sobre RDFS têm-se construído linguagens com primitivas mais expressivas e com maior capacidade de inferência, das quais se destaca a OWL.

Com a descrição da estrutura dos dados em XML, a representação da semântica simples desses dados em RDF(S) e a representação formal comummente aceite sobre o que significam esses dados em ontologias mais expressivas como OWL estão criadas as condições para que os agentes de software possam raciocinar sobre esses dados. Para realizar a inferência pode ser usada a linguagem SPARQL. No entanto, há situações em que as ontologias não expressam todo o conhecimento necessário para realizar a inferência. Assim, as regras de inferência, definidas por exemplo através da linguagem SWRL, proporcionam uma camada lógica básica para ampliar as capacidades dos agentes inteligentes na localização e recuperação de informação e na geração de conhecimento.

Este capítulo não teve como objectivo apresentar completamente a Camada Lógica, uma vez que essa camada, bem como as restantes camadas superiores da arquitectura para a WS ainda se encontram numa fase embrionária de desenvolvimento, não existindo por enquanto tecnologias recomendadas. Não obstante, as linguagens de regras e as linguagens de inferência apresentadas fornecem uma camada lógica básica que poderá evoluir à *posteriori*, uma vez que a arquitectura da WS é caracterizada pela escalabilidade.

Tal como referem Antoniou e Harmelen (2004), as linguagens e as tecnologias para os metadados, para as ontologias e para a lógica e a inferência são cruciais para o desenvolvimento da Web Semântica.

5- Tecnologias para os Agentes

Este capítulo tem como principal objectivo esclarecer os conceitos de agentes de software e descrever os principais requisitos dos sistemas de agentes móveis e inteligentes. A mobilidade e a inteligência nos agentes não são conceitos recentes. Porém, com o projecto da Web Semântica estes conceitos assumem-se como propriedades decisivas para a realização de determinadas tarefas em nome do utilizador.

A recuperação de objectos de informação é um dos principais objectivos desta tese. Neste sentido, foi também incluído neste capítulo o estado da arte dos mecanismos de pesquisa de informação na Internet.

Finalmente, após descrever a arquitectura genérica para um sistema de agentes móveis, são apresentadas as principais infra-estruturas que suportam o desenvolvimento de aplicações baseadas em agentes móveis.

5.1- Agentes de Software

O conceito de agente é uma metáfora utilizada em inúmeras áreas do conhecimento, tendo sido introduzido pela primeira vez no contexto das Ciências da Computação pela comunidade de Inteligência Artificial (Silva, 1999).

Um agente é um componente de software e/ou hardware capaz de actuar de forma a resolver tarefas em nome do seu utilizador (Nwana, 1996). Embora os agentes computacionais possam ser construídos tanto em hardware como em software, os agentes de software são aqueles que assumem particular importância no contexto deste documento.

A tecnologia para os agentes sofreu a influência de diferentes áreas científicas, tal como ilustrado na Figura 64 (Silva, 1999):

- Inteligência Artificial Distribuída;

- Engenharia de Software;
- Interfaces ou interacção homem-máquina.

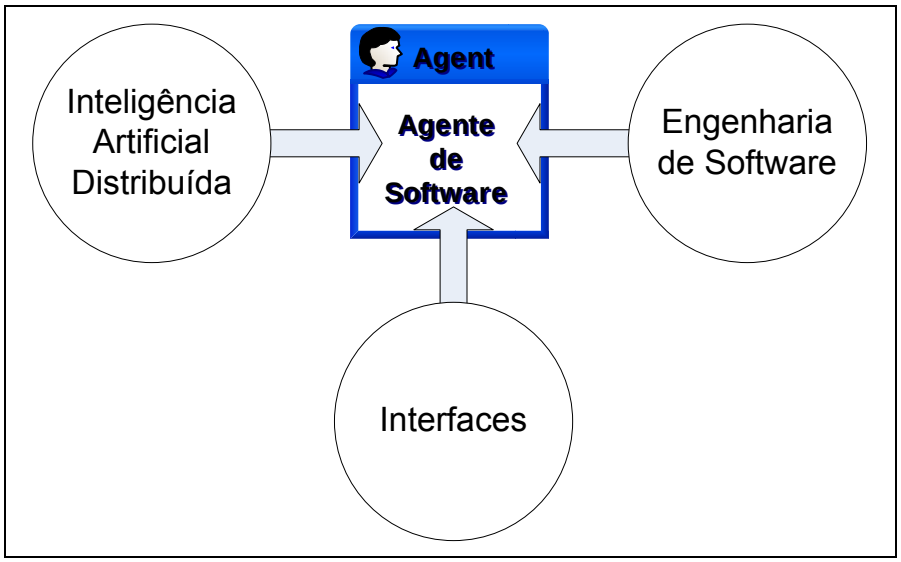


Figura 64 – Áreas científicas influentes na tecnologia para os agentes

Embora não exista uma definição consensual para "agentes de software" (em grande parte motivada pelos contributos dos domínios referidos que conduziram ao aparecimento de diversos tipos de agentes e outras tantas designações), podemos afirmar que são uma analogia dos agentes humanos, como os agentes de viagens ou agentes imobiliários, que nos auxiliam de forma especializada a encontrar aquilo que melhor satisfaz as nossas necessidades.

Actualmente, pesquisar informação educativa na Web com vista a gerar um plano de aprendizagem personalizado pode apresentar a mesma complexidade que preparar uma viagem ou adquirir uma casa. Por conseguinte, tal como no mundo real são necessários agentes de viagens e agentes imobiliários, também na Internet se exigem agentes que realizem tarefas análogas, nomeadamente ao nível da pesquisa, filtragem, recuperação, integração e reutilização de informação, do reconhecimento da linguagem e da aquisição ou geração de conhecimento.

Genericamente, podemos afirmar que os agentes de software são programas ou fragmentos de código, alojados nos computadores e nas redes, que realizam tarefas com vista a satisfazer as necessidades dos utilizadores.

Um agente pode realizar um conjunto de tarefas, como, por exemplo, acesso, filtragem e reutilização da informação e geração de novo conhecimento. Para tal, deverá ser capaz de analisar o ambiente em que se insere e reagir às alterações nele ocorridas, segundo uma base de conhecimento, isto é, seguindo um conjunto de regras previamente definidas ou entretanto apreendidas, com o intuito de comunicar com o utilizador ou interagir com outros agentes.

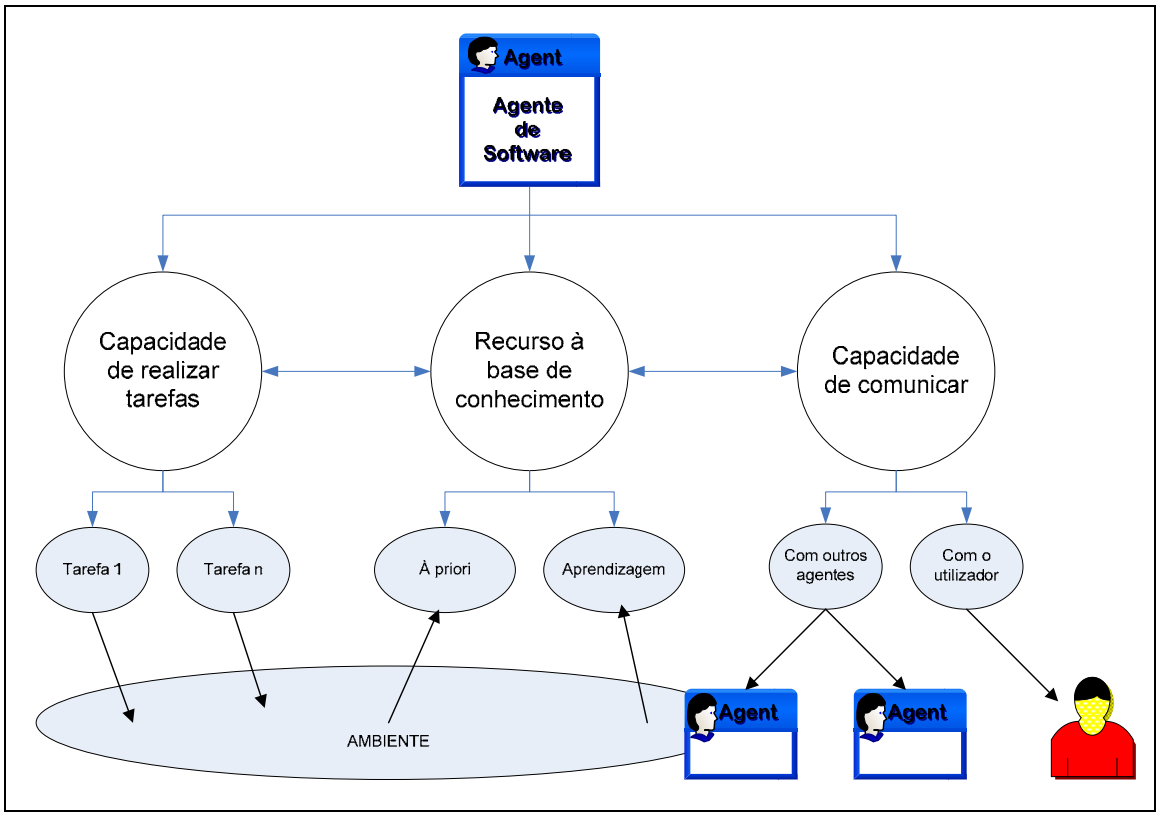


Figura 65 – Modelo de agente

As ontologias e as regras de inferência podem constituir a base de conhecimento indispensável à actuação dos agentes de software. Por conseguinte, a visão da WS traduzirá os conteúdos Web numa estrutura com significado e definirá a forma de lidar com essa estrutura, criando um ambiente propício para os agentes de software (Figura 65).

Nestes últimos anos têm surgido várias propostas na tentativa de classificar os diferentes tipos de agentes. Para que os possamos classificar, convém entender os seus atributos mais relevantes.

5.1.1- Atributos dos Agentes

Uma vez que cada caso é um caso, os agentes de software dependem dos requisitos da aplicação para o qual foram desenvolvidos e podem possuir diferentes atributos, alguns essenciais e outros opcionais (Caglayan e Harrison, 1997).

Para além de ter a característica de executar um conjunto de tarefas delegadas por um utilizador ou por outro agente, um agente deve possuir os seguintes atributos essenciais:

- **Autonomia**: o agente deve poder realizar as tarefas que nele foram delegadas, sem intervenção directa dos utilizadores e com base no conhecimento armazenado no ambiente sobre o qual actua. Um agente autónomo tem capacidade de planear e de executar as acções planeadas;
- **Reactividade**: o agente deve ser capaz de analisar o seu ambiente, com vista a poder responder em conformidade com as mudanças percebidas. Os agentes reactivos baseiam a sua actuação em três componentes principais: percepção, acção e comunicação;
- **Pró-actividade**: o agente deve ter capacidade de tomar a iniciativa na realização de determinadas tarefas, autonomamente e de acordo com objectivos, não actuando apenas por reacção às alterações do seu ambiente;
- **Persistência**: o agente deve manter o seu estado interno durante a sua existência (continuidade temporal);
- **Sociabilidade** (comunicabilidade): o agente deve interagir com o utilizador ou com outros agentes, quer para receber as tarefas a realizar, quer para comunicar sobre o estado ou o resultado das tarefas.

Alguns dos atributos opcionais que podem ser adicionados a um agente são:

- **Mobilidade**: capacidade de um agente se mover através de uma rede de computadores de forma a efectuar as suas tarefas e cumprir os seus objectivos (agentes móveis);
- **Intencionalidade**: se um agente tiver capacidade de raciocinar sobre uma base de conhecimento, para além de perceber o ambiente, agir sobre ele e comunicar o resultado da sua acção, então é um agente intencional ou cognitivo;
- **Aprendizagem**: capacidade de um agente aprender com a experiência, independentemente de se tratarem de sucessos ou fracassos. Tal como a autonomia,

este atributo assume particular importância no âmbito dos agentes inteligentes. Ao longo da sua existência, estes agentes vão reconhecendo modelos de comportamento e de preferências que lhes permitem actualizar a sua base de conhecimento;

- **Cooperação**: Capacidade que vários agentes têm de trabalharem em conjunto com vista a atingirem um objectivo comum. Para que os agentes possam cooperar é necessária a existência de uma linguagem de comunicação entre agentes (por exemplo, *Knowledge Query and Manipulation Language*). Quando os agentes necessitam de comunicar e cooperar para resolver problemas justifica-se o desenvolvimento de ambientes multiagente. A característica essencial da abordagem dos sistemas multiagente é a filosofia de resolução distribuída de problemas, na qual é adoptada uma estratégia de dividir para conquistar (Frigo et al., 2004).

A veracidade (capacidade de não comunicar informação falsa), a racionalidade (capacidade de não aceitar objectivos impossíveis de concretizar), a benevolência (capacidade de aceitar tarefas ou objectivos de outros agentes, desde que não sejam incompatíveis com os seus), a flexibilidade (capacidade de executar as acções no momento e na sequência que melhor se ajustarem às mutações do ambiente), a adaptabilidade (capacidade de re-planear as acções, caso a execução das acções planeadas falhe), a inteligência (capacidade de negociar efectivamente com ambiguidades) e a confiabilidade (capacidade de desempenhar, sem falhas, uma função requerida sob determinadas condições por um dado período de tempo), entre outros atributos, poderão estar também presentes num agente de acordo com o grau de funcionalidade que se lhe pretende atribuir.

5.1.2- Classificação dos Agentes

Ao longo da última década surgiram várias abordagens de classificação dos agentes com base nos seus atributos.

Uma abordagem genérica classifica os agentes em: **de software** ou **de hardware**, **estacionários** ou **móveis**, **persistentes** ou **temporários**, **reactivos** ou **cognitivos** (Ferber, 1999). Os agentes estacionários são agentes de software que não têm a habilidade de se moverem pela rede para outros computadores. Agentes móveis são agentes de software que têm a capacidade de se mover para outros ambientes através da rede, levando consigo seus

estados internos, ou seja, a sua representação e memória (Rabelo, 2002). Agentes persistentes são agentes de software que não podem ser excluídos do sistema. Agentes temporários são agentes de software que têm uma vida finita, geralmente de duração igual ao tempo de uma determinada tarefa, ou seja, finalizada a sua missão são excluídos do sistema, normalmente por eles próprios (Ferber, 1999). Os agentes reactivos baseiam-se em modelos de organização biológica ou etológica, como por exemplo, as sociedades de formigas ou térmites. Uma formiga por si só não constitui uma entidade inteligente, mas o formigueiro sim, pois apresenta comportamento inteligente aquando da recolha e armazenamento de alimentos, bem como da organização dos berçários. O modelo de funcionamento de um agente reactivo é o de estímulo-resposta. Genericamente, estes agentes não apresentam memória, não planeiam acções futuras e não comunicam com outros agentes, apercebendo-se das acções dos outros agentes pelas mudanças ocorridas no ambiente. Normalmente existem em grande quantidade no sistema e possuem baixa complexidade (Frigo et al., 2004). Os agentes cognitivos baseiam-se em organizações sociais humanas como grupos, hierarquias e mercados. Estes agentes possuem uma representação explícita do ambiente e dos outros agentes, dispõem de memória e, consequentemente, são capazes de planear acções futuras. Os agentes cognitivos podem comunicar entre si directamente, isto é, os seus sistemas de percepção e de comunicação são distintos, o que não acontece nos reactivos. Normalmente estão em pequena quantidade no sistema e são de média ou alta complexidade. Além disso, requerem sofisticados mecanismos de coordenação e protocolos de alto nível para suporte da interacção (Ferber, 1999). Os agentes cognitivos aumentam a qualidade do sistema sob o ponto de vista pedagógico porque permitem gerar um sistema mais perceptivo com autonomia, flexibilidade, colaboração e adaptação.

Uma outra abordagem classifica os agentes **quanto à mobilidade** (agentes estáticos ou móveis), **quanto à presença de um modelo de raciocínio simbólico** (agentes deliberativos ou reactivos), **quanto à funcionalidade** (agentes de informação e agentes de Internet), **quanto à presença dos atributos primários**: autonomia, cooperação e aprendizagem (agentes colaborativos, agentes colaborativos com capacidade de aprendizagem, agentes de interface e agentes inteligentes) e, finalmente, **quanto à existência de filosofias híbridas** que combinam duas ou mais tipologias no mesmo agente (Giese, 1998).

Com base na classificação de agentes baseada na presença de atributos primários, Nwana (1996) define sete tipos de agentes: agentes colaborativos, agentes de interface, agentes móveis, agentes de informação, agentes reactivos, agentes híbridos e agentes inteligentes.

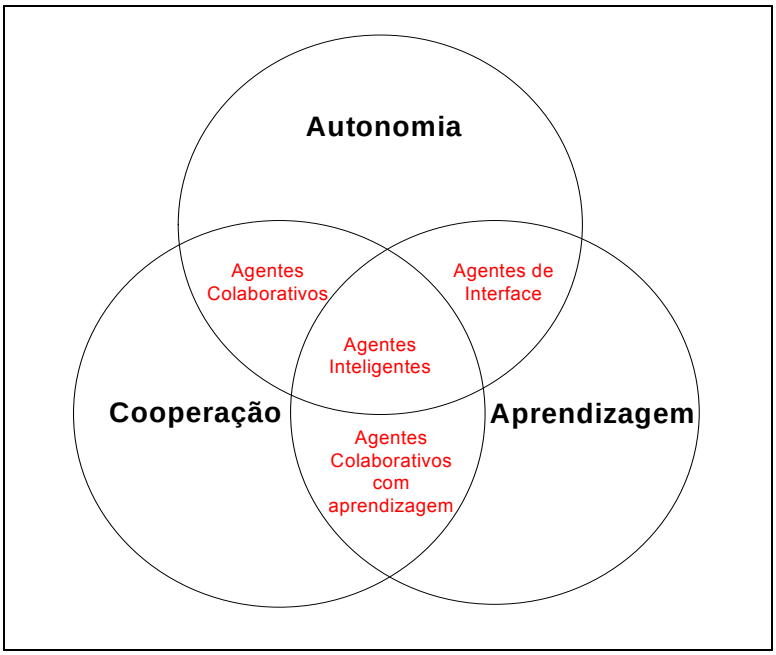


Figura 66 – Classificação de agentes baseada na presença de atributos primários

Finalmente, uma outra abordagem assenta na classificação dos agentes segundo três vectores (Silva, 1999): **ambiente em que actuam** (agentes de *desktop*: agentes de sistema operativo, agentes de aplicação e agentes de pacotes; agentes de Internet: agentes de pesquisa, agentes de filtragem de informação, agentes de recuperação de informações, agentes de notificação, agentes de servidores, agentes de serviço e agentes móveis; agentes de Intranet: agentes colaborativos, agentes de automação, agentes de bases de dados e agentes de recursos), **funcionalidade ou tarefas que realizam** (agentes de venda/compra de produtos, agentes de pesquisa de informação, agentes de acesso a informação e de filtragem de informação) e **representação do conhecimento** (agentes neuronais, agentes de aprendizagem, agentes reactivos e agentes cognitivos ou deliberativos). A existência ou não de representação simbólica de conhecimento é normalmente associada ao grau de inteligência do agente. De um lado, temos os agentes reactivos que não recorrem ou cujo recurso à base de conhecimento é fraco e, do outro, temos os agentes deliberativos ou cognitivos que internamente possuem um modelo de raciocínio e um modelo simbólico, os quais são utilizados para armazenar as situações apreendidas e planear tarefas de acordo com os seus objectivos. Entenda-se agentes inteligentes como sistemas automáticos de software que contêm mecanismos de Inteligência Artificial e que são capazes de tomar decisões e aperfeiçoar o seu desempenho (Berners-Lee et al., 2001).

5.1.3- Agentes de Software para a Internet

Uma vez que o contexto desta investigação é a Web, interessa apresentar alguns exemplos de aplicação de agentes de software para a Internet que surgiram com o intuito de solucionar (ou pelo menos minimizar) alguns dos problemas e limitações que o crescimento da Internet vieram levantar.

Tal como referido anteriormente, a proliferação de conteúdos na Web impulsionou o aparecimento de mecanismos de recuperação de informação que permitem aos utilizadores localizar páginas Web sobre um determinado tema ou assunto. Referimo-nos aos motores de pesquisa e aos directórios.

Normalmente, os directórios agrupam Websites que foram submetidos manualmente. Caso existam colaboradores para verificar a veracidade da informação submetida (nome, título, descrição, palavras-chave, etc), a informação indexada garante autenticidade nos resultados das pesquisas. No entanto, se ninguém submeter uma determinada página Web para esse directório, essa página nunca aparecerá nesse directório.

Um motor de pesquisa usa *robots* ou *spiders*, que percorrem automaticamente a Web seguindo as hiperligações, para indexar páginas Web. Por conseguinte, qualquer página pode ser indexada automaticamente desde que exista uma hiperligação para ela.

Um motor de pesquisa deve ser avaliado principalmente com base na qualidade dos seus resultados. Actualmente, o que fornece melhores resultados é o Google. Por isso, não é de estranhar que seja o motor de pesquisa mais conhecido e utilizado no mundo inteiro (Santos, 2006).

À medida que a Web foi crescendo, para além dos directórios e motores de busca generalistas, foram surgindo directórios e motores de busca temáticos ou especializados, directórios e motores de busca académicos ou de avaliação e, inclusive, directórios de motores de busca e motores de busca baseados nos resultados de outros motores de busca (também designados por meta motores de pesquisa). Os serviços mais relevantes de pesquisa e gestão de informação tentam, por um lado, tirar proveito das duas abordagens fornecendo funcionalidades híbridas de indexação e catalogação e, pelo outro, oferecer directórios ou motores de busca específicos para determinados domínios específicos. Por exemplo, serviços para áreas específicas de negócios ou do saber: ***Froogle*** - serviço de pesquisa específico para comércio electrónico que facilita a procura de informação sobre a venda de produtos e

serviços *on-line* (*http://froogle.google.com/*), ***Google Scholar*** - serviço de pesquisa específico para a educação que facilita a procura de literatura ou recursos educativos, tais como: publicações, teses, livros, sumários e relatórios técnicos de todas as áreas de investigação (*http://scholar.google.com*), ***Google Book Search*** - serviço de pesquisa específico para livros que facilita a sua localização através da indicação do assunto ou tema (por exemplo: *books about Semantic Web*), entre outros. Ou por exemplo, com o intuito de responder mais rapidamente ao utilizador: ***Answers.com*** - motor de pesquisa que dá resposta às questões colocadas filtrando a informação como se se tratasse de uma enciclopédia universal (*http://www.answers.com*) e ***Ask Jeeves*** - dá resposta detalhada para todas as formas de questionar informação existente na Web (*http://www.ask.com*), entre muitos outros. Ou simplesmente com o intuito de dar uma resposta enciclopédica ao termo pesquisado. Para tal, os motores de busca estabeleceram parcerias com enciclopédias ou dicionários *on-line* (por exemplo, parceria *Sapo/Priberam*). Surgiram também exemplos de serviços que disponibilizam a informação consoante a área geográfica pretendida: ***Local*** - serviço de pesquisa que permite procurar negócios e serviços na Internet numa determinada cidade, rua ou localização indicadas (*http://local.google.com*) e ***Tumba!*** (*Temos um motor de busca alternativo!*) - motor de busca que cobre a Web portuguesa e que conta vir a oferecer funcionalidades relacionadas com a exploração do conhecimento do contexto geográfico dos utilizadores e páginas Web, resultantes da integração de resultados de um projecto de investigação denominado *GREASE*. Este projecto incide essencialmente na investigação de um mecanismo de inferência baseado no contexto geográfico e de um mecanismo de busca Web semântico que apresente a informação de acordo com as preferências dos utilizadores e de acordo com a sua proximidade geográfica (*http://www.tumba.pt*). Com vista a guardar as preferências do utilizador, o Google disponibiliza uma outra funcionalidade para guardar, numa área denominada "*My Search History*", as pesquisas que realizámos, permitindo posteriormente acesso directo às mesmas através de um pequeno calendário exibido no canto superior direito das páginas (*https://www.google.com/searchhistory/login*), entre outros serviços específicos disponíveis em *http://www.google.com/intl/en_ALL/help/features.html*.

Independentemente do âmbito dos mecanismos, do tamanho das suas bases de dados, dos critérios de indexação das páginas e das formas de ordenação dos resultados, a pesquisa de informação na Web não deixou de ser entediante já que a análise semântica dos resultados devolvidos continua a ter que ser efectuada pelo utilizador. Não obstante, os exemplos apresentados mostram a tendência da Web para disponibilizar serviços que libertam o

utilizador de tarefas monótonas, repetitivas e morosas. Por conseguinte, a WS e os seus agentes de software apresentam-se como o caminho a seguir na obtenção de motores de busca semânticos que possibilitem pesquisas em linguagem natural, ou seja, motores de busca que nos entendam (o utilizador pode fazer uma pergunta normalmente em vez de usar palavras-chave e operadores boleanos) e que analisem a Web com vista a devolver apenas a informação que melhor satisfaz os nossos requisitos. Por um lado, o facto de um documento conter uma determinada palavra-chave não significa tacitamente que o mesmo deva ser devolvido como resultado e, pelo outro, não conter essa mesma palavra também não pode ser factor de exclusão, uma vez que poderá conter sinónimos ou conceitos relacionados a essa palavra.

Embora ainda haja muito trabalho a fazer para alcançar esta visão, já existem alguns exemplos de motores de busca semânticos. Destacamos o motor de busca **Tropes/Zoom** da empresa francesa Acetic (*http://www.acetic.fr*) e da sua parceira portuguesa *Cyberlex* (*http://www.cyberlex.pt*), pelo facto de ser o primeiro que se conhece a operar em português. O componente *Tropes* é um software de análise de textos, numa perspectiva semântica e pragmática, que garante a fiabilidade das interpretações graças à automatização das análises. O componente *Zoom* assume-se como um motor de busca semântico que indexa e pesquisa em linguagem natural e efectua análise documental baseada na compreensão dos conteúdos a tratar. Recorrendo à estrutura sintáctica e regras gramaticais das diferentes frases, consegue identificar idiomas estrangeiros (alemão, francês, inglês, italiano ou espanhol), distinguir substantivos de adjectivos, conjugar verbos, estabelecer relações entre várias palavras e respectivos significados, resolver ambiguidades e identificar frases consideradas mais fortes, o que lhe permite alcançar a pertinência dos resultados. Para estender a sua capacidade de análise, o *Zoom* está ligado ao *Tropes* e a um agente (*spider*) de recolha de informação na Internet.

Em suma, os agentes de software assumem-se como um dos factores críticos de sucesso mais visíveis no âmbito da iniciativa da WS.

Genericamente, os agentes ao serviço do utilizador Web podem ser qualquer software que filtra e recupera o conteúdo Web para utilizadores. Isto pode incluir *browsers* Web, *players* de media, *plug-ins* e outros programas (incluindo os assistentes ou *wizards* utilizados entre clientes/servidores) que auxiliam o utilizador na procura, filtragem e recuperação dos conteúdos Web (W3C, 2002).

Face ao exposto, podem ser identificados os seguintes agentes de Internet (Coser, 1999):

- **Agentes de pesquisa**: visam a pesquisa de informação na Web (por exemplo: *softbots*, *robots* de informação, *crawlers* e *spiders*, entre outros agentes Web);
- **Agentes de filtragem de informação**: tal como o nome indica, filtram as informações na Internet de acordo com as preferências do utilizador. Alguns destes agentes podem ser uma combinação de agentes de Internet com agentes de *desktop* (filtragem de *cookies*, filtragem de e-mail, *anti-spyware* e bloqueadores de janelas ou *popup killers*, entre outras aplicações Web que personalizam informações, tais como notícias, anúncios ou mensagens, de acordo com critérios previamente introduzidos pelo utilizador);
- **Agentes de recuperação de informação**: permitem adquirir pacotes personalizados de informações de acordo com as preferências do utilizador (por exemplo: agentes de entrega de informação *off-line* inerentes à tecnologia *push* ou mesmo algumas aplicações de gestão de *downloads*);
- **Agentes de notificação**: avisam o utilizador sobre eventos do seu interesse perscrutados da Internet (por exemplo: informam sobre a adição, actualização ou remoção de conteúdos Web, descida do preço de um determinado produto ou serviço, flutuações na cotação de uma empresa na Bolsa, data de aniversário de um indivíduo, etc);
- **Agentes servidores**: correspondem a agentes que fornecem serviços específicos bastando para tal aceder a um endereço fixo na Web (pode tratar-se de um agente que facilite a comunicação entre outros agentes com vista a alcançar a solução do problema e/ou de um agente que faça o atendimento aos clientes de uma empresa que visitam um endereço específico à procura de informações sobre produtos ou serviços);
- **Agentes de serviço**: fornecem serviços especializados aos utilizadores que, geralmente, incluem características de filtragem e/ou notificação (por exemplo: agentes mediadores de procura/oferta de emprego, de compra/venda de produtos, de encontros virtuais entre pessoas ou mesmo gestão de perfis, nomes de utilizador e *passwords*);

- **Agentes móveis**: têm a capacidade de migrar autonomamente entre diferentes nós da rede para executarem a sua tarefa. Normalmente, um agente móvel é um programa enviado por um cliente a um servidor para realizar uma tarefa e que, posteriormente, pode mover-se para outros servidores, informar o cliente dos seus resultados ou simplesmente regressar à origem.

5.1.4- Agentes de Software Inteligentes e Móveis

Embora não exista unanimidade sobre as propriedades de inteligência e mobilidade dos agentes, é geralmente aceite que estas propriedades lhes conferem maior flexibilidade e controlo das suas próprias acções.

Os agentes de software inteligentes correspondem a agentes que possuem um modelo de representação de símbolos (ou estruturas cognitivas) e um modelo de raciocínio, os quais são utilizados para armazenar as situações apreendidas, planear tarefas e tomar decisões de acordo com os seus objectivos. Isto pressupõe que, para além de características essenciais como a autonomia e a sociabilidade, esses agentes inteligentes devem possuir como atributos adicionais a intencionalidade, a cooperação e a aprendizagem de modo a serem capazes de aperfeiçoar o seu desempenho.

Em primeira instância, para falarmos de agentes móveis, teremos que abordar a evolução da computação distribuída. A computação distribuída tem como objectivo tirar partido computacional de diversos computadores interligados por uma rede de computadores para processar colaborativamente determinadas tarefas de forma coerente e transparente, ou seja, como se fosse apenas um único computador central que estivesse a executar essas tarefas.

Inicialmente, a computação distribuída era uma área que impunha desafios extremamente difíceis, complexos e fastidiosos. A normalização através do CORBA (*Common Object Request Broker Architecture*), o provimento de confiabilidade à rede, a melhoria da largura de banda, a máquina virtual de Java permitindo um ambiente de execução com compatibilidade binária para código e dados, as normas para o projecto orientado aos objectos e os diversos projectos de aplicações distribuídas trouxeram melhorias significativas para o projecto de aplicações distribuídas.

Tipicamente, as aplicações de sistemas distribuídos baseiam-se no modelo Cliente/Servidor. Os servidores são responsáveis por disponibilizar e gerir recursos (ambientes aplicacionais, base de dados, ficheiros, etc.). Os clientes servem de interface com os utilizadores e permitem solicitar informação e serviços aos servidores, para além de apresentarem e processarem parte da informação. Para que cliente e servidor possam estabelecer diálogo precisam de falar a mesma linguagem, ou seja, necessitam seguir um protocolo previamente acordado entre eles e suportado por um conjunto de serviços de baixo nível designados por *middleware*, por exemplo: CORBA, ORB (*Object Request Broker*), DCOM (*Distributed Component Object Model*), RPC (*Remote Procedure Calls*) ou RMI (*Remote Method Invocation*) (Gonçalves, 2002).

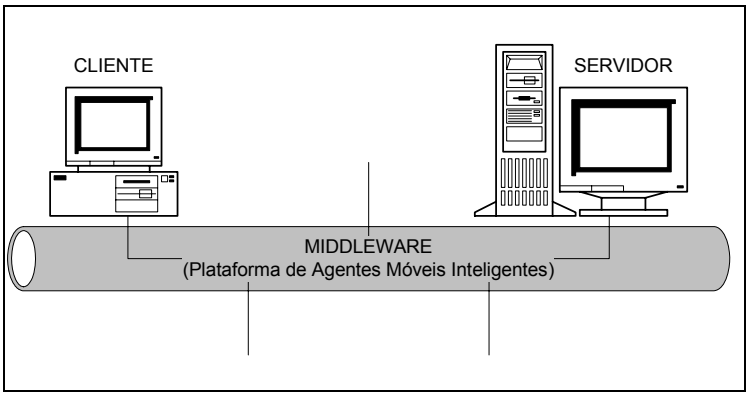


Figura 67 – Sistemas distribuídos baseados em agentes

Existem obviamente problemas de desempenho, no que diz respeito ao ponto onde os recursos da plataforma estão em execução: memória, capacidade de processamento (CPU) e largura de banda. Os processos cliente são executados em máquinas remotas que comunicam com o servidor para executarem as tarefas, pelo que podem surgir congestionamentos derivados do elevado tráfego na rede. Assumir as plataformas de desenvolvimento e execução de agentes móveis inteligentes como *middleware* favorece a redução do tráfego na rede e, consequentemente, resolve substancialmente os problemas apresentados, uma vez que o pedido do cliente é transportado para o servidor e as interacções podem passar a realizar-se automaticamente e de forma assíncrona.

Os agentes móveis correpondem a agentes que têm a capacidade de se moverem numa rede de computadores, migrando de computador em computador até cumprirem o seu itinerário ou os seus objectivos. Para além das características base dos agentes, a característica mais relevante deste tipo de agentes é a mobilidade.

A mobilidade de agentes resultou da evolução das anteriores tecnologias de movimento de código: chamada a procedimentos remotos (RPC – *Remote Procedure Calls*), migração de processos e execução remota (REV – *Remote Evaluation*) (Oliveira, 2003). Estas tecnologias permitiram o aparecimento de aplicações distribuídas mais elaboradas e baseadas em arquitecturas de objectos distribuídos, onde aplicações orientadas a objectos lidam directamente com referências a objectos em processos remotos.

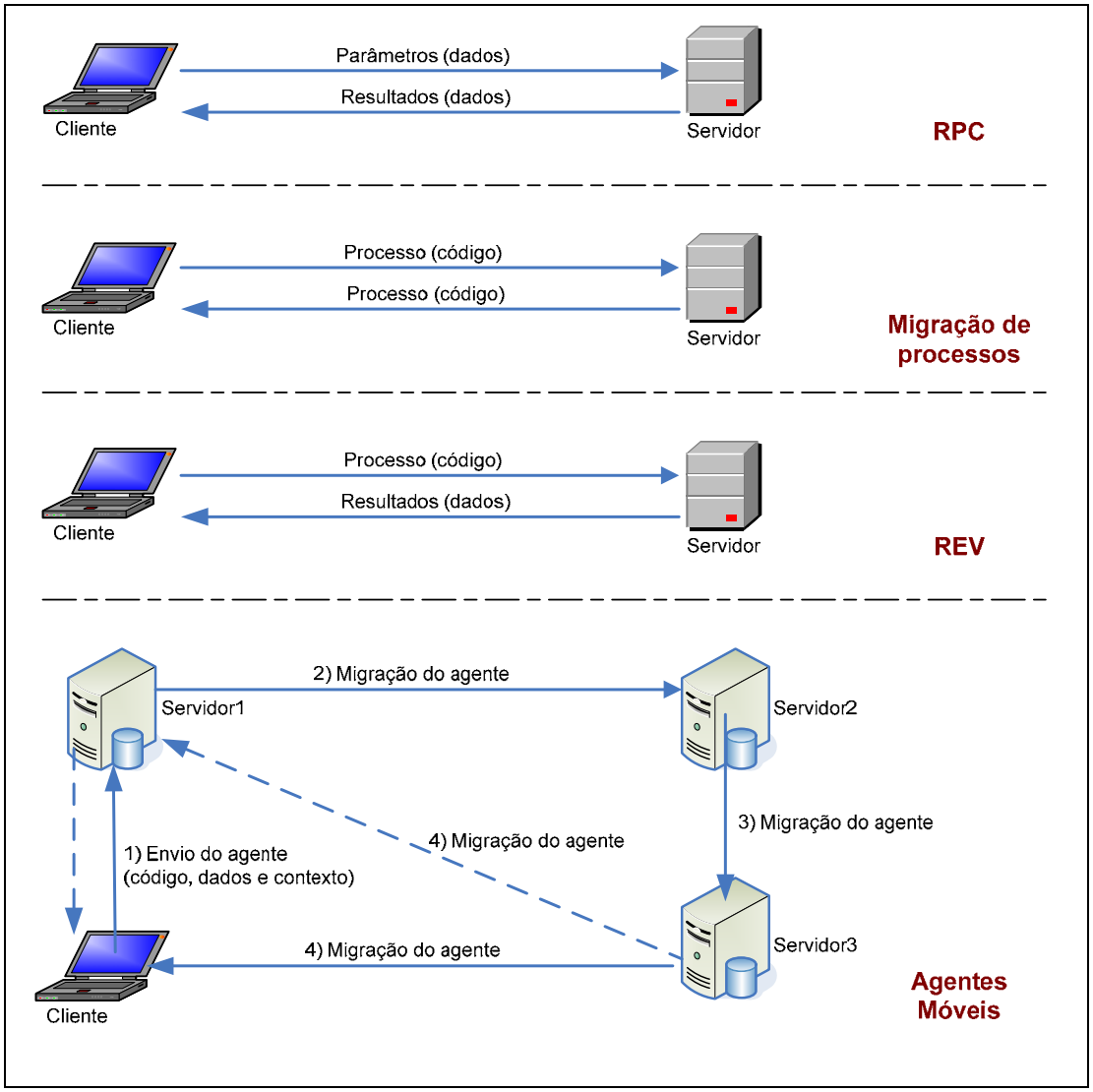


Figura 68 – Evolução do paradigma dos agentes móveis

Na tecnologia RPC, o cliente transmite os parâmetros ou dados ao servidor através de mensagens, usando chamadas a procedimentos remotos, e fica suspenso a aguardar os resultados. Na migração de processos, o código é transmitido nas duas direcções. Na REV, o cliente envia o código do seu próprio processo ao servidor e pede-lhe para o executar e devolver os resultados. Em contraste, um agente móvel é uma entidade programada (código,

dados e contexto) enviada pelo cliente ao servidor e que, posteriormente, pode migrar para outros servidores de acordo com um percurso previamente estabelecido ou determinado pelo próprio agente, transmitir informação à sua origem ou retornar ao cliente. Portanto, um agente móvel pode determinar o que fazer a seguir e não tem que obrigatoriamente retornar dados ao cliente, logo tem mais autonomia que as tecnologias antecessoras de movimento de código, tal como se pode observar na Figura 68.

As características dos agentes móveis permitem: reduzir o tráfego na rede, aumentar a execução assíncrona entre clientes e servidores, aumentar a robustez das aplicações e superar a latência da rede, adicionar funcionalidades específicas aos clientes dos servidores e actualizar dinamicamente os servidores.

Uma das aplicações que podem tirar proveito das capacidades dos agentes móveis é, sem dúvida, a pesquisa e recuperação de informação. Se as aplicações responsáveis por recuperar remotamente informação de servidores forem construídas usando agentes móveis, os agentes podem ser executados no servidor e aceder aos dados sem usar a rede. Apenas os resultados filtrados atravessarão a rede para serem apresentados ao utilizador, o que leva a uma redução dos requisitos de largura de banda por parte da aplicação.

Os agentes inteligentes e os agentes móveis assumem-se como a componente que concretiza a visão da WS, permitindo que a Internet evolua de um espaço estático, cuja utilização requer uma grande intervenção humana, para um espaço mais complexo e dinâmico caracterizado por um crescente recurso a agentes de software e, consequentemente, pela diminuição do número de horas despendidas em tarefas pouco criativas e monótonas por parte do utilizador final. Assim, a pesquisa, filtragem e acesso a informação, a procura e memorização de eventos, a organização da agenda, a organização de planos de férias, a organização de planos de formação, a marcação de reuniões ou outros compromissos que envolvam vários intervenientes, a gestão de contas bancárias, a gestão de correio electrónico e de outro expediente pessoal, a compra/venda de produtos, a gestão de processos de negócio e a coordenação de actividades em grupo (*workflow*), entre outras tarefas, podem ser realizadas total ou parcialmente por agentes de software desde que a informação de que necessitam esteja num formato semanticamente legível para os agentes.

Cada agente terá atributos e características que utilizará individualmente, comportando-se como um assistente pessoal, ou que combinará com outros agentes formando um sistema multiagente (vários agentes inteligentes a actuar de forma integrada e

cooperativa). Geralmente, cada agente possui conhecimentos próprios e diferentes. Estes agentes interagem entre si, partilhando informação e conhecimento para a resolução de problemas mais complexos, os quais dificilmente seriam resolvidos por qualquer um dos agentes isoladamente, mesmo tratando-se de agentes móveis que percorrem de forma independente a rede. Nesta perspectiva, os sistemas multiagente (estacionários ou móveis) apresentam claras vantagens na resolução dos problemas Web actuais, tanto mais que o uso das filosofias "a união faz a força" e "dividir as tarefas para conquistar os objectivos" permitem obter resultados de forma mais rápida e precisa. Mas, não esqueçamos que os agentes inteligentes (móveis ou estacionários) não pensam, apenas dão a sensação de ser inteligentes graças às linguagens e protocolos que utilizarão as páginas para a WS.

5.2- Sistemas de Agentes Móveis

Quando comparado com os paradigmas tradicionais da computação distribuída, a utilização de agentes móveis nas aplicações acarreta diversas vantagens. O facto de podermos transportar código para perto dos dados, distribuindo o processamento pelos servidores de recursos de informação, pode melhorar o desempenho de várias aplicações.

Assim, podemos encarar as redes como um conjunto de ambientes multiagente compostos por agentes móveis capazes de migrar de um computador para outro, de controlar as suas acções e de reagir às mudanças do ambiente em que estão a ser executados (Lange e Oshima, 1998).

5.2.1- Arquitectura Genérica de um Sistema de Agentes Móveis

Um sistema de agentes móveis (SAM) é uma plataforma de software que disponibiliza os mecanismos de mobilidade e de comunicação entre agentes móveis. O SAM tem como objectivo principal fornecer suporte para a criação, activação, desactivação e gestão de

agentes, incluindo mecanismos para migração, comunicação, persistência, segurança, monitorização e tolerância a falhas.

Genericamente, os principais componentes de um SAM são:

- **Agente**: corresponde ao agente móvel que mais não é do que uma entidade com os seguintes atributos:
  - Identificador: identificação do agente móvel que o distingue inequivocamente de todos os outros, permitindo o seu reconhecimento e localização enquanto se desloca;
  - Estado: permite que o agente retome a execução ao longo do seu trajecto;
  - Implementação: corresponde ao código que o agente vai executar no destino, ou seja permite a execução do agente independentemente da sua localização;
  - Interface: é um conjunto de métodos que permitem a comunicação de e com o agente;
  - *Principals*: permite que seja determinada a responsabilidade legal e moral do agente (é como que uma referência ao seu dono).
- **Local de execução** (*place*): local de execução de cada agente. O ambiente de execução corresponde a cada um dos locais para os quais o agente se desloca. Logo, os locais também devem estar identificados de forma unívoca (endereçamento de rede).

Portanto, com base nestes componentes, um SAM fornece o suporte necessário para a criação e a destruição dinâmicas de agentes, para a persistência (activação/desactivação temporária de agentes num determinado local) e para a migração (transferência ou mobilidade dos agentes independentemente do local).

A interoperabilidade entre as aplicações e entre os seus agentes é um requisito essencial nos SAMs. Assim, no contexto particular do desenvolvimento de aplicações de agentes, destacam-se dois tipos de linguagens:

- **Linguagens de programação de agentes:** linguagens interpretadas (por exemplo: Java, Tcl/Tk, Telescript, Obliq, Phantom, Python, Prolog) ou compiladas (por exemplo: C, C++, Smalltalk, Pascal) que permitem a implementação de agentes. De referir que para que possamos usufruir da construção de agentes móveis, teremos que recorrer a linguagens interpretadas. Destas linguagens, destaca-se a linguagem de

programação Java uma vez que disponibiliza uma arquitectura neutra (ideal para ambientes como a Internet), características distribuídas possibilitando a implementação de agentes para uma rede TCP/IP e poderosas ferramentas para programadores (que facilitam a construção de agentes visto que a programação é orientada a objectos), entre outras características que justificam o seu actual sucesso.

- **Linguagens de comunicação entre agentes:** linguagens específicas que tentam normalizar vocabulários através dos quais os agentes possam comunicar entre si (AgentTalk, AOP, ACL, KIF, KQML). A linguagem AgentTalk permite a descrição de protocolos de coordenação para sistemas multiagente. A linguagem ACL - *Agent Communication Language* é uma das mais conhecidas, baseia-se na abordagem declarativa e a sua sintaxe é representada em XML. Para além de um vocabulário próprio, engloba a linguagem de representação do conhecimento KIF (*Knowledge Interchange Format*) e a linguagem de comunicação entre agentes KQML (*Knowledge Query and Manipulation Language*). KQML especifica um conjunto de primitivas que definem os objectivos e as tarefas que os agentes podem realizar sobre o conhecimento.

De acordo com o exposto acima, Oliveira (2003) apresentou uma arquitectura genérica para os SAMs baseados na linguagem de programação Java constituída por seis componentes principais: um gestor de agentes, um gestor de comunicação entre agentes, um gestor de segurança, um gestor de persistência, uma interface para a aplicação e um gestor de directórios, tal como se pode observar na Figura 69.

O gestor de agentes é responsável pelo envio para máquinas remotas e pela recepção de agentes para serem executados na máquina local.

O gestor de comunicação entre agentes faculta a comunicação entre os agentes móveis em circulação na rede.

O gestor de segurança é responsável por autenticar o agente antes de este ser executado. Assim, é protegida a máquina e os agentes móveis contra acessos não autorizados.

O gestor de persistência garante a robustez do SAM, salvaguardando os agentes de possíveis falhas do sistema ou do servidor e assegurando que os agentes cheguem ao destino.

A interface para a aplicação corresponde ao ponto de interacção dos agentes com os servidores de aplicações, tais como bases de dados.

Finalmente, o gestor de directórios permite que os agentes identifiquem a localização de um servidor através de um serviço de nomes e de directório, por exemplo *Java Naming and Directory Interface* (JNDI), para que possam migrar facilmente para a máquina correspondente a esse servidor.

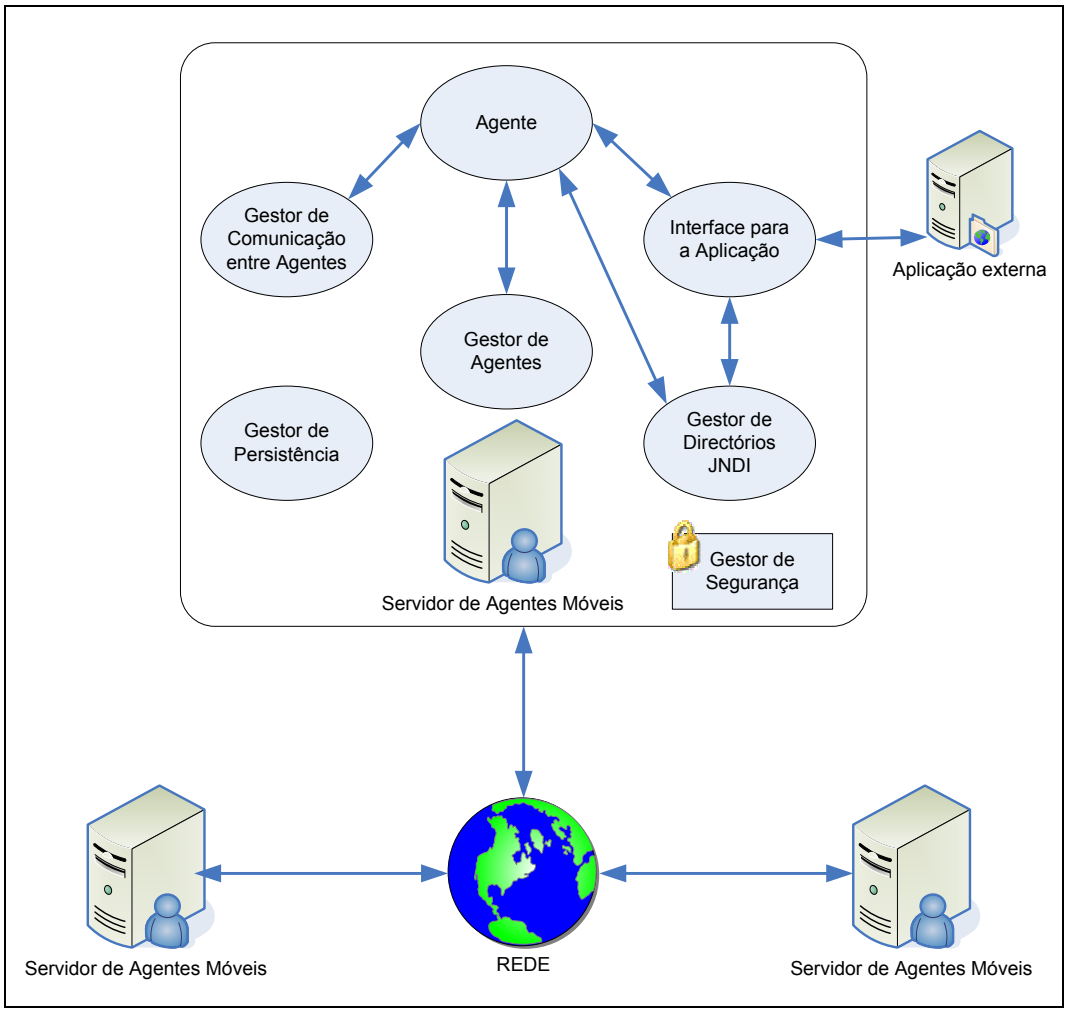


Figura 69 – Arquitectura genérica de um sistema de agentes móveis

5.2.2- Infra-estruturas para a Criação de Aplicações Baseadas em Agentes

Com vista ao desenvolvimento de aplicações baseadas em agentes têm vindo a surgir diversas ferramentas para a construção de agentes e infra-estruturas ou plataformas de agentes móveis, tanto fruto de projectos ou investigações científicas como de iniciativas de I&D comerciais.

O anexo F apresenta sucintamente algumas das ferramentas, plataformas e infra-estruturas no âmbito do desenvolvimento de sistemas multiagente, nomeadamente de agentes inteligentes e de agentes móveis. Mas a escolha das linguagens, tecnologias ou infra-estruturas para a construção de agentes depende do tipo de aplicação que se pretende desenvolver. Sem querermos contrariar a premissa de que a tecnologia deve ser escolhida para cada situação particular e não o contrário, uma questão que se pode levantar relaciona-se com os protocolos e serviços na camada *middleware*: RMI não é mais do que um conjunto de protocolos que permitem que objectos de Java comuniquem remotamente com outros objectos de Java. RMI é relativamente simples, mas lida apenas com objectos Java, enquanto que CORBA e DCOM são protocolos mais complexos projectados para suportar objectos criados em qualquer linguagem. Por conseguinte, nem todas as plataformas se adequam a todas as situações.

Com base na descrição comparativa efectuada por Oliveira (2003), a Tabela 5 destaca algumas das características das plataformas de agentes móveis.

Plataformas	Identificação do agente	Mobilidade do agente	Comunicação
AgentTcl ou D'Agents	Dependente da localização (baseada no DNS).	Mobilidade absoluta (*agent_jump*). Um clone do agente é enviado através do método *agent_fork.*	Envio de mensagens usando os métodos *agent_send* e *agent_receive*. Comunicação baseada em *streams* usando *agent_meet* e *agent_accept.*
Aglets ou ASDK	Dependente da localização (URLs baseados em nomes DNS).	Mobilidade absoluta (*dispatch*). Suporta abstracção com *Itinerary*.	Envio/recepção de objectos *Message*. Suporta comunicação síncrona, *one-way* e *future-reply*.
Concordia	Dependente da localização (baseada no DNS).	Mobilidade absoluta, baseada no conteúdo do *Itinerary* do agente.	Invocação a métodos locais. Possibilidade de *multicast* usando o construtor *AgentGroup*.
Tacoma	Dependente da localização (baseada no DNS).	Um único método (*meet*) suporta os dois tipos de mobilidade (absoluta e relativa).	Os agentes podem encontrar-se no mesmo servidor e trocar dados usando o método *meet.*
Telescript	Dependente da localização (baseada no DNS).	Mobilidade absoluta (*go*) e mobilidade relativa (*meet*).	Invocação a métodos locais.
Voyager	Identificador global e independente da localização, e *proxies* locais.	O método (*moveTo*) suporta os dois tipos de mobilidade (absoluta e relativa).	Invocações síncronas, *one-way*, *future-reply* e *multicast.* Suporta RMI, CORBA e DCOM.

Tabela 5 – Plataformas de agentes móveis

As plataformas de agentes móveis referidas diferem substancialmente quanto à arquitectura e implementação. Portanto, a falta de normalização no que diz respeito à interoperabilidade entre as plataformas de agentes de diferentes fabricantes poderia ser um problema. Com vista a solucionar este problema, o grupo OMG (*Object Management Group*) propôs a norma MASIF (*Mobile Agent System Interoperability Facility*). MASIF normaliza a gestão de agentes, a mobilidade ou transferência de agentes, os nomes dos sistemas de agentes e dos seus agentes, os tipos de sistemas de agentes e a sintaxe da sua localização.

Os agentes inteligentes devem possuir uma base de conhecimentos. Cada aplicação constrói a sua própria base de conhecimentos e utiliza linguagens e metodologias particulares. Com o projecto da WS, a normalização RDF/XML e OWL promovida pelo W3C favorece a compatibilidade e interoperabilidade necessária entre fabricantes para partilha de conhecimentos através de agentes.

5.3- Resumo

Este capítulo apresentou sucintamente a tecnologia de agentes com vista a expor os principais conceitos sobre agentes móveis e inteligentes, no âmbito da especificação, construção e utilização de sistemas multiagente.

Os agentes inteligentes e os agentes móveis assumem-se como a componente que concretiza a visão da Web Semântica, permitindo que a Internet evolua de um espaço estático, cuja utilização requer uma grande intervenção humana, para um espaço mais complexo e dinâmico, caracterizado por um crescente recurso a agentes de software e, consequentemente, pela diminuição do número de horas despendidas em tarefas pouco criativas e monótonas por parte do utilizador final.

Tradicionalmente, as aplicações de sistemas distribuídos baseiam-se no modelo Cliente/Servidor. Os processos cliente são executados em máquinas remotas que comunicam com o servidor para executarem as tarefas, pelo que podem surgir congestionamentos derivados do elevado tráfego na rede. Assumir as plataformas de desenvolvimento e execução de agentes móveis inteligentes como *middleware* favorece a redução do tráfego na rede e,

consequentemente, pode resolver as limitações dos tradicionais sistemas distribuídos, uma vez que o pedido do cliente é transportado entre servidores e as interacções podem passar a realizar-se automaticamente e de forma assíncrona.

Uma das aplicações que podem tirar proveito das capacidades dos agentes móveis é, sem dúvida, a pesquisa e recuperação de informação na Web. Neste sentido, foi apresentada a arquitectura genérica de um sistema de agentes móveis. Para implementar essa arquitectura foram apresentadas sucinta e comparativamente algumas das principais infra-estrutruras para a criação de aplicações baseadas na tecnologia de agentes, tendo-se verificado que a grande maioria usa a linguagem de programação Java para a implementação de agentes.

Em suma, com este capítulo concluímos a apresentação das tecnologias para a Web Semântica, para o e-Learning e para os agentes que integrámos no planeamento e desenvolvimento do sistema WSE proposto nesta tese.

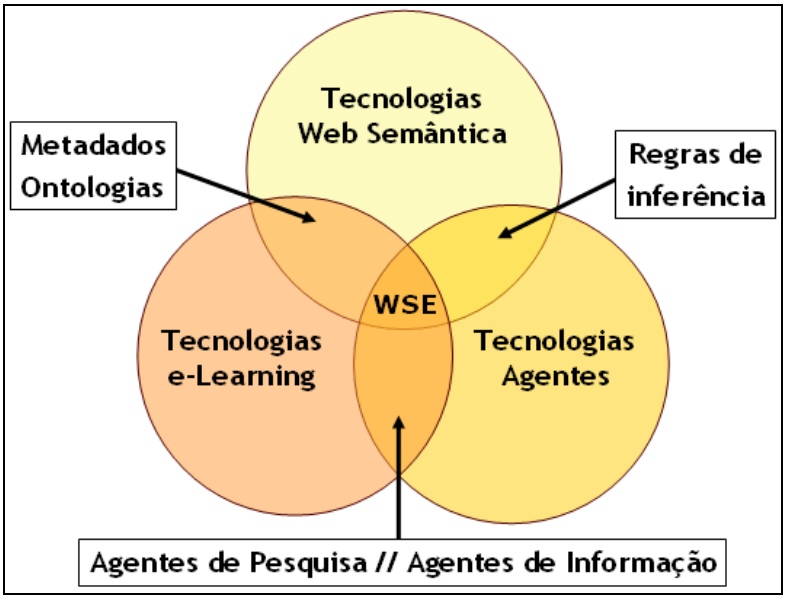


Figura 70 – Integração das Tecnologias para a Web Semântica, e-Learning e agentes

6- Arquitectura do Sistema WSE

O sucesso da Internet deve-se essencialmente à crescente utilização da Web como principal serviço de comunicação e disseminação de informação. O seu crescimento envolve todas as áreas de actividade, desde a institucional à comercial e à educativa, podendo-se encontrar todo o tipo de informação na Web, o que obriga os utilizadores a despender demasiado tempo para encontrar o que procuram. Mas, se a informatização dos documentos e a sua distribuição na Web visam facilitar a consulta e introduzir rapidez no processo de busca, porque é que isso não acontece? Esta foi uma das questões à qual tentámos responder nos capítulos anteriores, apresentando novos paradigmas para a solucionar.

A Web é uma biblioteca global desordenada. Exige-se a ordenação deste caos. A visão da Web Semântica apresenta-se como a iniciativa mais viável para o conseguir. Para tal, as máquinas têm que entender o conteúdo de uma página Web, interpretá-lo, analisá-lo e classificá-lo de acordo com critérios ou ontologias para que o utilizador, ao realizar uma pesquisa, obtenha informação o mais pertinente possível para a sua procura.

Actualmente, as instituições educativas recorrem à sua Intranet ou à Web para publicar/recolher informações de suporte à tomada de decisão ou para actividades de formação, quer através de Websites gerais, quer através de sistemas de e-Learning ou portais temáticos. O uso de diferentes linguagens e formatos para gerir esse conhecimento em diferentes níveis da organização, torna muitas vezes difícil integrar os dados de forma a gerar facilmente nova informação.

Neste sentido, a finalidade deste capítulo é propor um modelo ou arquitectura para um sistema de recuperação de recursos educativos, considerando os cenários que a visão da Web Semântica poderá originar no e-Learning, os conceitos e fundamentos apresentados, o estado da arte das tecnologias e especificações destes dois campos e, claro, as actividades de planeamento e de análise e especificação de requisitos relativas ao nosso caso de estudo.

O desenvolvimento de um sistema de informação deverá entender-se como um processo de mudança que visa melhorar o desempenho de um (sub)sistema de informação (Carvalho e Amaral, 1993). Para tal, as actividades de análise e de especificação do sistema permitem descrever o domínio do sistema e os seus principais requisitos e funcionalidades. Assim, este capítulo apresenta a modelação do contexto e dos requisitos, a modelação do comportamento e da estrutura e, finalmente, a modelação da arquitectura do sistema.

6.1- Planeamento e Requisitos

O planeamento de um sistema de informação é a actividade da organização onde se define o futuro desejado para esse sistema, para o modo como deverá ser suportado pelas TICs e para a forma de concretizar esse suporte (Carvalho e Amaral, 1993).

A actividade de planeamento deverá conduzir a uma representação da visão global do sistema de informação e, simultaneamente, incluir os elementos necessários para a sua operacionalização. O plano resultante deverá ser construído na procura simultânea da satisfação dos utilizadores e de um correcto suporte e tratamento das influências entre a organização e o seu sistema de informação.

A arquitectura do sistema que propomos neste capítulo poderá ter aplicação tanto no âmbito específico de uma universidade, de um instituto politécnico ou mesmo de um agrupamento de escolas dos ensinos básico e secundário, como no âmbito mais geral, oferecendo serviços de localização, recuperação e reutilização de objectos de aprendizagem para suportar actividades de auto-estudo ou auto-aprendizagem.

Neste sentido, esta secção apresenta sucintamente os modelos que orientaram as actividades de planeamento e de análise e especificação dos requisitos do sistema Web Semântica Educativa (WSE).

6.1.1- Actividade de Planeamento

O desenvolvimento de um sistema de informação para a Web deve ser impulsionado pelo planeamento do sistema e, posteriormente, validado pelos utilizadores. Primeiramente, impõe-se responder às seguintes questões: “Onde estamos?” (análise estratégica), “Para onde queremos ir?” (definição estratégica), “O que fazer para lá chegar?” (implementação da estratégia) (Amaral, 1994; Gonçalves, 2002). As reflexões inerentes a estas questões encontram-se sintetizadas na Figura 71 e foram suficientemente expostas ao longo dos capítulos anteriores desta tese. Posteriormente, com base na análise do contexto, na identificação inicial dos problemas e dos requisitos para o contexto educativo e no estado da arte das tecnologias emergentes, concluímos que a integração da visão da Web Semântica nos

ambientes virtuais de aprendizagem seria o caminho mais viável para satisfazer os requisitos iniciais identificados.

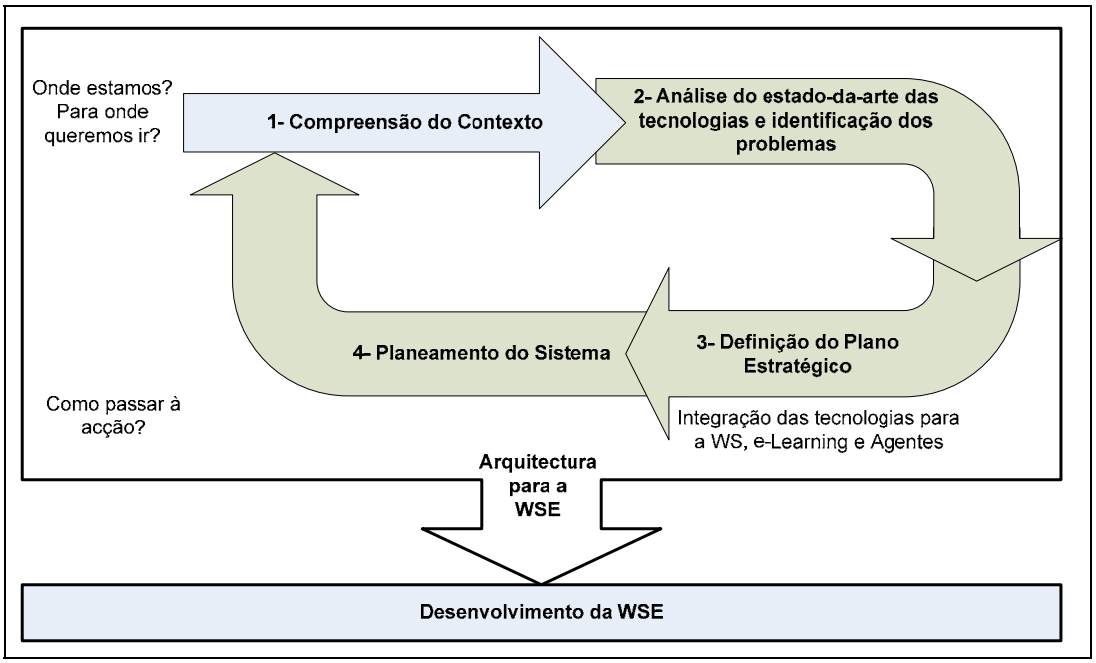


Figura 71 – Modelo genérico da actividade de planeamento

A análise do contexto e dos principais problemas e requisitos iniciais, bem como a estratégia a seguir, encontram-se sintetizados no texto abaixo. Genericamente, identificámos o contexto e os factores que influenciam as instituições educativas e utilizadores em geral, no que diz respeito à localização e recuperação de conteúdos educativos para suportar actividades de auto-aprendizagem.

A Web constitui a maior base de conteúdos educativos para professores e alunos, oferecendo oportunidades de geração de novo conhecimento que nem sempre é possível concretizar dadas as dificuldades de localização das fontes de informação.

A formação ao longo da vida assume-se como um mecanismo crítico de suporte à competitividade organizacional não só do ponto de vista da Educação contínua, mas também do ponto de vista da nova Economia (Drucker, 2000). A velocidade e volatilidade dos mercados actuais requer processos de aprendizagem rápidos, no momento em que são necessários, no local de trabalho ou em casa, personalizados, relevantes e de acordo com os requisitos de quem os procura (*fast, just-in-time, at-work-place, at-home, customized, relevant, on-demand*) para suportar a geração de conhecimento (*know-how*) nas organizações e seus empregados. O contexto organizacional exige que os processos de aprendizagem sejam

integrados no dia-a-dia dos empregados, desenvolvendo uma cultura "aprende de manhã, aplica à tarde".

Genericamente, podemos identificar duas formas de auto-aprendizagem:

- Através de conteúdos Web educativos dispersos;
- Através de conteúdos educativos de cursos de e-Learning.

Os conteúdos Web educativos dispersos podem ser recuperados através de motores de pesquisa ou de directórios para usar em actividades de auto-formação. Tal como referimos, existem na Web milhares de conteúdos ou recursos de aprendizagem de qualidade disponibilizados gratuitamente por investigadores, professores, alunos e entidades do saber (universidades, escolas, bibliotecas, editoras, centros de documentação, museus, centros de I&D, etc.). No entanto, o acesso aos mesmos é cada vez mais um processo que exige um esforço demasiado grande e moroso, uma vez que o utilizador despende muito tempo nas tarefas de pesquisa, filtragem, indexação e compilação de conteúdos com vista à organização do seu plano de auto-formação. Pese embora os esforços das empresas responsáveis pelos motores de pesquisa e directórios e as dicas de profissionais para a optimização das procuras (*search engine positioning* para posicionar as páginas no topo dos motores de pesquisa através de constantes actualizações das páginas; estruturar as páginas Web na perspectiva dos *robots* ou *spiders*, evitando *frames* e outros elementos que desorientem esses agentes de pesquisa; utilização das meta *tags* escolhendo as palavras-chave certas para identificar o tema ou assunto da página Web (Santos, 2007)), a verdade é que a vida do utilizador na procura de recursos de aprendizagem é cada vez mais um calvário.

Nestes últimos dois ou três anos o problema agravou-se significativamente face à evolução da Web para um estado onde os conteúdos são cada vez mais gerados pelos utilizadores. Falamos mais concretamente da evolução que tem vindo a ocorrer na Web desde 2004 e que ficou conhecida como Web 2.0 (termo derivado de uma afirmação utilizada pela *O'Reilly Media* numa conferência em 2004). Esta nova realidade pretende identificar a mais recente geração de serviços baseados na Internet, cujo expoente máximo pode ser encontrado nas redes sociais colaborativas *on-line* formadas por blogs, wikis e demais ferramentas de comunicação interactiva. De forma simplificada, isto quer dizer que o utilizador abandona uma posição passiva de mero consumidor de conteúdos para se tornar um utilizador activo, produtor e distribuidor de conteúdos que alimentam a rede (Infante e Oliveira, 2007).

Quanto aos conteúdos educativos de cursos de e-Learning referimo-nos essencialmente a e-cursos de acesso livre que qualquer indivíduo pode usar na sua auto-formação. Tal como foi mencionado anteriormente, as vantagens dos cursos de e-Learning são evidentes no âmbito da formação contínua. No entanto, nem sempre é possível ou está acessível um e-curso à medida das reais necessidades do utilizador. Embora existam sistemas desenvolvidos à medida, a grande maioria dos sistemas de e-Learning disponíveis foram implementados usando soluções comerciais (*Blackboard*, *WebCT*, *Lotus*, *Luvit*, etc) ou *open source* (Moodle, Atutor, etc) com vista a permitir a criação, apresentação e distribuição de cursos e respectivos materiais. Todavia, a transposição directa dos conteúdos ministrados no ensino presencial para o ensino a distância não é a solução mais adequada para criar cursos de e-Learning com qualidade e susceptíveis de propiciar os benefícios esperados. Se bem que estas soluções se assumam como plataformas que favorecem a aprendizagem interactiva, tanto a aprendizagem como o nível de gestão da informação acabam por ser limitados pelas tecnologias de baixo nível subjacentes à recuperação de informação. Ou seja, em muitos casos são os docentes ou formadores que têm de recolher, criar, reunir e organizar os conteúdos manualmente e, por sua vez, os alunos sentem cada vez mais dificuldades em obter resultados satisfatórios nas tarefas de pesquisa e relacionamento dos conteúdos, para além do nível de aprendizagem ser constantemente questionado. Por conseguinte, os utilizadores destes ambientes de aprendizagem exigem novas formas de apresentação e recuperação de informação, incluindo o seu relacionamento semântico com a aprendizagem em questão. Portanto, um dos problemas actuais na área do e-Learning é identificar que metodologias devem ser aplicadas à estruturação de conteúdos e as normas de interface a que estes deverão obedecer, com vista a favorecer a sua reutilização por parte dos agentes de software.

Todavia, a Sociedade da Informação e do Conhecimento exige não só mecanismos que tratem da organização e estrutura dos recursos educativos, mas também processos que disponibilizem conteúdos de aprendizagem filtrados (especificidade em vez de generalidade). Tal como vimos, a aplicação da filosofia dos objectos de aprendizagem ao e-Learning permite que cada item seja facilmente organizado em cursos de aprendizagem (se possível projectando a aprendizagem em conformidade com o *Learning Design*) e entregue ao utilizador que os procura de acordo com o seu perfil (Stojanovic, 2001).

Por conseguinte, ambas as alternativas de auto-formação exigem uma visão segundo objectos de aprendizagem devidamente descritos através de metadados, significado

semanticamente explícito e baseado em ontologias que descrevam os seus domínios, com vista a favorecer a procura e combinação dos mesmos em cursos personalizados.

As organizações têm vindo progressivamente a recorrer a vários sistemas de informação para a Web para disponibilizar o acesso à informação. Um outro aspecto a salientar é a entrada em funcionamento dos cursos adequados ao processo de Bolonha. Esta alteração origina a identificação de novos requisitos intrínsecos à necessidade de descobrir conhecimento indispensável para a tomada de decisões, directa ou indirectamente relacionadas com o processo de ensino/aprendizagem. Mas a descoberta de conhecimento nem sempre é uma tarefa fácil, já que o processo de recuperação, análise e agregação de informação é cada vez mais complicado. A grande quantidade de informação e a discrepância existente entre essa informação, a variedade das fontes de informação e a falta de mecanismos de integração dessas fontes, bem como as limitações dos mecanismos de recuperação disponíveis, são alguns dos motivos que afectam negativamente esta situação.

Alguns destes problemas relacionam-se com a ausência de mecanismos de pesquisa que permitam procuras num determinado contexto específico. A forma de procurar informação, disponibilizada pelos motores de busca actuais, através de palavras-chave desprovidas de significado e contexto, leva a que muitas das vezes a informação passe despercebida, seja difícil de localizar ou, simplesmente, esteja inacessível em bases de dados apenas pesquisáveis com aplicações específicas.

A WS abre um vasto conjunto de possibilidades para a descoberta e reutilização inteligentes de recursos de informação, através da partilha de ontologias. Assim, o desempenho dos agentes de software na descoberta e processamento de recursos de informação em geral, e de objectos de aprendizagem em particular, será bem mais fiável. Esta é uma vantagem excepcional quando comparada com o facto de ter pessoas a examinar manualmente milhares de páginas devolvidas por um motor de busca num *browser* que não tem nenhuma compreensão semântica dos dados.

Considerando um docente que quer criar um curso usando objectos de aprendizagem na WS para um grupo de alunos, a tarefa de pesquisa é entregue a um agente, que constrói uma interpretação semântica da questão submetida usando uma ontologia apropriada. O agente move-se pela Rede à procura dos objectos de aprendizagem que respondem à questão com vista a satisfazer os requisitos do utilizador. Pode haver diversos objectos de aprendizagem que estão ligados à mesma ontologia que é usada pelo agente. Por conseguinte,

existirá uma compreensão comum sobre o que é requerido. No entanto, outros objectos de aprendizagem podem também satisfazer os critérios, mas usam ontologias diferentes. Se houver alinhamento entre essas ontologias e a que está a ser usada pelo agente, é possível que alguma compreensão semântica ocorra. Em suma, com a WS é assim possível executar pesquisas mais significativas para objectos de aprendizagem.

6.1.2- Análise e Especificação de Requisitos

Assumindo que o desenvolvimento de um sistema de informação deverá resultar numa melhoria do desempenho de um (sub)sistema de informação, as actividades de análise e especificação dos requisitos constituem os pilares dessa mudança uma vez que reflectem a opinião dos utilizadores de todos os níveis da organização.

A análise de requisitos tem como objectivo determinar e organizar os requisitos para o sistema a desenvolver. As tarefas ou momentos a destacar nesta actividade são a **modelação do contexto** (compreensão do domínio com base nas reflexões emanadas da actividade de planeamento, identificando o que está dentro e fora do contexto do sistema), o **levantamento dos requisitos** (identificação detalhada dos requisitos do sistema e respectiva identificação de prioridades) e a **especificação dos requisitos** (validados os requisitos, há que descrever "o que o sistema deve fazer e como deve fazer", mas não "como fazer o sistema" – isso será detalhado no capítulo VII). Posteriormente, a especificação concentra-se nos aspectos conceptuais do projecto do sistema, apresentando soluções para os problemas e requisitos identificados. A Figura 72 representa as principais etapas da análise e especificação de requisitos no âmbito do desenvolvimento do sistema proposto nesta tese.

Nos processos de desenvolvimento orientado por objectos, tanto os dados como as funções são agregados conjuntamente em objectos que comunicam com outros objectos. Esta característica permite a reutilização de software e uma manutenção mais fácil e rápida, facilitando significativamente a gestão das alterações de requisitos (Gonçalves, 2002).

Actualmente, encarar cada processo como um conjunto de objectos é a filosofia mais adoptada no desenvolvimento de sistemas. Por conseguinte, esta foi a filosofia seguida para a modelação e especificação do sistema que aqui se propõe. Para tal, recorremos aos diagramas disponibilizados pela *Unified Modelling Language* (UML) uma vez que é a linguagem de

modelação normalizada pelo OMG no âmbito da engenharia de software orientada por objectos.

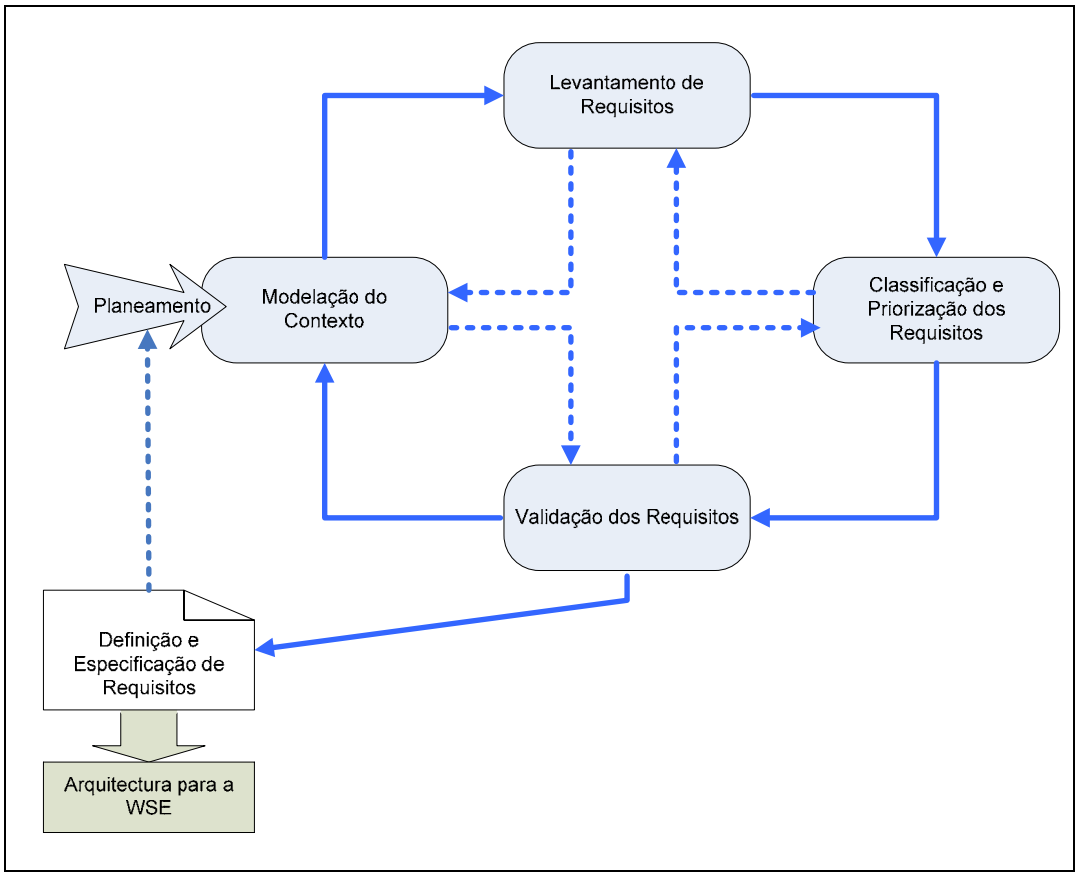


Figura 72 – Modelo genérico da actividade de análise de requisitos

Quando comparada com outras propostas existentes, a UML é extensível e adaptável a diferentes tipos de modelação, tais como a modelação de sistemas distribuídos e de aplicações e serviços baseados na Web – que é o caso do sistema que aqui se propõe. Os principais diagramas que esta linguagem nos oferece permitem-nos:

- **Modelar o contexto e os requisitos**: diagramas de casos de utilização;
- **Modelar o comportamento**: diagramas de casos de utilização, diagramas de estado, diagramas de actividade, diagramas de interacção (diagrama de sequência e diagrama de comportamento);
- **Modelar a estrutura**: diagramas de classes e diagramas de objectos;
- **Modelar a arquitectura**: diagramas de componentes e diagramas de instalação.

Os cinco diagramas da modelação do comportamento permitiram descrever as partes dinâmicas do sistema, enquanto que os quatro diagramas da modelação da estrutura e da arquitectura permitiram descrever as suas partes estáticas.

6.2- Modelação do Contexto e dos Requisitos

Na modelação do contexto os diagramas de caso de utilização permitiram identificar o domínio do sistema e mostrar os actores com os quais interage. Tal como se pode verificar na Figura 73, os actores não são apenas utilizadores, podem ser também agentes de software ou sistemas informáticos, tais como servidores de conteúdos, sistemas de e-Learning, repositórios de objectos de aprendizagem ou motores de pesquisa.

Tal como referimos anteriormente, os directórios e motores de pesquisa apoiam o utilizador na localização de recursos educativos dispersos e os sistemas de e-Learning permitem que os alunos ou formandos interajam socialmente ou com conteúdos publicados pelos professores ou formadores.

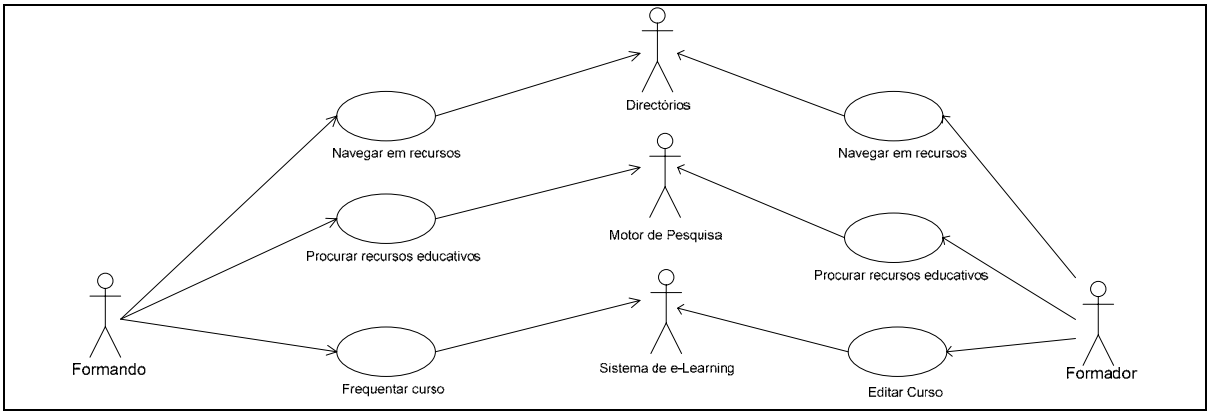


Figura 73 – Modelação do contexto actual

Por conseguinte, exige-se um sistema que forneça uma visão integradora destes sistemas e permita a localização, a navegação e a recuperação de objectos de aprendizagem remotamente, atenuando as limitações actuais.

6.2.1- Casos de Utilização

Com base na modelação do contexto do sistema de informação actual foi possível delimitar as fronteiras do sistema e descrever o que deveria fazer (primeiro, na perspectiva do analista e, posteriormente, na perspectiva dos utilizadores), com vista a facilitar a comunicação com formandos e formadores e identificar os seus requisitos.

Os questionários, a observação das actividades, a promoção de reuniões e a elaboração e validação dos protótipos do sistema foram algumas das formas que permitiram descobrir aquilo que o sistema faz actualmente e deverá fazer futuramente.

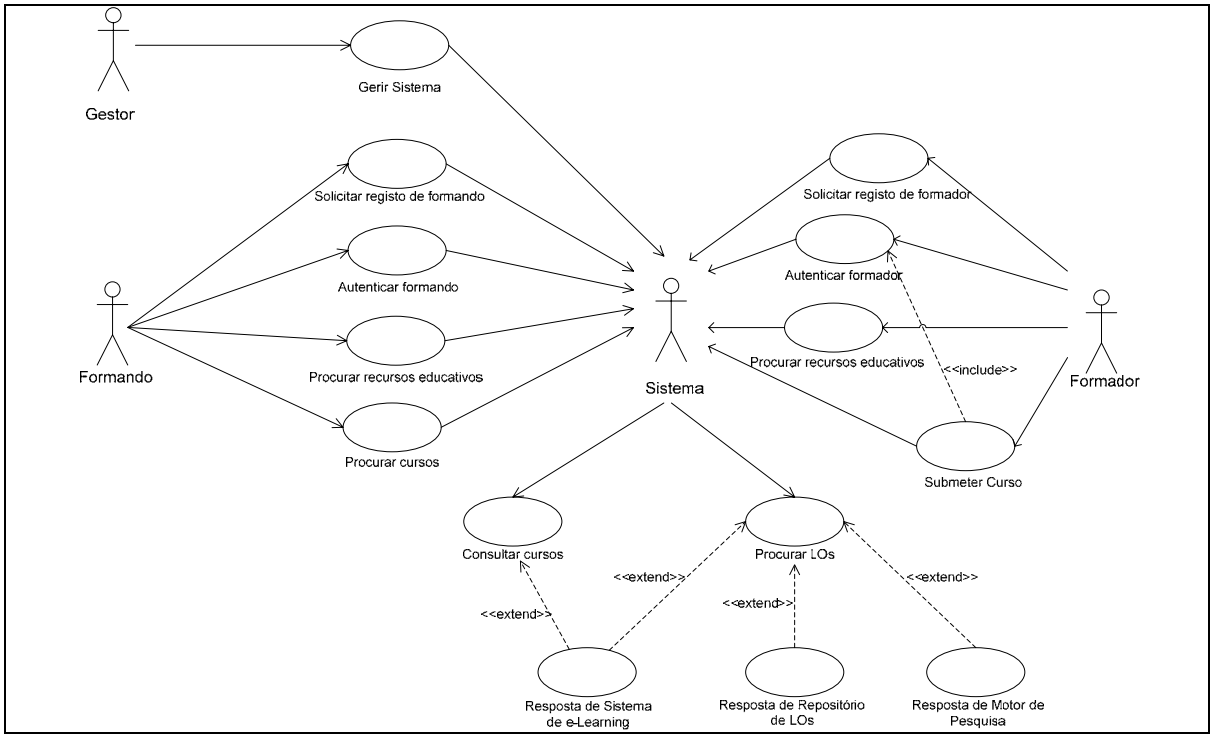


Figura 74 – Modelação do contexto do sistema proposto

Um requisito corresponde a uma funcionalidade ou condição que o sistema deverá satisfazer. Tipicamente, os requisitos podem dividir-se em funcionais e não-funcionais. Os requisitos funcionais descrevem a funcionalidade desejada para o sistema, uma vez que permitem descrever as funções que os utilizadores esperam, necessitam ou requerem do sistema. Os requisitos não-funcionais correspondem a características globais, tais como manutenção, gestão da base de dados, usabilidade, desempenho, fiabilidade e segurança.

Com base na modelação do contexto e dos requisitos (ver Figura 74 e Figura 75), foram detectados cinco tipos de actores:

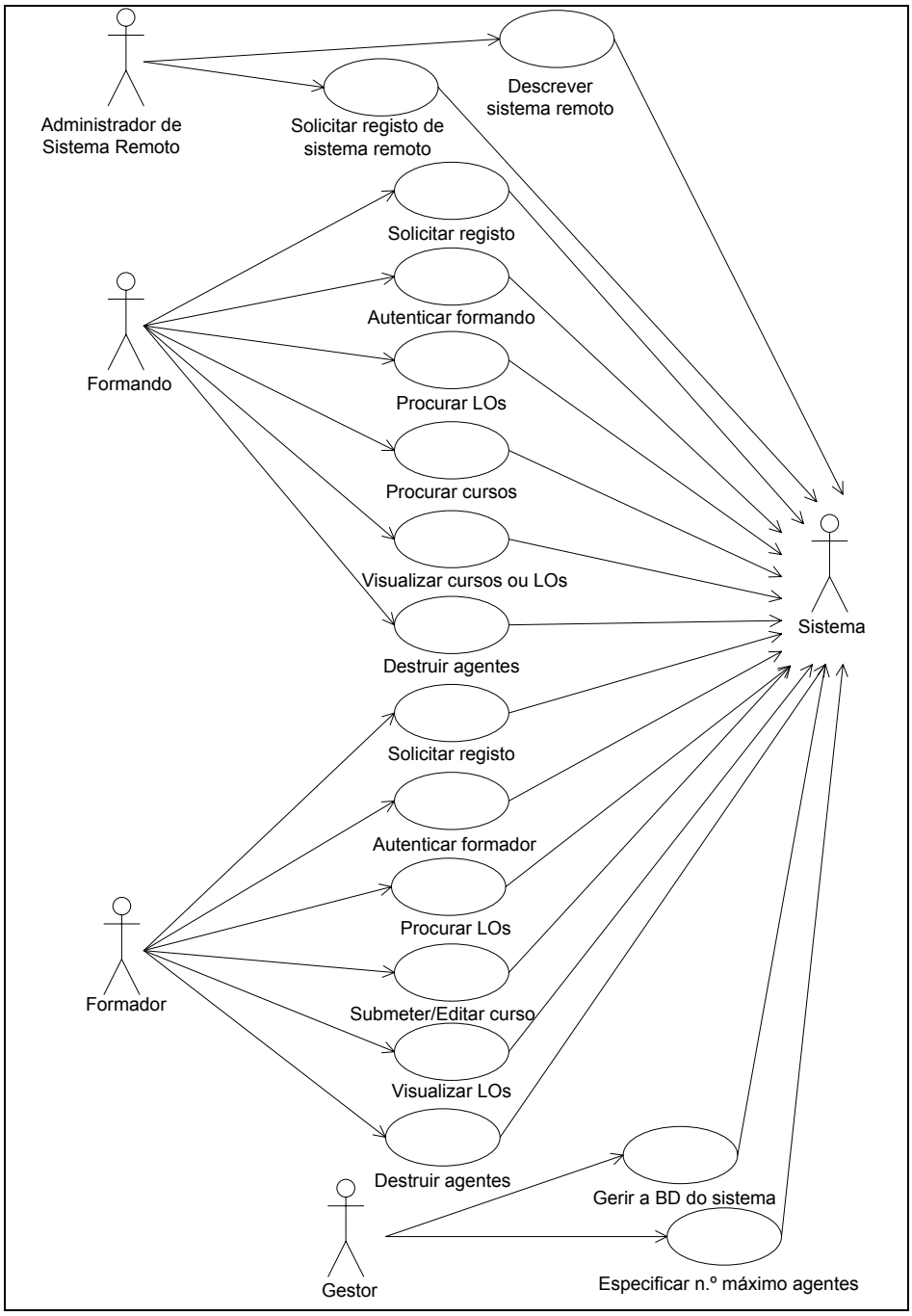


Figura 75 – Diagrama UML de casos de utilização

- **Formando**: aluno ou potencial aluno que irá usar o sistema para procurar e visualizar objectos de aprendizagem ou e-cursos;
- **Formador**: professor que submete e edita planos de e-cursos, associando recursos aos conteúdos apresentados nos planos (para tal poderá ter necessidade de procurar recursos);

- **Administrador de sistema remoto**: pessoa responsável pela gestão do servidor de um sistema externo de e-Learning, repositório de objectos de aprendizagem, sistema de gestão de conteúdos ou outro sistema Web que disponibilize recursos educativos;
- **Gestor (Administrador)**: pessoa responsável pela gestão do sistema proposto. O gestor realiza tarefas de administração da base de dados do sistema, tais como: adicionar/eliminar utilizadores, editar mapas de tópicos e ontologias e registar sistemas remotos;
- **Sistema**: ambiente que disponibiliza serviços que permitem a gestão do sistema e a pesquisa e recuperação de objectos de aprendizagem armazenados em sistemas remotos e externos.

Os casos de utilização "procurar LOs" e "procurar cursos" prevêem duas formas complementares de pesquisa semântica:

1) Pesquisa semântica básica composta por um campo para introduzir palavras-chave simples e algumas caixas de selecção de parâmetros de pesquisa básicos;
2) Pesquisa semântica avançada composta por vários campos para permitir localizar conceitos semanticamente através de elementos dos esquemas de metadados e de algumas relações ontológicas aplicáveis.

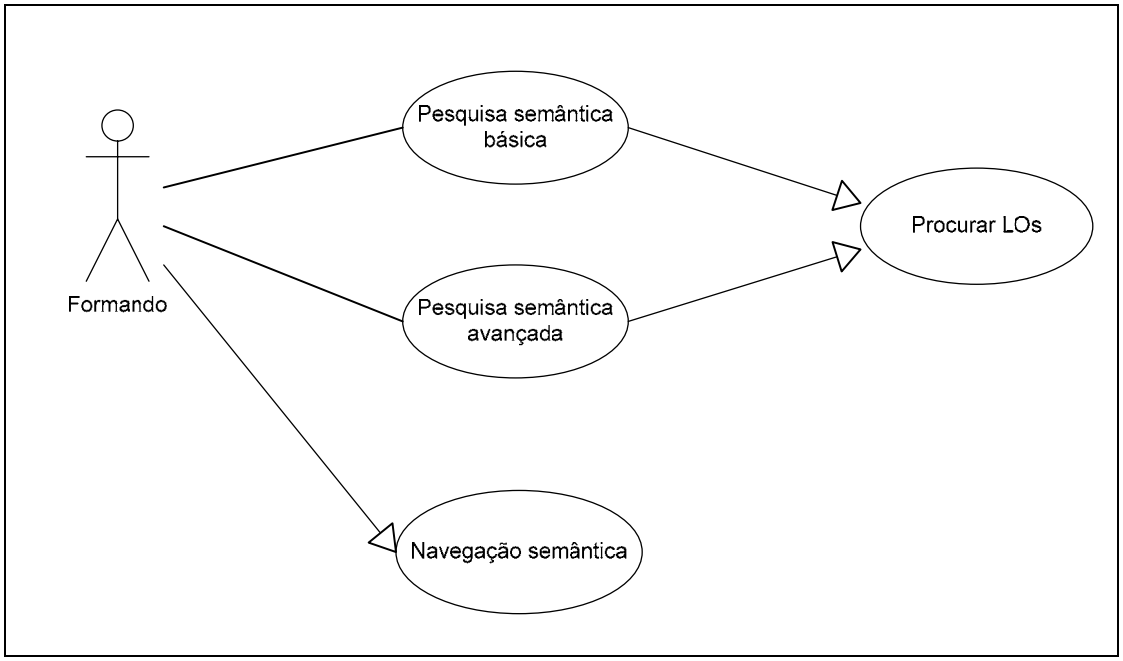


Figura 76 – Casos de utilização: procurar LOs

Para possibilitar a navegação semântica, torna-se necessário disponibilizar um meio de os formadores poderem submeter disciplinas ou cursos de formação sob a forma de mapas de tópicos. Tal como se pode observar na Figura 77, caso o formador não possua um mapa de tópicos da disciplina em formato XTM para submeter (*upload*), é conveniente que o sistema que aqui se propõe forneça as interfaces necessárias para a criação e edição dos correspondentes mapas de tópicos e para a associação dos LOs aos tópicos (note-se que esta associação corresponde a associar um tópico ao URI do recurso). Além desta funcionalidade, tanto os formadores como os administradores de sistemas remotos também podem descrever esses sistemas e os respectivos cursos, criando assim uma camada semântica associada aos sistemas registados no sistema WSE que orientará os agentes na pesquisa de LOs.

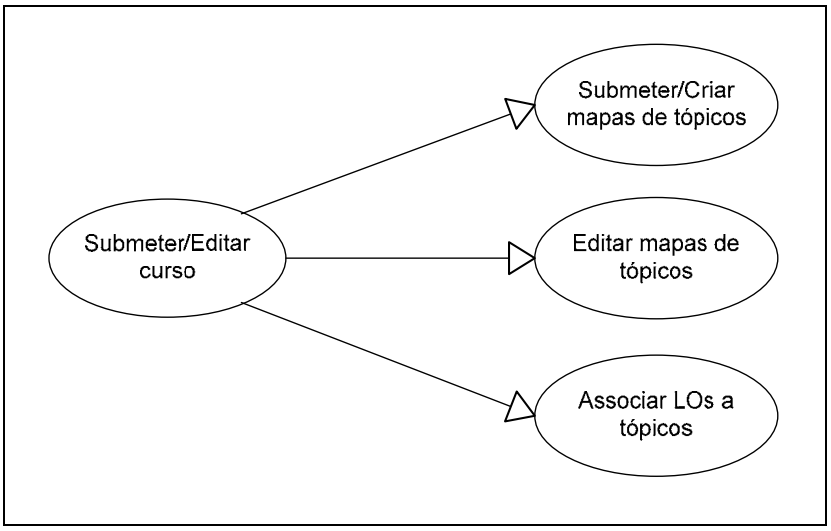


Figura 77 – Casos de utilização: submeter/editar curso

Assim, o formando poderia usufruir da navegação semântica proporcionada pelo mapa de tópicos correspondente ao curso que procurava. Ou seja, o resultado da procura (listagem de cursos, links e respectiva descrição) poderá incluir a navegação semântica em cursos através dos tópicos e suas associações e ocorrências (LOs) proporcionada pelo mapa de tópicos existente no sistema WSE e não nos sistemas remotos nos quais esses objectos de aprendizagem se encontram armazenados.

Por conseguinte, no âmbito da visualização dos cursos ou LOs, é desejável dispor de duas formas alternativas de visualização da informação:

a) A navegação através da típica listagem de cursos ou de LOs e respectivos links;

b) A navegação semântica através de mapas de tópicos.

Mas, no caso da pesquisa de LOs seria também desejável a associação automática de LOs aos tópicos dos mapas de tópicos.

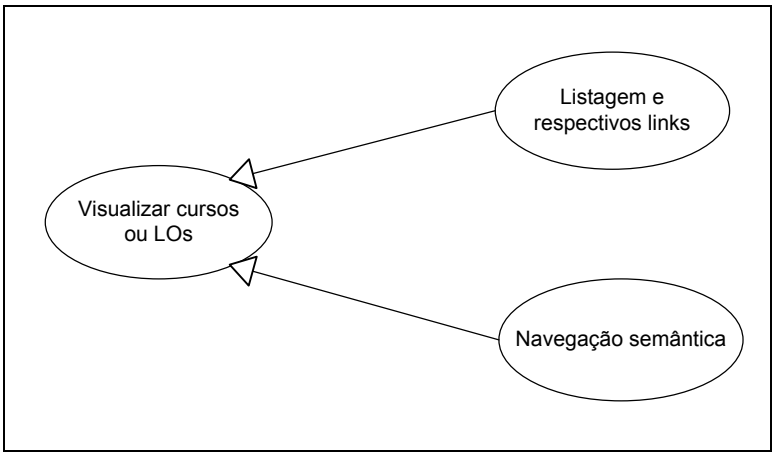


Figura 78 – Casos de utilização: visualizar cursos ou LOs

A complexidade dos mecanismos de inferência semântica (campo ao qual só agora se começa a atribuir verdadeiro interesse de investigação, tendo em vista a recomendação ou normalização de tecnologias para a inferência) poderá limitar o desenvolvimento deste requisito.

Na modelação do contexto e dos requisitos aqui exposta usou-se uma abordagem genérica, mostrando alguns dos modelos que suportaram a comunicação com os utilizadores. Isoladamente, os modelos aqui apresentados não permitem obter uma visão completa do sistema proposto e dos sistemas remotos com os quais se relaciona (sistemas de e-Learning, repositórios de LOs, servidores Web de conteúdos); no entanto, parece-nos serem suficientes para expressar a visão global que nos permitiu projectar a arquitectura do sistema e demonstrar a sua utilidade nas actividades de modelação e especificação dos requisitos.

6.2.2- Arquitectura de Alto Nível para o Sistema WSE

A especificação de requisitos através do diagrama de casos de utilização da Figura 75 permitiu perceber as principais funcionalidades e cenários de aplicação para projectar uma arquitectura de alto nível para o processo de pesquisa de recursos educativos, ou seja, permitiu descrever o que se espera que o processo de pesquisa do sistema faça, sem preocupações de como o fará. Posteriormente e na perspectiva do analista, houve a preocupação de ter em

atenção as tecnologias emergentes, tanto ao nível das tecnologias de estruturação sintáctica e semântica da informação (tecnologias para a Web Semântica), como ao nível da reutilização e interoperabilidade de objectos de aprendizagem (tecnologias para o e-Learning) e do aperfeiçoamento dos mecanismos de pesquisa (tecnologias para os agentes de software).

Tomando como base a arquitectura proposta por Sycara (1999), podemos destacar três tipos de agentes: **agentes de interface** (que recebem, modelam e utilizam as solicitações dos utilizadores, conduzindo a coordenação do sistema para a execução de tarefas); **agentes de tarefa** (que suportam a tomada de decisão e executam tarefas inerentes à formulação da pesquisa submetida, realizando consultas e trocas de informação com outros agentes) e **agentes de informação** (que processam inteligente e eficientemente a recuperação da informação). Esta solução favorecerá o acesso, a localização e a reutilização dos conteúdos educacionais através da implementação de um mecanismo de busca baseado em metadados e ontologias para localizar e recuperar objectos de aprendizagem armazenados em sistemas Web espalhados pela Internet.

Por conseguinte, propomos um modelo baseado num sistema multiagente de agentes inteligentes e móveis e nas tecnologias para a Web Semântica através do qual, recorrendo a anotações semânticas, será possível agentes localizarem e recuperarem conteúdos educativos a partir de servidores remotos (ver Figura 79).

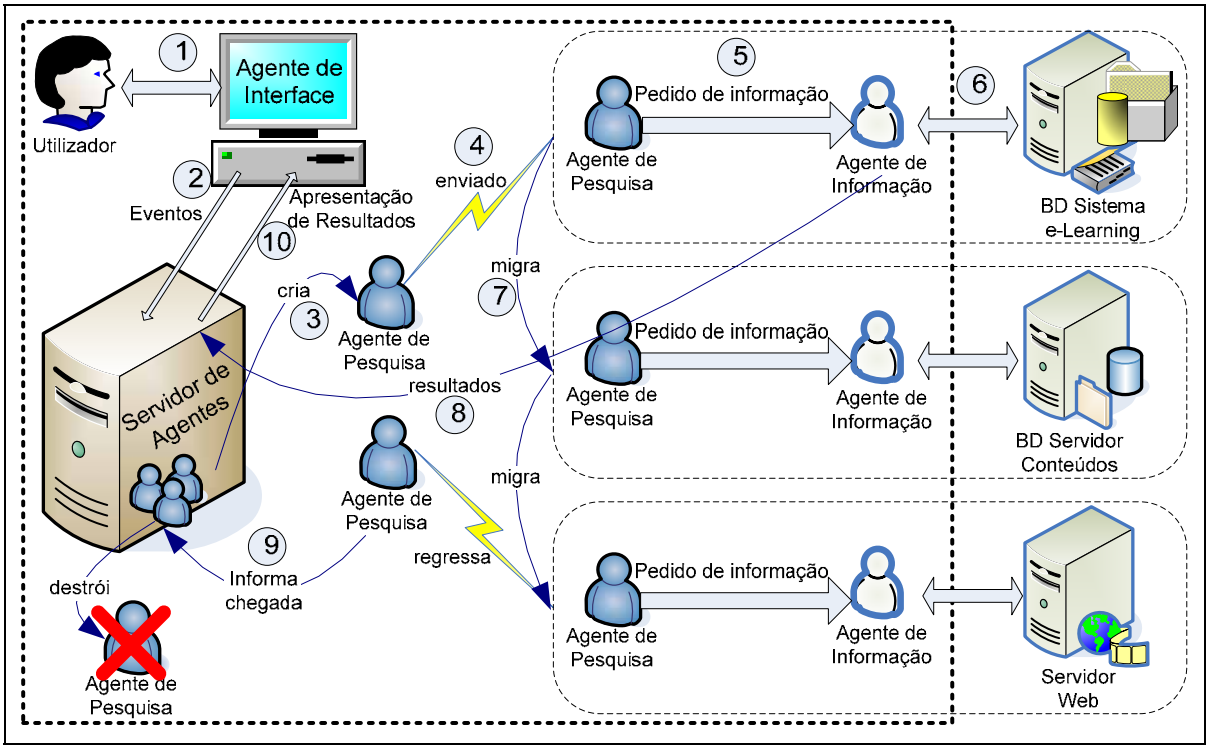


Figura 79 – Arquitectura genérica do Sistema WSE

Considerando um nível de abstracção alto, podemos resumir os passos para a realização da pesquisa de objectos de aprendizagem da seguinte forma:

1) Formador e formando interagem com o sistema através de uma interface gráfica denominada agente de interface, a partir da qual submetem uma pesquisa sobre conteúdo educativo que desejam localizar;
2) O agente de interface informa o servidor de agentes da necessidade de realizar uma tarefa: procurar informação sobre um dado tema (ou mais concretamente, procurar objectos de aprendizagem);
3) De acordo com a tarefa de pesquisa requisitada, o servidor de agentes cria e envia para a rede um agente orientado à realização dessa tarefa (agente de pesquisa);
4) O agente de pesquisa migra para o servidor de um sistema de e-Learning, de um repositório de objectos de aprendizagem ou de outro servidor Web de conteúdos educativos registado no sistema que recorra ou não a bases de dados para armazenar conteúdos educativos;
5) Não sendo especialista na comunicação com as bases de dados ou na consulta de documentos semânticos (RDF/XML, incluindo documentos RDFS e/ou OWL), o agente de pesquisa móvel requisita essa funcionalidade ao agente de informação;
6) O agente de informação efectua a pesquisa a fim de localizar e recuperar os objectos de aprendizagem solicitados pelo agente de pesquisa, recorrendo aos metadados DCM e/ou LOM (armazenados na BD e expressos em RDF/XML) e às ontologias disponíveis de acordo com as regras de inferência previamente estabelecidas;
7) Enquanto o agente de informação realiza a sua tarefa, o agente de pesquisa segue o seu itinerário, migrando para outro servidor Web;
8) Caso existam objectos de aprendizagem, o agente de informação devolve directamente o resultado ao servidor de agentes (lista de recursos e respectivos links);
9) Após ter cumprido o itinerário, o agente de pesquisa informa o servidor de agentes do seu regresso, para que este proceda à sua destruição;
10) A lista de recursos e respectivos links é visualizada pelo utilizador. Seria desejável que os LOs que se enquadram em pelo menos um dos mapas de tópicos existentes no sistema (planos dos e-cursos) fossem incluídos como links dos conceitos ou tópicos descritos em XTM, agrupando-os num curso personalizado navegável semanticamente. Contudo, a

integração automática dos LOs nos mapas de tópicos pode traduzir-se em cursos de formação pedagogicamente incoerentes. Este passo poderá constituir por si só um desafio para outros trabalhos de investigação.

Em suma, por intermédio do servidor de agentes, os agentes de pesquisa móveis recebem as tarefas dos agentes de interface que, por sua vez, requisitam dados aos agentes de informação. Os agentes de informação disponibilizam os resultados das pesquisas ao servidor de agentes móveis, que os vão disponibilizando ao utilizador através dos agentes de interface. Por conseguinte, o servidor de agentes é o núcleo de todo o sistema, uma vez que é o responsável pelo envio de agentes móveis para a rede e pela coordenação e gestão de todo o processo. Mas para que seja possível recuperar recursos educativos, a informação desses servidores deverá estar organizada na perspectiva de objectos de aprendizagem (IMS/SCORM). Para facilitar a pesquisa desses objectos de aprendizagem, os mesmos devem previamente ser descritos através de metadados (RDF/LOM/DCM) e o seu significado representado formalmente em ontologias.

Face à dimensão do sistema proposto e de acordo com a Figura 72 – Modelo genérico da actividade de análise de requisitos, definimos três níveis de prioridade (reduzida, média e elevada). Os requisitos inerentes à adição (semi)automática de metadados aos objectos de aprendizagem de um sistema de e-Learning remoto e à pesquisa e recuperação dos mesmos foram considerados requisitos de elevada prioridade. Os requisitos referentes à anotação semântica dos sistemas remotos, à pesquisa e recuperação de objectos de aprendizagem inerente ao núcleo do sistema ou servidor de agentes WSE e aos mecanismos de pesquisa semântica dos agentes de informação foram avaliados com prioridade média. Finalmente, os requisitos respeitantes à organização e navegação dos objectos de aprendizagem em mapas de tópicos (ou ontologias) formando cursos personalizados, à definição de regras de inferência, à edição *on-line* de ontologias e mapas de tópicos, à exploração da expressividade das ontologias e à optimização do motor de inferência semântica foram classificados com menos prioridade.

6.2.3- Modelação dos Requisitos de um Sistema de e-Learning

Os sistemas de e-Learning, repositórios de objectos de aprendizagem ou outros servidores Web de conteúdos educativos contituem os sistemas remotos do Sistema WSE. Para que os agentes de informação tirem proveito das potencialidades da WS, é indispensável que os recursos de aprendizagem armazenados nesses sistemas remotos estejam devidamente descritos com metadados e o seu domínio modelado através de uma ontologia. Portanto, a garantia de que o agente de informação é capaz de localizar e recuperar LOs através dos seus metadados contitui o principal requisito para que esses sistemas possam solicitar o registo no sistema WSE.

Sendo os sistemas de e-Learning os sistemas remotos que mais proveito poderão tirar do registo no sistema WSE, a presente secção tem como principal objectivo apresentar a modelação dos requisitos de um sistema de e-Learning.

Tal como referimos, os cursos adequados ao processo de Bolonha originaram a identificação de novos requisitos inerentes à necessidade de descobrir conhecimento indispensável para a tomada de decisões. Referíamo-nos mais concretamente ao seguinte:

- Os cursos adequados ao processo de Bolonha permitem que os alunos possam escolher disciplinas de outros cursos para complementarem o seu plano de estudo. Para tal, necessitam de informação actual sobre os programas e conteúdos leccionados nas diferentes disciplinas para poderem tomar decisões;
- Reconhece-se desde há muito tempo que a prática da interdisciplinaridade é relevante para o processo de aprendizagem e de formação do aluno. Mas, o diálogo entre professores nem sempre é possível e muitas vezes a interdisciplinaridade fica apenas manifestada nos planos, tornando-se difícil que cada professor possa definir, individualmente, estratégias para o alinhamento e integração de temas ou conteúdos transversais por falta de acesso a informação sobre o processo de aprendizagem das restantes disciplinas do mesmo curso.

Essa informação pode residir numa plataforma de e-Learning. Contudo, não encontrámos nenhuma que oferecesse um mecanismo de pesquisa que permitisse localizar e recuperar informação sobre os LOs (i.e. metadados), garantindo que os LOs propriamente ditos não fossem visualizados por alunos ou professores não autorizados. Ou seja, para além de auxiliarem os agentes de software na localização e recuperação de LOs, os metadados

podem também fornecer informação útil aos alunos e professores que dela necessitem para tomar decisões inerentes aos dois tipos de requisitos referidos, sem que haja necessidade de visualizar o LO correspondente. Portanto, um dos requisitos a incluir na especificação é o acesso aos metadados dos cursos e aos metadados sobre os recursos educativos desses cursos por parte de todos os utilizadores registados na plataforma, independentemente de terem acesso autorizado ao curso ou à disciplina e aos respectivos objectos de aprendizagem.

Para realizar esse tipo pesquisas, os planos das disciplinas e os LOs devem previamente ser descritos através de metadados. No entanto, nem todas as plataformas permitem a anotação dos LOs criados *on-line* na própria plataforma. A possibilidade de importação de pacotes SCORM, IMS-CP e IMS-LD inclui LOs descritos com metadados. Contudo, a observação da prática docente mostrou-nos que muito raramente os professores o faziam.

Por conseguinte, um novo requisito foi identificado: a anotação semântica dos objectos de aprendizagem criados *on-line* num sistema de e-Learning. O recurso a metadados e ontologias permitirá a concretização dessa anotação semântica.

Com vista a aliviar o professor desse processo entediante e evitar eventuais erros de anotação por falta de familiarização com as especificações, a descrição dos LOs deveria ser automatizada tanto quanto possível.

Na mesma linha de pensamento, todos os outros tipos de sistemas remotos e externos ao sistema que aqui se propõe deverão garantir este requisito para que possam solicitar o seu registo.

A identificação destes novos requisitos foi a principal razão que levou à definição das prioridades que se apresentaram anteriormente. Assim, interessa agora apresentar sucintamente a modelação dos requisitos inerente aos sistemas de e-Learning.

Na perspectiva do formador haverá necessidade de anotar os LOs existentes numa disciplina ou curso de formação. Tal como referimos, a anotação manual dos LOs é um processo que exige tempo e familiaridade com as especificações de metadados. A anotação (semi)automática dos LOs em geral e dos programas e sumários de uma disciplina apresentou-se como o requisito indispensável para garantir que todas as disciplinas ou cursos de formação fossem mínima e automaticamente anotados com metadados. Tal como se pode observar na Figura 80, o caso de utilização “Criar Programas e Sumários” incorpora explicitamente o caso de utilização “Anotar LOs existentes”.

A consulta de informação sobre disciplinas ou cursos de formação corresponde à visualização dos metadados dos sumários e programas das disciplinas decorrentes do caso de utilização "anotar programas e sumários" do formador. Tratando-se de informação de acesso público, a sua visualização não exige acesso como formador ou como formando. O mesmo poderá não suceder com os metadados dos restantes objectos de aprendizagem.

Normalmente, qualquer visitante poderá solicitar a consulta de informações sobre os recursos de uma disciplina (metadados dos objectos de aprendizagem), a menos que o formador decida o contrário. No entanto, para aceder a uma disciplina ou curso de formação e/ou visualizar os objectos de aprendizagem correspondentes terá que obrigatoriamente entrar no sistema como utilizador registado (formador ou formando).

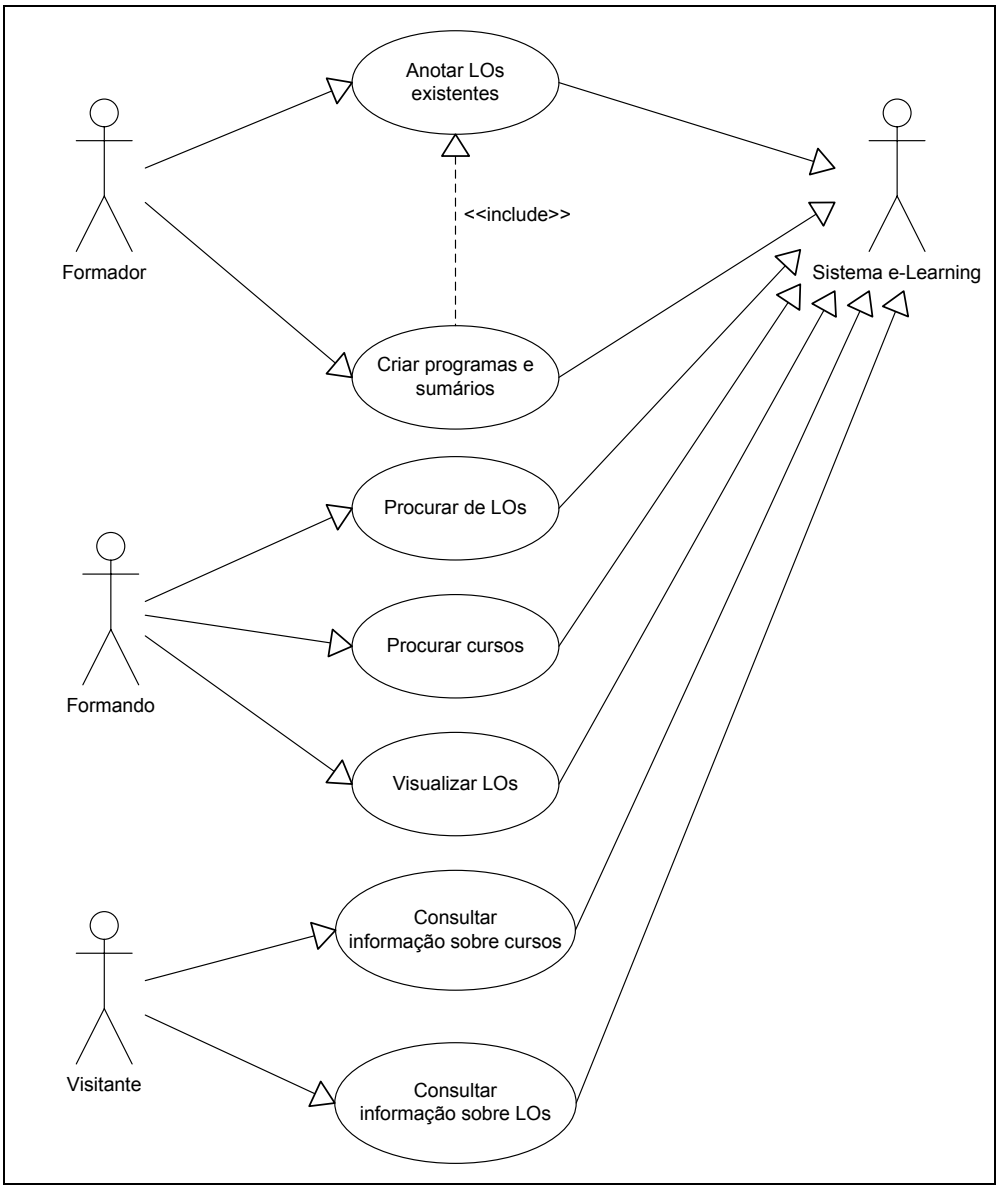


Figura 80 – Casos de utilização: sistema de e-Learning

Quanto à procura de LOs por parte do formando, é conveniente haver duas formas de pesquisar semanticamente as disciplinas ou cursos de formação:

1) Pesquisa semântica básica composta apenas por um campo para introduzir palavras-chave simples;

2) Pesquisa semântica avançada composta por vários campos que permitem localizar semanticamente objectos de aprendizagem de acordo com o seu autor, o professor da disciplina ou o criador de metadados e com o assunto ou escolhendo qualquer um dos elementos de uma das normas utilizadas na anotação, incidindo não só sobre os esquemas de metadados, mas também sobre as relações ontológicas aplicáveis.

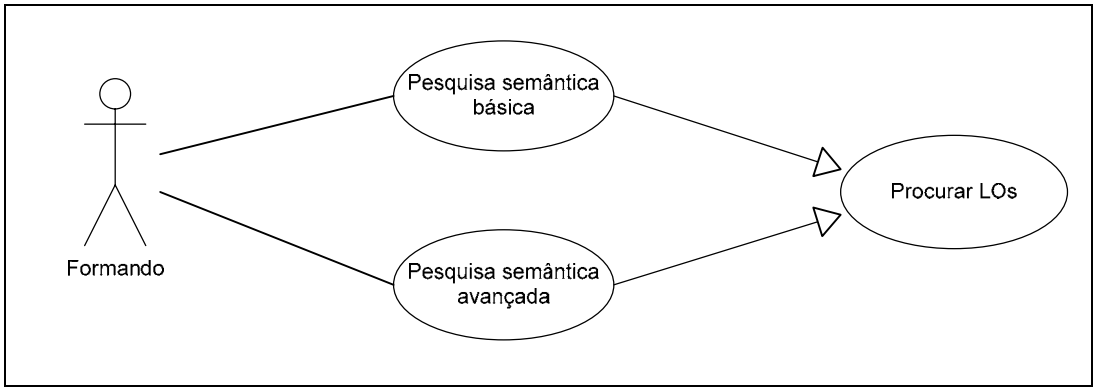


Figura 81 – Casos de utilização: procurar LOs num sistema de e-Learning

Intencionalmente, os modelos apresentados fornecem uma visão de elevada abstracção, evitando expor aqui uma modelação detalhada com vista a focar a atenção nos aspectos mais relevantes para especificar a arquitectura desta parte de um sistema de e-Learning.

6.2.4- Arquitectura de Alto Nível para um Sistema de e-Learning

Face aos requisitos identificados, o modelo genérico de anotação e pesquisa semântica de LOs para um sistema de e-Learning inclui:

- Dois módulos que correspondem a duas actividades distintas para a anotação de LOs;
- Um bloco de pesquisa baseado nos metadados armazenados na base de dados do sistema de e-Learning;
- Um motor de inferência baseado nos metadados e nas ontologias.

A Figura 82 ilustra esses componentes e a sua integração e principais funções no âmbito dessa arquitectura de alto nível para um sistema de e-Learning.

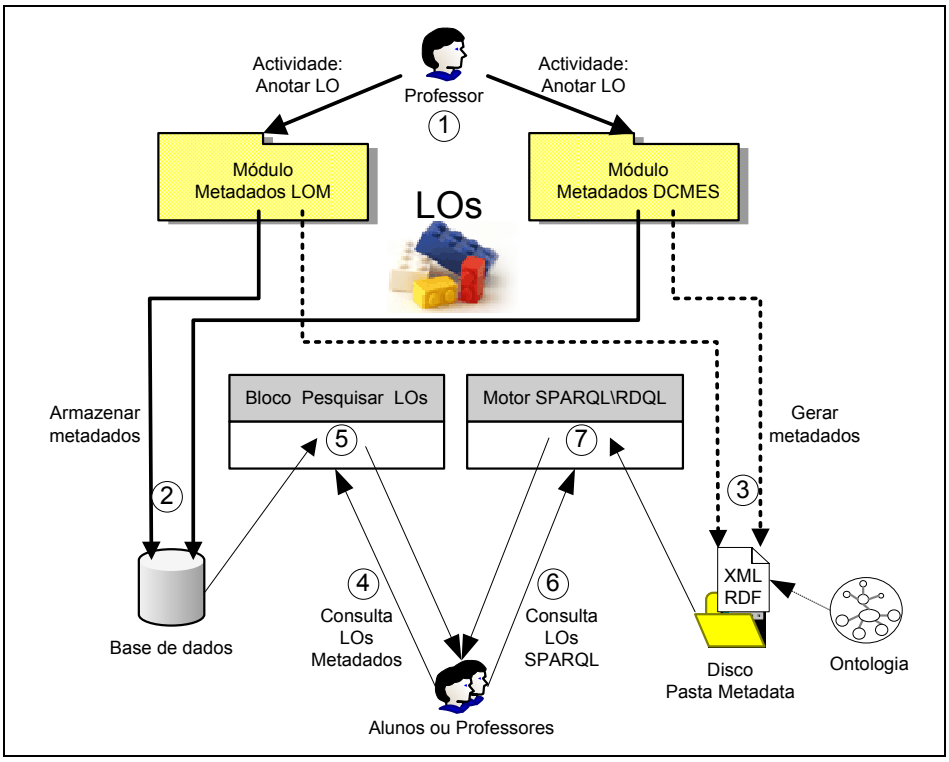


Figura 82 – Arquitectura genérica dos módulos e blocos para um sistema de e-Learning

As principais funções destes componentes podem ser resumidas nos seguintes passos:

1) No contexto de uma disciplina de um sistema de e-Learning, os professores podem descrever os LOs da mesma através de metadados;

2) Os metadados devem ser gerados automaticamente e armazenados na BD do sistema de e-Learning aquando da criação do LO ou, caso este já exista ou tenha sido importado, aquando da indicação do professor. Não obstante, o professor pode completar ou alterar os metadados sempre que assim o entender;

3) Os metadados devem também ser armazenados no disco em formato XML e RDF/XML;

4) Assim, o processo de pesquisa de LOs poderá ser efectuado através dos registos de metadados existentes na BD do sistema de e-Learning ou através dos ficheiros RDF/XML gerados automaticamente;

5) A pesquisa realizada na BD deve permitir a consulta de todos os LOs pelo nome do criador, professor, palavras-chave ou assunto e escolhendo um dos elementos de uma das normas (DCM ou LOM). É conveniente também que o utilizador possa escolher o âmbito

da pesquisa: em todos os cursos, apenas nos cursos do utilizador ou num determinado curso específico. Caso o utilizador não tenha autorização para visualizar os LOs devolvidos, deverá poder, no entanto, ver os metadados de acordo com as definições atribuídas pelo professor/criador dos metadados;

6) Uma vez que a disciplina pode conter pacotes SCORM/IMS recheados de metadados, para além dos metadados gerados pelos módulos, a pesquisa semântica deverá ser suportada por um motor SPARQL/RDQL para possibilitar realizar consultas baseadas na ontologia do sistema e nos ficheiros RDF/XML;

7) O resultado devolvido em XML deve ser formatado recorrendo a folhas de estilo em linguagem XSL (*eXtensible Style Language*).

Em suma, o bloco de pesquisa deve disponibilizar duas modalidades: uma das pesquisas deve permitir localizar os LOs através dos elementos de metadados armazenados na base de dados do sistema de e-Learning; a outra deve possibilitar a localização desses objectos directamente nos metadados armazenados em ficheiros XML ou na descrição RDF/XML. Esta última opção permite navegar nas relações existentes entre os LOs. Obviamente, a navegação semântica dependerá do nível de expressividade semântica desses metadados.

6.3- Modelação do Comportamento

A linguagem UML permite especificar a dinâmica ou o comportamento de um sistema através de Diagramas de Estado, Diagramas de Actividades e Diagramas de Interacção entre objectos (Diagramas de Sequência e Diagramas de Colaboração), para além dos Diagramas de Casos de Utilização, uma vez que representam uma visão do comportamento do sistema na perspectiva do seu utilizador.

Os Diagramas de Casos de Utilização organizam os comportamentos do sistema. Os Diagramas de Estados focam as mudanças de estados do sistema motivadas por eventos, ou seja, mostram o fluxo de controlo de um estado para outro. Os Diagramas de Actividades mostram o fluxo de controlo de uma actividade para outra. Os Diagramas de Sequência têm

como foco a ordem temporal das mensagens. E os Diagramas de Colaboração têm como objectivo a organização estrutural dos objectos que enviam e recebem mensagens (Booch et al., 1999).

Com base na modelação e especificação dos requisitos do sistema foi possível identificar três tipos de agentes: agente de interface para interagir directamente com o utilizador e registar os pedidos de consulta de LOs ou cursos, agentes móveis de pesquisa para transportar os pedidos até aos sistemas remotos e os agentes de informação educativa para realizar a inferência de acordo com as especificidades de cada um dos sistemas remotos.

Na Figura 83 é apresentado o diagrama de estados do agente de interface na perspectiva de um formando. Na perspectiva do formador acrescem as funcionalidades para complementar as anotações semânticas, submeter e criar/editar ontologias e mapas de tópicos.

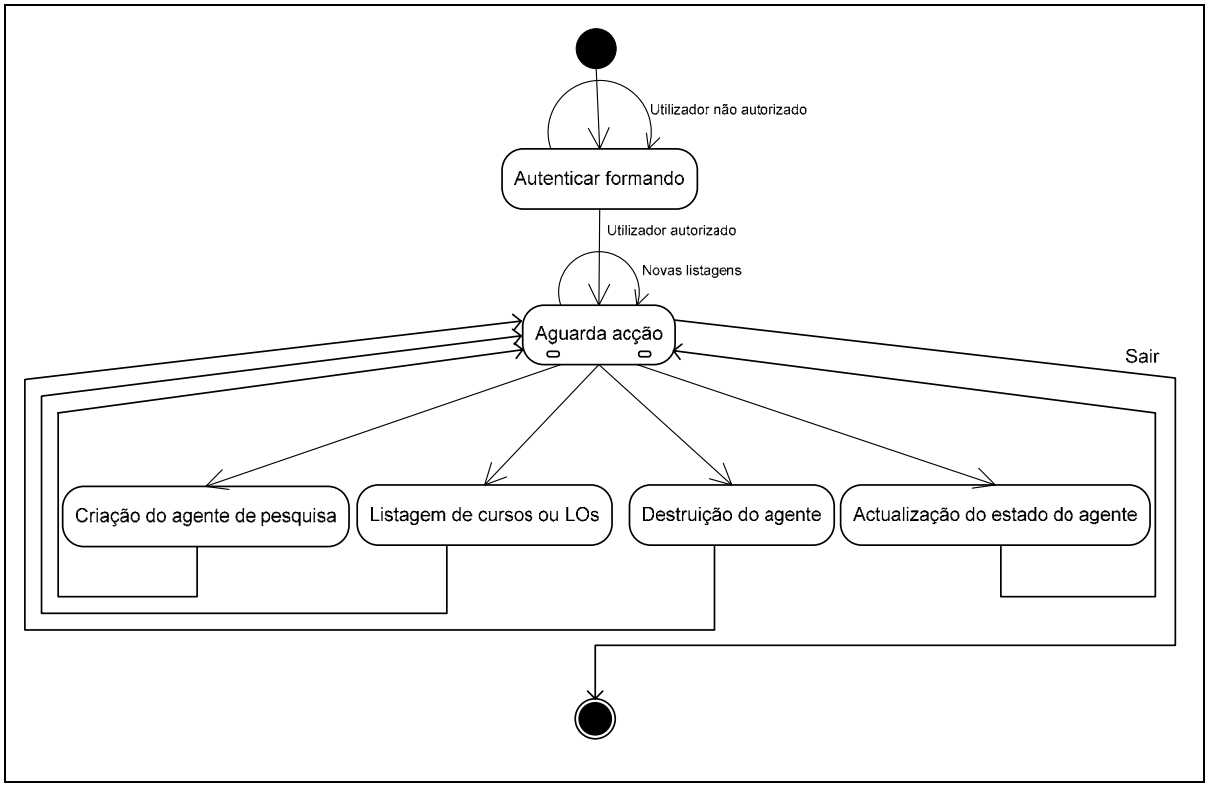


Figura 83 – Diagrama de estados do agente de interface

Primeiramente é verificado se o utilizador está autorizado a utilizar as funcionalidades do sistema. Caso existam listagens de cursos ou de LOs ainda não visualizadas inerentes ao regresso de um agente de pesquisa, o agente de interface informa o utilizador. Esta funcionalidade é oferecida pelo agente de interface, uma vez que não é necessário que o utilizador esteja ligado ao sistema para que os seus agentes móveis continuem à procura de

LOs ou cursos. Caso contrário, o agente de interface aguarda a escolha da funcionalidade pretendida pelo utilizador. Face à escolha do utilizador, o agente de interface satisfaz o seu pedido e regressa ao estado de espera.

Na Figura 84 é apresentado o diagrama de actividade referente ao servidor de agentes móveis. As primeiras actividades são a criação dos objectos necessários para disponibilizar as diferentes funcionalidades aos diversos componentes do sistema (incluindo formandos e formadores) e ao administrador do sistema.

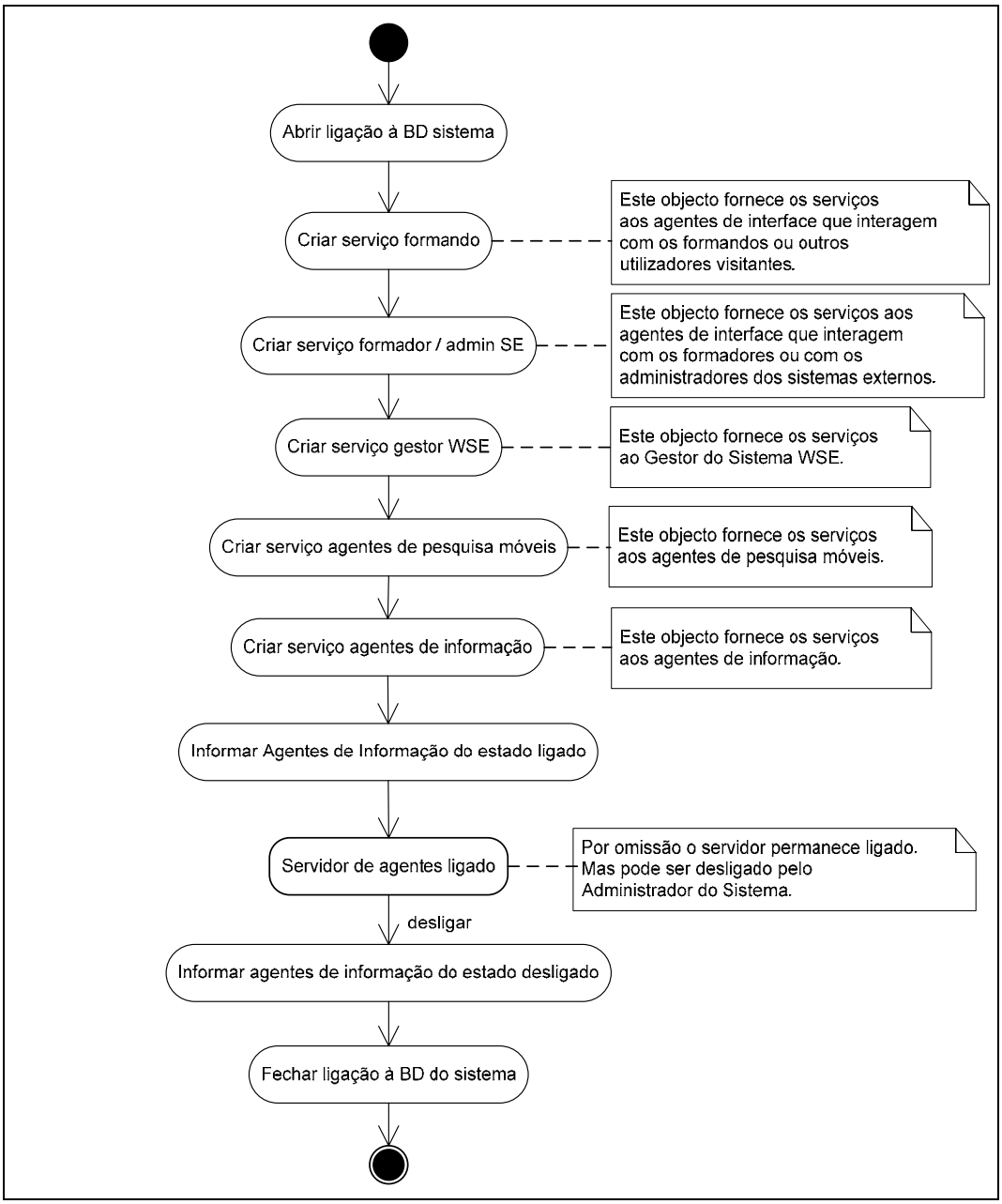


Figura 84 – Diagrama de actividade do servidor de agentes móveis

Na Figura 85 podemos observar o diagrama de actividade do agente de informação responsável pelo acesso à base de dados do sistema remoto e à pasta onde são guardados os LOs e pela criação dos objectos que disponibilizam as diversas funcionalidades aos agentes de informação e ao servidor de agentes. Após a criação destes serviços, o agente de informação informa o servidor de agentes que está ligado e disponível para receber agentes móveis.

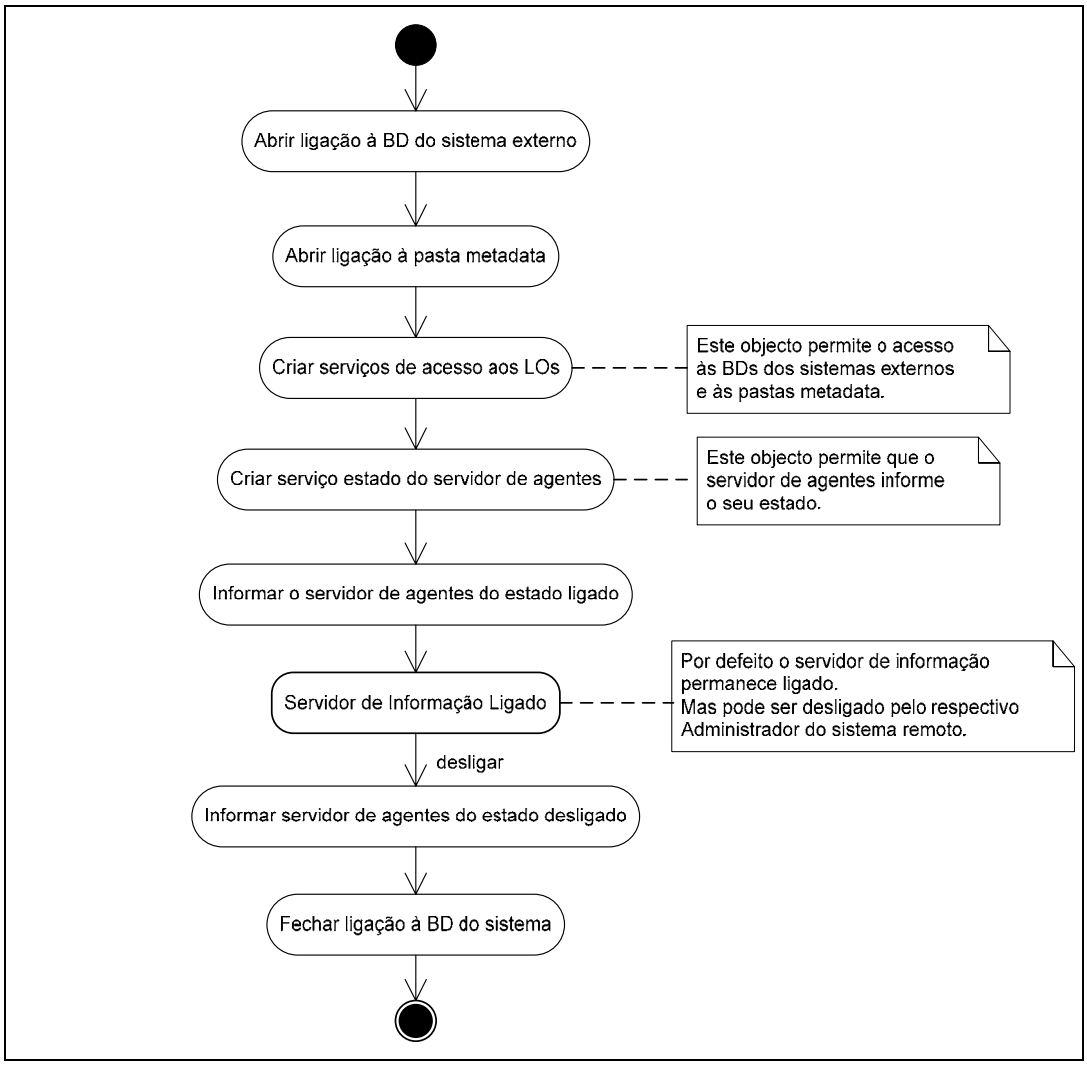


Figura 85 – Diagrama de actividade do agente de informação

Os diagramas apresentados permitem-nos mostrar as principais mudanças de estado e de actividade do sistema e alguns dos fluxos de controlo de um estado para outro ou de uma actividade para outra. Facilmente se percebe que o modelo proposto descreve um espaço onde agentes estacionários interagem com agentes móveis trocando informação com vista à localização e recuperação de cursos ou objectos de aprendizagem de acordo com as condições do utilizador. Estes modelos serão explanados ao longo deste capítulo.

6.4- Modelação da Estrutura

A modelação da estrutura de um sistema consiste essencialmente na identificação de objectos (instâncias), de classes (conjuntos de objectos que partilham os mesmos atributos, operações, relações e semântica), de interfaces (classes de objectos que permitem a comunicação entre o mundo exterior e o sistema), de colaborações (sociedades de classes, interfaces e outros elementos que proporcionam algum comportamento cooperativo) e das suas relações, entre outros aspectos estáticos do sistema. A UML disponibiliza dois tipos de diagramas complementares para visualizar, especificar, construir e documentar a estrutura de um sistema: Diagrama de Classes e Diagrama de Objectos.

Com base na especificação de requisitos, os diagramas de classes e de objectos permitem-nos obter uma visão do sistema na perspectiva de classes, de objectos, de interfaces e das suas relações que representamos de forma geral através da Figura 86.

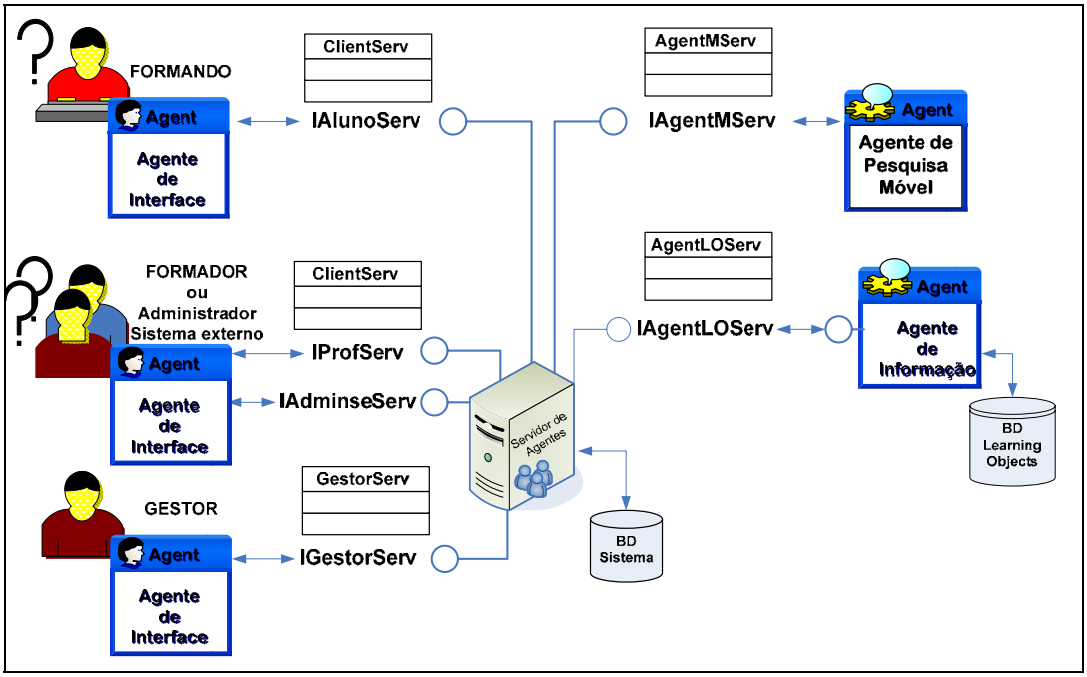


Figura 86 – Modelação da estrutura do servidor de agentes

No âmbito do servidor de agentes móveis, os objectos “ClientServ”, “GestorServ”, “AgentMServ” e “AgentLOServ” correspondem às classes e os objectos “IAlunoServ”, “IProfServ”, “IGestorServ”, “IAdminseServ”, “IAgentMServ” e “IAgentLOServ” às interfaces. As classes implementam as interfaces, ou seja, disponibilizam toda a funcionalidade para os métodos especificados nas interfaces. Este assunto será detalhado na secção 5.5.2, nomeadamente no âmbito da modelação lógica da BD e dos restantes componentes do sistema.

6.5- Modelação da Arquitectura

Uma arquitectura fornece uma imagem que sugere aquilo que se procura criar na realidade. Genericamente mostra como os componentes se enquadram conjuntamente para projectar essa nova realidade. Os diagramas de arquitectura permitem-nos descrever aspectos relevantes para a fase de construção e instalação do sistema, nomeadamente através de Diagramas de Componentes e Diagramas de Instalação, que dão uma visão da disposição dos componentes de software e hardware de um sistema.

Esta secção tem por objectivo propor uma arquitectura baseada num sistema multiagente de agentes inteligentes e móveis e na estrutura da Web Semântica, onde através de metadados e ontologias será possível localizar e recuperar conteúdos educacionais a partir de servidores que favoreçam a busca de informação por parte dos agentes, nomeadamente sistemas de e-Learning. Aplicar metadados e ontologias segundo a filosofia da Web Semântica aos sistemas de e-Learning favorecerá a personalização dos serviços de ensino/aprendizagem apontando para a satisfação das necessidades dos formandos e formadores enquanto utilizadores do sistema.

Não obstante, se assumirmos que a filosofia de um sistema de e-Learning se baseia em cursos, então será necessário aumentar o grau de granularidade de "curso" para fornecer a semântica no nível mais baixo possível. LOs de elevada granularidade, compostos por LOs o mais pequenos possível e anotados com metadados, são aqueles que oferecem uma maior possibilidade de reutilização. Assim, será possível optimizar a pesquisa de conteúdos educacionais usando a visão da Web Semântica, com vista a recuperá-los e reutilizá-los, independentemente da hierarquia existente entre eles.

Tal como podemos observar na Figura 87, a arquitectura do sistema WSE é composta por três níveis diferentes, mas complementares, cada um dos quais incluindo diferentes agentes consoante a funcionalidade exigida:

- **Nível cliente** – agente de interface para comunicação entre os diferentes perfis de utilizador e a aplicação do servidor de agentes;
- **Nível intermédio** – servidor de agentes que envia e gere os agentes de pesquisa móveis de acordo com os metadados e ontologias inerentes aos sistemas remotos;

- **Nível Fornecedor de LOs** – agentes de informação que acedem aos LOs e a outros objectos de informação, consultando as bases de dados dos sistemas ou inferindo os documentos RDF(S) e OWL de acordo com os pedidos dos agentes de pesquisa.

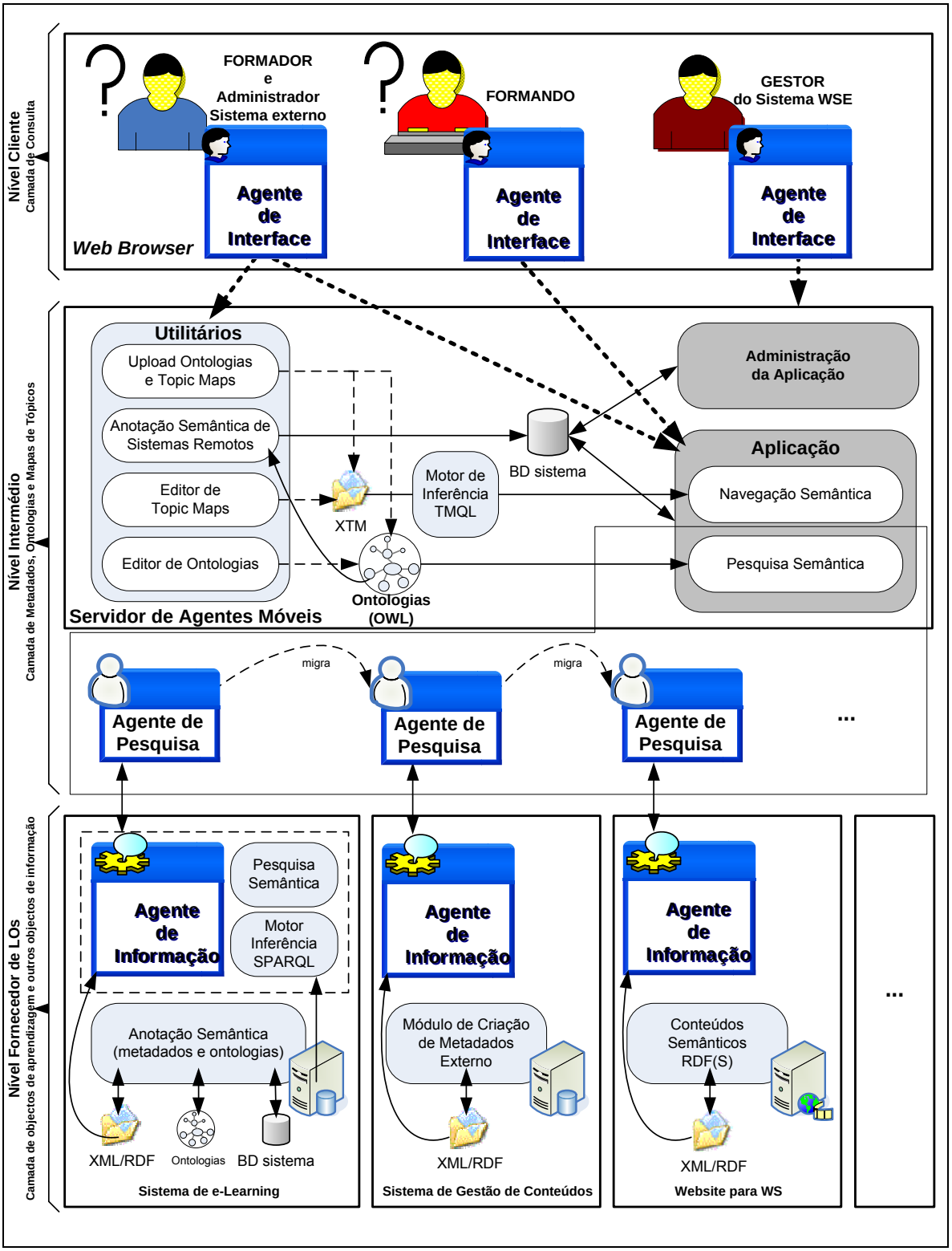


Figura 87 – Arquitectura do Sistema WSE e dos sistemas remotos

Nível cliente:

Genericamente, este nível disponibiliza o acesso ao serviço de agentes móveis e às respectivas funcionalidades de acordo com o perfil dos utilizadores. Através de uma interface gráfica denominada agente de interface:

- Os formandos podem indicar quais os objectos de aprendizagem que pretendem procurar disciplinas ou cursos de formação que pretendem frequentar e, posteriormente, visualizar o resultado da pesquisa efectuada. Caso existam mapas de tópicos que descrevam o plano das disciplinas solicitadas ou que contenham tópicos que correspondam às palavras-chave submetidas também pode ser disponibilizada a navegação semântica para essas disciplinas;
- Os formadores e os administradores dos sistemas remotos podem usar os utilitários *on-line* para descrever semanticamente os sistemas remotos registados no sistema WSE (Anotação Semântica de Sistemas Remotos), para além de poderem submeter ou editar mapas de tópicos e alterações à ontologia do sistema. Esta camada de metadados ficará armazenada na BD do sistema WSE;
- O gestor do sistema WSE pode aceder incondicionalmente a todas as funcionalidades disponibilizadas pela aplicação e utilitários do servidor de agentes.

Nível intermédio:

Este nível tem como finalidade a gestão do sistema WSE, mais concretamente do servidor de agentes móveis. As suas principais funcionalidades são:

- Fornecer os agentes de interface consoante o perfil do utilizador;
- Criar, enviar, gerir e receber os agentes de pesquisa móveis de acordo com os metadados e ontologias inerentes aos sistemas remotos registados no sistema WSE;
- Facultar um mecanismo de inferência sobre os documentos XTM que descrevem os cursos submetidos pelos formadores, permitindo a navegação semântica sobre os resultados dessa inferência;
- Gerir a informação, nomeadamente a inscrição manual de utilizadores e o registo inicial dos sistemas externos (sistemas de e-Learning, repositórios de objectos de aprendizagem e outros servidores de conteúdos que favoreçam a pesquisa de LOs de acordo com as condições específicas do sistema WSE);

- Garantir a comunicação necessária entre os diversos objectos que constituem o sistema.

Nível fornecedor de LOs:

Uma vez que os agentes de pesquisa móveis não têm capacidade para interagir com as bases de dados e os documentos semânticos dos sistemas remotos, o nível fornecedor de LOs tem como responsabilidade disponibilizar as interfaces de acesso aos dados e aos documentos que descrevem esses dados. Estas interfaces de acesso designam-se agentes de informação e têm como principais funcionalidades:

- Fornecer o acesso às bases de dados específicas dos sistemas remotos;
- Receber os pedidos dos agentes de pesquisa e redireccioná-los para o motor de inferência para realização da pesquisa semântica;
- Obter os resultados da pesquisa e devolvê-los ao servidor de agentes móveis.

Em cada um destes níveis podem ser identificadas diferentes camadas de informação. No nível cliente é disponibilizada uma **camada de consulta** para parametrizar a pesquisa semântica. Os utilizadores podem questionar o sistema especificando informação sobre o tipo de LOs e de disciplinas ou cursos de formação que desejam procurar através de interfaces de pesquisa semântica básica e de pesquisa semântica avançada. Genericamente, a pesquisa básica fornece um primeiro nível de metadados existente no próprio sistema WSE e referente à anotação semântica dos sistemas remotos feita pelo gestor da WSE e complementada pelos formadores ou administradores dos sistemas remotos, enquanto que a pesquisa avançada permite refinar o primeiro nível de metadados e especificar um segundo nível de metadados referente à descrição semântica existente nos sistemas remotos.

No nível intermédio é disponibilizada a **camada de metadados, ontologias e mapas de tópicos** que permitem descrever os sistemas remotos e os seus cursos. Os agentes de pesquisa são orientados em conformidade com os metadados e ontologias existentes. Tal como mencionado anteriormente, é desejável a existência de um módulo no sistema WSE que disponibilize um motor de inferência para manipular e consultar os documentos XTM locais com vista a proporcionar navegação semântica em disciplinas ou cursos disponibilizados pelos formadores. Para além disso, a base de dados do sistema também regista informação acerca dos utilizadores e dos agentes que estes possuem a circular na rede.

Finalmente, a **camada de objectos de aprendizagem e outros objectos de informação** corresponde aos recursos educativos existentes nas bases de dados dos sistemas remotos. Tal como referimos, é conveniente que esses objectos de aprendizagem estejam devidamente descritos de modo a permitir que os agentes de informação devolvam resultados o mais precisos e filtrados possível para satisfazer os requisitos dos utilizadores.

O objectivo desta arquitectura é não só contribuir para a adopção de um modelo de estruturação, representação e procura de conteúdos educacionais, mas também abrir caminho para outros trabalhos que venham a contribuir para a criação de uma infra-estrutura segundo a nossa visão integradora das tecnologias subjacentes ao e-Learning e à Web Semântica.

6.5.1- Agente de Interface

O agente de interface corresponde a um agente estacionário que é executado no cliente. Este componente do sistema é descarregado via HTTP do Servidor Web a partir do servidor de agentes para a máquina cliente do utilizador. Desde que o agente de interface esteja carregado no cliente, as funcionalidades são disponibilizadas comunicando directamente com o servidor de agentes, tal como ilustrado na Figura 88.

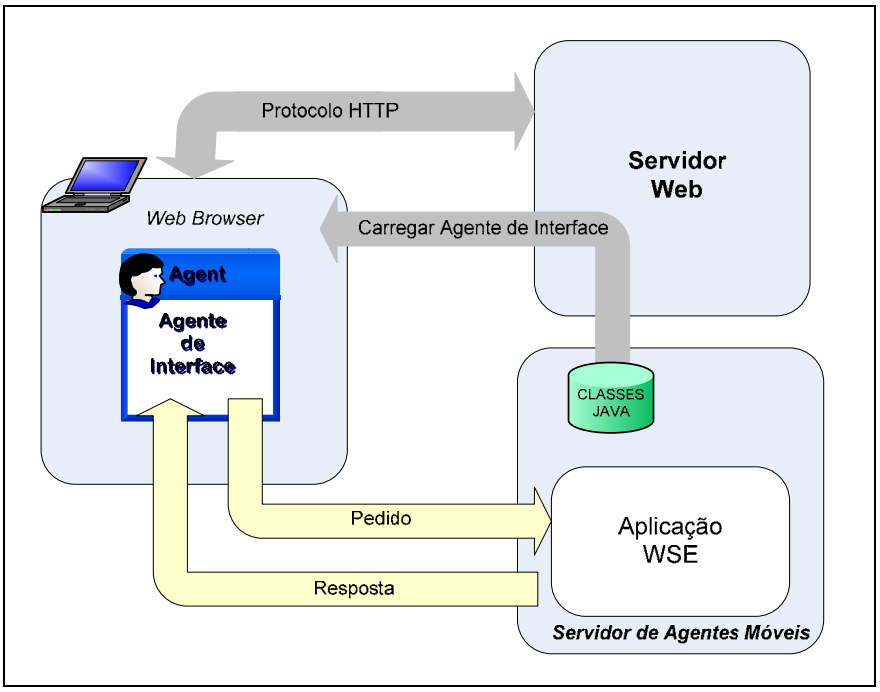


Figura 88 – Comunicação entre agente de interface e servidor de agentes

Tal como especificado na Figura 83 – Diagrama de estados do agente de interface, as principais funcionalidades deste componente do sistema WSE são: a autenticação do utilizador, a criação do agente de pesquisa móvel, a listagem de cursos ou LOs, a destruição do agente de pesquisa móvel e a actualização do estado do agente de pesquisa móvel.

Autenticação do utilizador:

Uma vez que não é necessário que o utilizador esteja ligado ao sistema para que os seus agentes móveis continuem à procura de LOs ou cursos, será necessário identificá-lo para que, aquando da sua próxima visita ao sistema, o servidor de agentes possa disponibilizar-lhe as listagens de disciplinas ou LOs entretanto devolvidas.

O diagrama de actividade do processo de autenticação de um utilizador é apresentado na Figura 89.

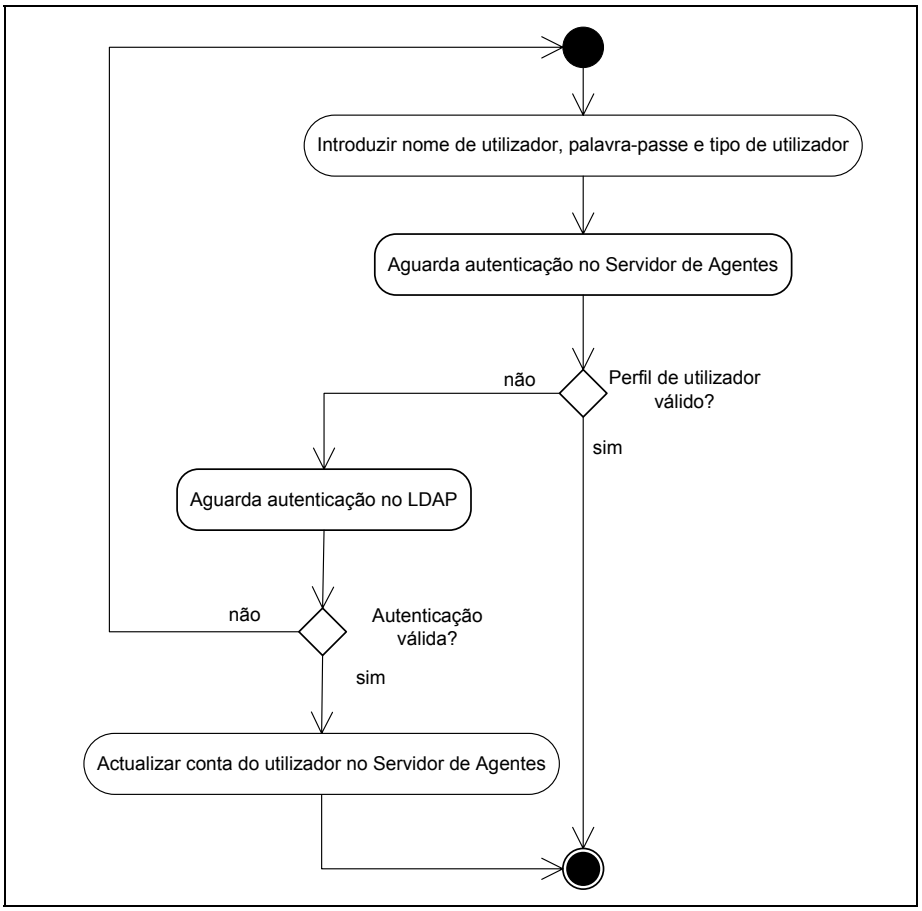


Figura 89 – Diagrama de actividade: autenticação do utilizador

Os utilizadores não registados podem usar a conta de visitante (*guest*) com a qual podem solicitar o registo no servidor de agentes. A autenticação poderá ser realizada por um servidor LDAP (servidor central de autenticação onde os alunos e professores de uma

instituição são autenticados através do nome de utilizador e da respectiva palavra-passe, sendo também considerandos automaticamente formandos e formadores pelo servidor de agentes, respectivamente), evitando que os utilizadores dos sistemas remotos tenham que realizar novamente o registo. Ou seja, aquando do primeiro acesso ao sistema com uma conta de utilizador válida do servidor LDAP, a tabela de utilizadores do servidor de agentes será actualizada com o nome de utilizador, palavra-passe e tipo de utilizador.

Criação de um agente de pesquisa móvel:

Com base na informação especificada pelo utilizador através das interfaces de pesquisa, o servidor de agentes permite a criação de agentes de pesquisa móveis desde que esse utilizador não tenha já um agente igual a circular na rede ou não tenha sido atingido o número máximo de agentes móveis previamente especificado pelo gestor do sistema WSE. Assim, evitamos a duplicação de agentes de pesquisa em circulação na rede. Ou seja, se um determinado utilizador especificar um agente móvel com as mesmas palavras-chave e parâmetros de um agente em circulação na rede, o servidor de agentes impedirá a sua criação até que esse agente regresse ou seja destruído.

O diagrama de actividade do processo de criação de um agente de pesquisa móvel é apresentado na Figura 90.

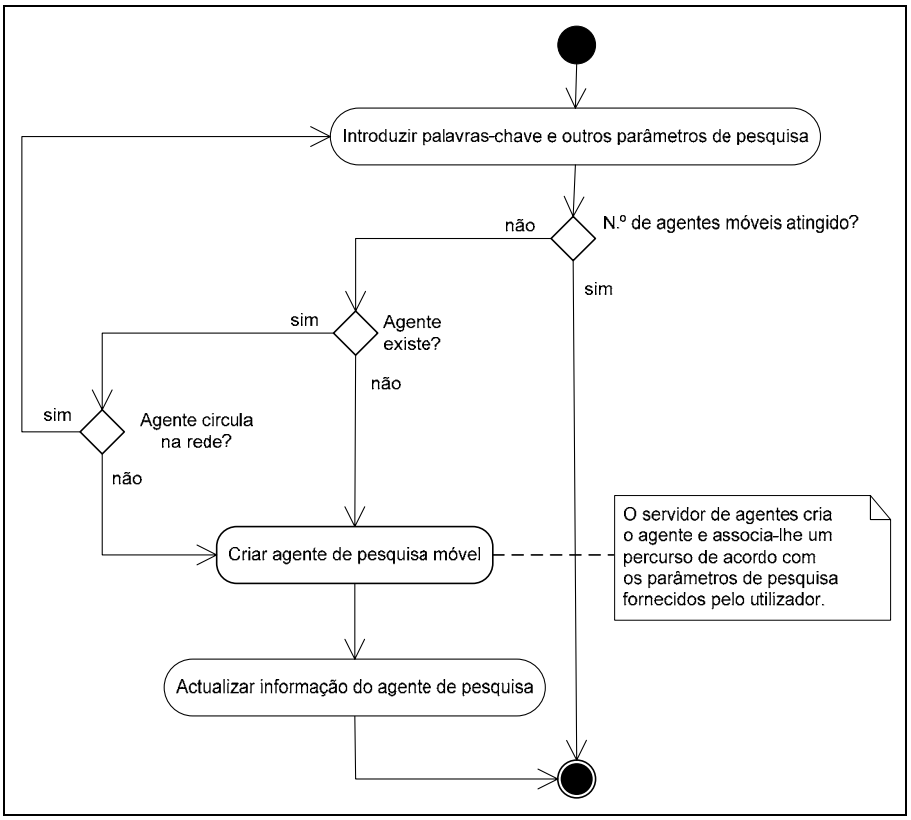


Figura 90 – Diagrama de actividade: criação de um agente de pesquisa móvel

Listagem de cursos ou LOs:

À medida que o agente vai migrando de servidor para servidor, a informação devolvida pelos agentes de informação vai sendo apresentada ao utilizador. Essa informação é apresentada sob a forma de listagens com a designação e respectivo link e descrição. Tal como referimos, seria desejável a organização desses resultados em mapas de tópicos de forma a proporcionar uma navegação semântica. Não obstante, a arquitectura actual inclui uma funcionalidade alternativa que permite através de um motor de inferência para documentos XTM pesquisar se existem disciplinas ou cursos de formação armazenados localmente cujos tópicos correspondam às palavras-chave introduzidas.

Destruição de um agente de pesquisa móvel:

O diagrama de actividade do processo de destruição de um agente de pesquisa móvel é apresentado na Figura 91.

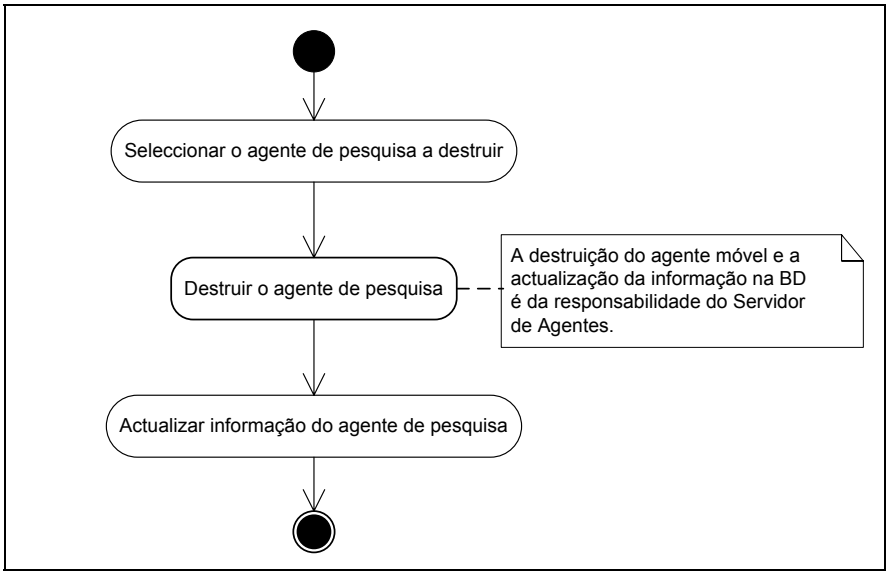


Figura 91 – Diagrama de actividade: destruição de um agente de pesquisa móvel

Caso o utilizador pretenda anular uma pesquisa submetida, esta funcionalidade permite destruir o agente de pesquisa correspondente que se encontra em circulação na rede. Para tal, o utilizador através do agente de interface deverá seleccionar esse agente e solicitar a sua destruição. Solicitada a destruição do agente de pesquisa, o agente de interface fica a aguardar a confirmação por parte do servidor de agentes. Cabe portanto ao servidor de agentes "matar" o agente, ou seja, fazer com que ele deixe de circular na rede. Uma vez que a informação sobre o agente de pesquisa e os resultados da pesquisa que vão chegando ao servidor de agentes são armazenados em tabelas da base de dados do sistema WSE, eliminar

os registos inerentes a esse agente é o primeiro passo para destruir o agente. No entanto, se o agente de pesquisa ainda não tiver regressado, o servidor de agentes terá que previamente informar os agentes de informação sobre o agente que vai deixar de existir. A possibilidade de "matar" o agente antes dele regressar permite diminuir o tráfego na rede e evitar o registo de informação na BD do sistema que o utilizador já não pretende.

Actualização do estado do agente de pesquisa:

No âmbito do agente de interface, a actividade de actualização do estado de um agente de pesquisa é requerida aquando da criação desse agente e aquando da sua destruição, tal como se pode observar na Figura 90 e na Figura 91.

De referir ainda que, caso o utilizador tencione sair do sistema e ainda existam agentes de pesquisa que não regressaram, pode optar ou não por manter os seus agentes em migração. Caso saia do sistema mantendo os agentes, aquando da sua próxima entrada no sistema, a informação entretanto devolvida ser-lhe-á apresentada. Caso contrário, todos os agentes serão automaticamente destruídos.

Em suma, as funcionalidades disponibilizadas pelo agente de interface constituem pedidos que são atendidos pelo servidor de agentes móveis, já que é a ele que compete gerir praticamente todo o sistema. Assim, o agente de interface fornece ao utilizador um nível de transparência e de abstracção elevados sobre os detalhes do servidor de agentes móveis.

6.5.2- Servidor de Agentes

Tomando por base a arquitectura genérica de agentes móveis da Figura 69 e as responsabilidades e funcionalidades especificadas anteriormente, a arquitectura do servidor de agentes móveis para o sistema WSE aqui proposto é constituída por diversos componentes que se relacionam entre si, tal como apresentado na Figura 92.

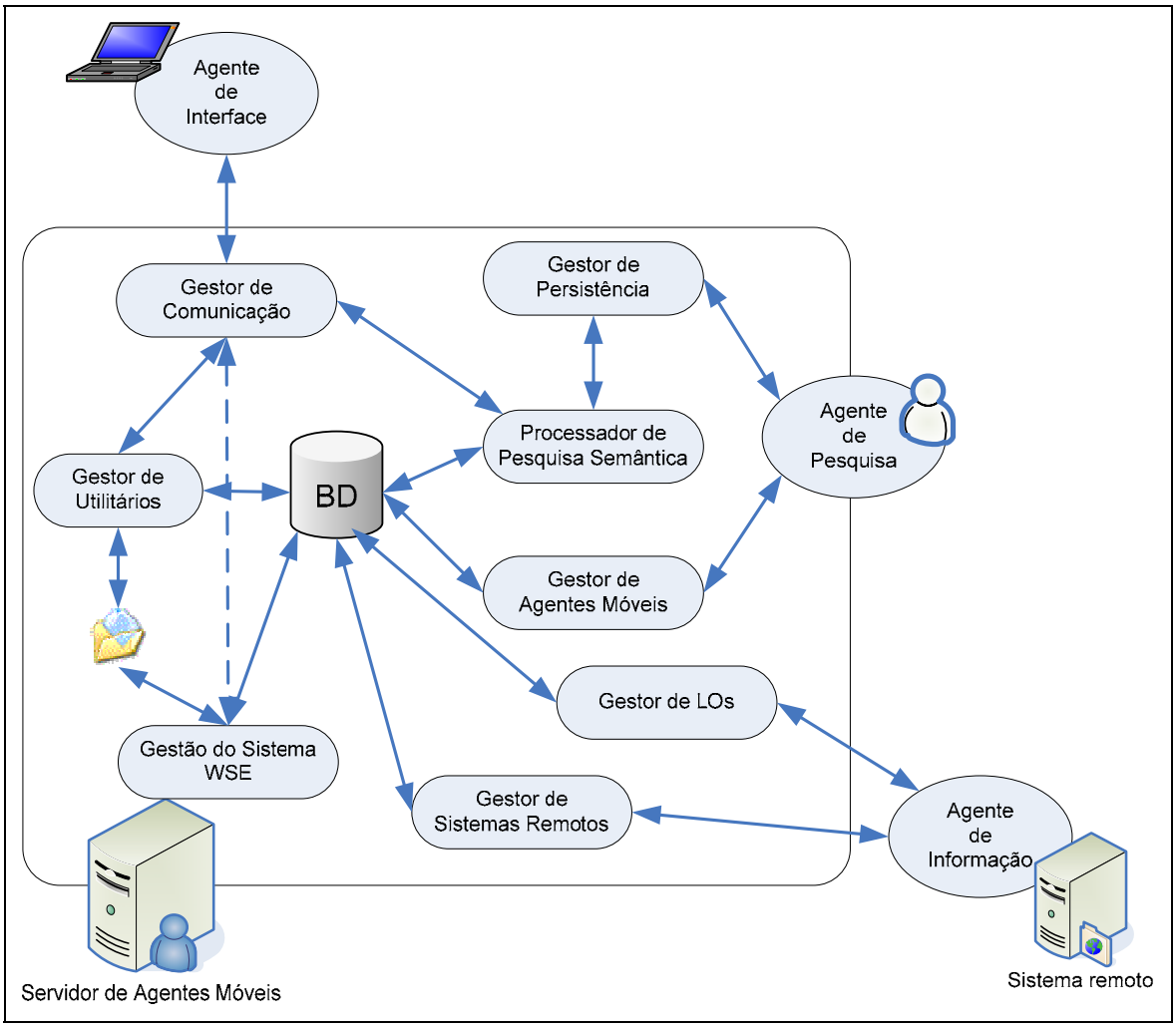


Figura 92 – Arquitectura do servidor de agentes

Seguidamente, apresentamos de forma sucinta cada um destes componentes.

Gestor de comunicação:

É o componente responsável pela comunicação entre os agentes de interface e o servidor de agentes móveis. Tal como referimos anteriormente, o agente de interface solicita os serviços do servidor de agentes para autenticar o utilizador e desencadear os processos de criação e de destruição de agentes. Estas funcionalidades serão asseguradas remotamente pelo gestor de comunicação. No caso do formador ou administrador dos sistemas remotos, o gestor de comunicação também deverá gerir os pedidos inerentes aos utilitários, tais como: anotação semântica e submissão/edição de ontologias e mapas de tópicos.

Gestor de utilitários:

Quando é solicitada a anotação semântica de um sistema externo ou remoto, este componente é responsável pelas consultas às ontologias (ou mesmo aos mapas de tópicos) e,

posteriormente, por actualizar a base de dados do sistema. Aquando do *upload* ou da edição de mapas de tópicos ou de ontologias este componente é responsável pela actualização da base de dados e pela geração automática dos documentos XTM ou OWL e respectivo armazenamento na pasta correspondente.

Processador de pesquisa semântica:

Quando é solicitada a criação de um agente de pesquisa através do agente de interface, este componente tem como principais responsabilidades: seleccionar os sistemas remotos de acordo com os parâmetros indicados pelo utilizador e com base nos metadados e ontologias; especificar o percurso do agente de pesquisa móvel; e criar esse agente de pesquisa.

Gestor de persistência:

Este componente tem como principal responsabilidade enviar os agentes de pesquisa para os servidores remotos e recebê-los quando regressam ao servidor de agentes. Por conseguinte, a missão deste componente é assegurar que os agentes de pesquisa são recebidos pelos gestores de persistência das máquinas remotas.

Gestor de agentes móveis:

Este componente tem como principal responsabilidade assegurar a comunicação entre os agentes de pesquisa móveis e o servidor de agentes. Um agente de pesquisa, quando regressa ao servidor de agentes após cumprir o seu percurso, informa este componente da sua chegada. Consequentemente, o gestor de persistência actualiza a BD do sistema WSE.

Gestor de sistemas remotos:

Este componente permite que os agentes de informação notifiquem o servidor de agentes do seu estado. A responsabilidade do gestor de sistemas remotos é pois garantir que essa informação se mantém actualizada. Assim, evita-se que os percursos dos agentes de pesquisa contenham máquinas remotas nas quais o agente de informação está desligado.

Gestor de LOs:

Uma vez que o agente de pesquisa entrega o pedido ao agente de informação e segue o seu caminho migrando para outra máquina, compete ao agente de informação comunicar ao servidor de agentes a existência de objectos de aprendizagem que corroboram o pedido. Este componente é não só responsável por esta comunicação, mas também pelo armazenamento na BD do sistema WSE das listagens de informação e da respectiva associação ao utilizador e ao agente que as solicitou.

Gestão do Sistema WSE:

Tal como referido anteriormente, o componente de gestão do sistema WSE inclui todas as funcionalidades para administrar o servidor de agentes móveis, suas aplicações e utilitários, das quais se destacam: gerir utilizadores e sistemas remotos, anotar semanticamente sistemas remotos, submeter e editar mapas de tópicos e ontologias, visualizar informação sobre todos os agentes de pesquisa criados, definir o número máximo de agentes de pesquisa a circular na rede por utilizador. A grande maioria dessas funcionalidades está também disponível via Web através de um agente de interface que as solicita remotamente ao gestor de comunicação.

Finalmente, a BD do sistema que propomos tem como principal finalidade armazenar informação sobre os utilizadores, os sistemas aderentes e os seus metadados, os agentes de pesquisa e seus donos (utilizadores que solicitaram pesquisas) e os resultados da pesquisa (listagens dos vários objectos de aprendizagem independentemente da sua granularidade). De salientar que os objectos de aprendizagem ou outros objectos de informação de baixa granularidade e as disciplinas ou cursos de formação (enquanto objectos de granularidade elevada) serão armazenados indistintamente na base de dados considerando-se ambos objectos de aprendizagem. No entanto, para o utilizador que os solicita serão apresentados de forma diferente, uma vez que são disponibilizadas modalidades de pesquisa diferentes, para além de que as especificações de metadados permitem expressar a granularidade do objecto de aprendizagem. Por exemplo, a especificação LOM permite expressar a granularidade através dos elementos *Structure* e *Aggregation Level*. Normalmente, um LO com *Structure*="*atomic*" terá *AggregationLevel*=1. Enquanto que um LO com *Structure*="*collection*", "*linear*", "*hierarchical*" ou "*networked*" terá *AggregationLevel*=2, 3 ou 4. Por conseguinte, a um curso ou meta-disciplina (conjunto de módulos, disciplinas ou acções de formação) corresponderá um nível de agregação de 4, a uma disciplina corresponderá um nível de agregação de 3 e um recurso ou actividade terá nível de agregação de 1 (ou mesmo de 2 caso se trate de uma lição ou recurso similar).

Tal como referido anteriormente, a especificação de requisitos permitiu-nos identificar os principais objectos e classes do sistema. Na Figura 93 é apresentado o diagrama de classes da base de dados do sistema WSE. Os metadados e ontologia serão especificados no capítulo VII. Mais pormenores sobre a ontologia para o sistema WSE podem ser visualizados no anexo G.

O servidor de agentes é responsável pela gestão de toda a informação relativa aos agentes de pesquisa móveis que cria e envia para a rede, para além de gerir a informação

respeitante aos agentes de interface (utilizadores) e aos agentes de informação (sistemas remotos) registados no sistema. Os sistemas remotos são descritos semanticamente através de metadados que por sua vez se baseiam em ontologias, nomeadamente na ontologia do sistema, tal como se pode observar nas relações de dependência representadas na Figura 93.

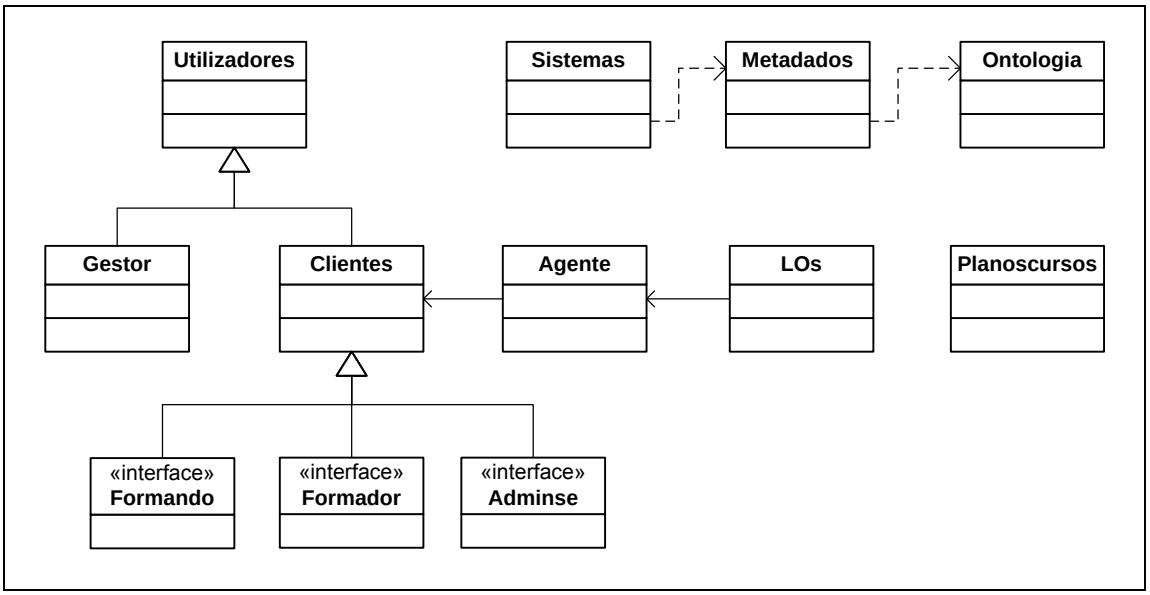


Figura 93 – Diagrama de classes: base de dados do sistema WSE

O servidor de agentes constitui assim o núcleo do sistema uma vez que regista e gere toda a informação necessária ao funcionamento do sistema.

Desta forma, quando um formando pretende procurar LOs, o agente de interface correspondente invoca o servidor de agentes, através do componente gestor de comunicação para solicitar a criação de um agente de pesquisa.

O gestor de comunicação recebe as palavras-chave e restantes parâmetros e reenvia-os para o processador de pesquisa semântica. Por sua vez, este componente define o percurso para o agente de pesquisa móvel com base nos metadados e ontologia inerentes aos sistemas remotos registados na BD do sistema e cujo agente de informação se encontra activo ou em execução. Definido o percurso do agente de pesquisa, cria-o. A identificação deste agente de pesquisa é armazenada no registo do seu dono (utilizador formando) existente na BD do sistema.

De seguida, o gestor de persistência encarrega-se de enviar o agente de pesquisa para a primeira máquina remota definida no seu percurso. O agente migra para essa máquina e entrega o seu pedido ao agente de informação e migra para a próxima máquina. Caso existam LOs que correspondam ao pedido, o agente de informação encarrega-se de enviar a resposta (título, descrição e identificador URI) ao componente gestor de LOs que a armazenará na BD

do sistema juntamente com a identificação do agente de pesquisa. Após cumprir o seu percurso, o agente de pesquisa regressa finalmente ao servidor de agentes onde será destruído logo que informe da sua chegada.

A arquitectura proposta baseia-se em objectos que se relacionam entre si. Quando um objecto se relaciona com outro, ele não acede aos seus recursos ou funcionalidades directamente, mas sim através da sua interface. Por conseguinte, a grande maioria das funcionalidades dos diversos componentes desta arquitectura são disponibilizadas através de interfaces. Tal como referimos anteriormente, os agentes de interface comunicam com o servidor de agentes através das interfaces "IAlunoServ" (interface para o formando), "IProfServ" (interface para o formador) e "IAdminseServ" (interface para o administrador do sistema remoto) de acordo com o perfil de utilizador. Face aos requisitos específicos do gestor do sistema WSE, para este utilizador foi definida a interface "IGestorServ". Os agentes de pesquisa comunicam com o servidor através da interface "IAgentMServ" e os agentes de informação através da interface "IAgentLOServ".

Aquando da modelação do comportamento do servidor de agentes ilustrada no diagrama de actividade da Figura 84 já haviam sido identificados os objectos que implementam estas interfaces. Os objectos, as interfaces e os componentes para os quais disponibilizam funcionalidades são apresentados na Tabela 6.

Comunicação com Agente	Objecto/Classe	Interface	Componentes
Agente de interface para o formando	ClientServ	IAlunoServ	Gestor de comunicação Processador de pesquisa semântica
Agente de interface para o formador	ClientServ	IProfServ	Gestor de comunicação Processador de pesquisa semântica Gestor de utilitários
Agente de Interface para o administrador de sistema remoto	ClientServ	IAdminseServ	Gestor de comunicação Processador de pesquisa semântica Gestor de utilitários
Gestor do sistema WSE	GestorServ	IGestorServ	Gestão do Sistema WSE
Agente de pesquisa móvel	AgentMServ	IAgentMServ	Gestor de agentes móveis
Agente de informação para sistemas remotos	AgentLOServ	IAgentLOServ	Gestor de Los Gestor de sistemas remotos

Tabela 6 – Classes e interfaces do servidor de agentes

O objecto “ClientServ” implementa as interfaces que disponibilizam todas as funcionalidades aos utilizadores do sistema através do agente de interface. Após a autenticação no sistema, o utilizador passa a ter ao seu dispor um leque de funcionalidades consoante o seu perfil. Tal como referimos anteriormente, os agentes de interface solicitam ao servidor de agentes funcionalidades como criação de agentes de pesquisa, destruição de agentes e actualização de informação na base de dados. Para tal, todas essas funcionalidades requisitam previamente a identificação do utilizador que as solicita, de modo a possibilitar actualizações posteriores da BD do sistema, nomeadamente actualização de informação inerente aos agentes e à informação devolvida.

O objecto “GestorServ” implementa a interface para a gestão de todo o sistema WSE. Após a autenticação do gestor, este objecto disponibiliza um conjunto de funcionalidades para administração da BD do sistema, nomeadamente ao nível da visualização, inserção, alteração e eliminação de informação relacionada com os utilizadores e seus agentes (incluindo a definição do número máximo de agentes de pesquisa móveis por utilizador), sistemas remotos e seus metadados (anotação semântica dos sistemas remotos e edição da ontologia do sistema), para além da manutenção e visualização de outra informação de suporte.

O gestor do sistema WSE será responsável por validar e editar propostas de registo de sistemas remotos e de utilizadores, para além de monitorizar o sistema. De referir ainda que, caso o estado de um agente de informação afecto a um sistema remoto aderente permaneça desligado durante mais de 24 horas, o gestor será avisado aquando da sua entrada no sistema.

O objecto “AgentMServ” implementa a interface que permite a comunicação entre os agentes de pesquisa e o servidor de agentes, para além de fornecer as funcionalidades afectas ao gestor de agentes móveis. Quando um agente de pesquisa regressa ao servidor de agentes, as informações sobre esse agente existentes na BD do sistema são actualizadas após identificar o seu dono. Caso o seu dono se encontre ainda ligado ao sistema, o agente de interface correspondente é informado do regresso desse agente. A Figura 94 representa o diagrama de actividade do objecto “AgentMServ”.

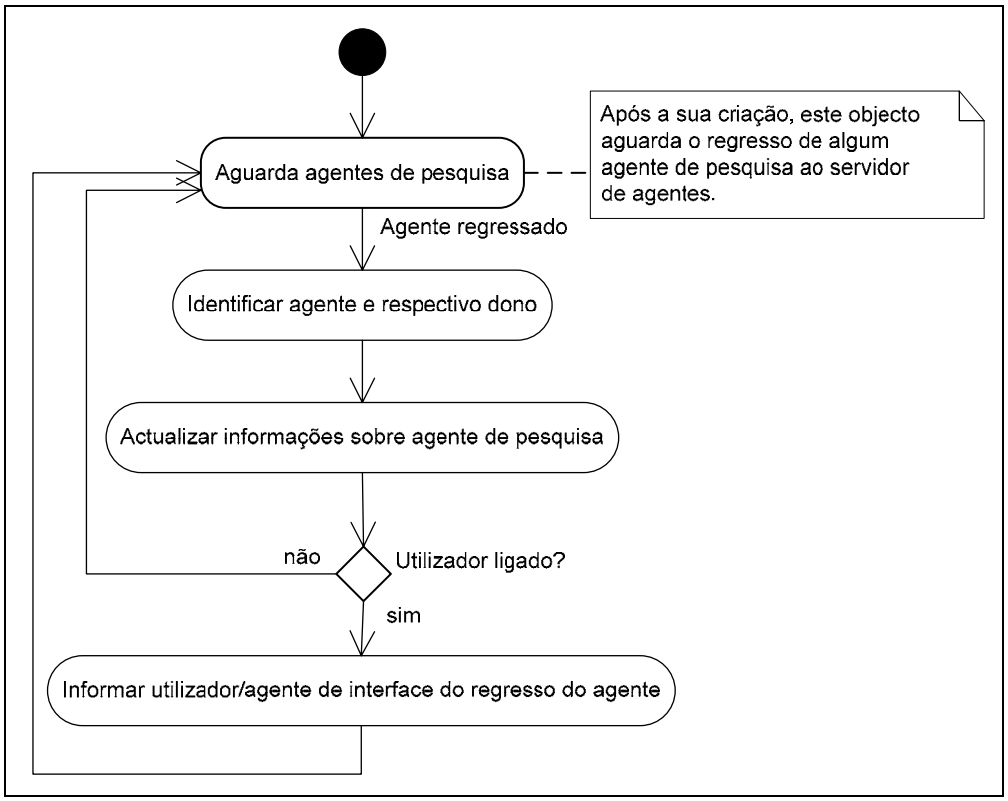


Figura 94 – Diagrama de actividade: objecto AgentMServ

O objecto "AgentLOServ" implementa a interface que permite a comunicação entre os agentes de informação inerentes aos sistemas remotos aderentes ao sistema e o servidor de agentes. Ou seja, fornece as funcionalidades necessárias aos componentes "gestor de LOs" e "gestor de sistemas remotos".

O componente "gestor de sistemas remotos" permite que os agentes de informação informem o servidor de agentes do seu estado. Enquanto que o componente "gestor de LOs" permite que os agentes de informação respondam ao pedido e enviem a informação resultante para armazenar na BD do sistema WSE de acordo com a identificação do utilizador e do agente que a solicitou. Neste caso será necessário informar o agente de interface do utilizador das novas listagens de objectos de aprendizagem que correspondem a um determinado agente de pesquisa.

Tal como representado na Figura 95, as principais tarefas disponibilizadas pelo objecto "AgentLOServ" são: actualizar o estado do agente de informação e armazenar informação na BD do sistema.

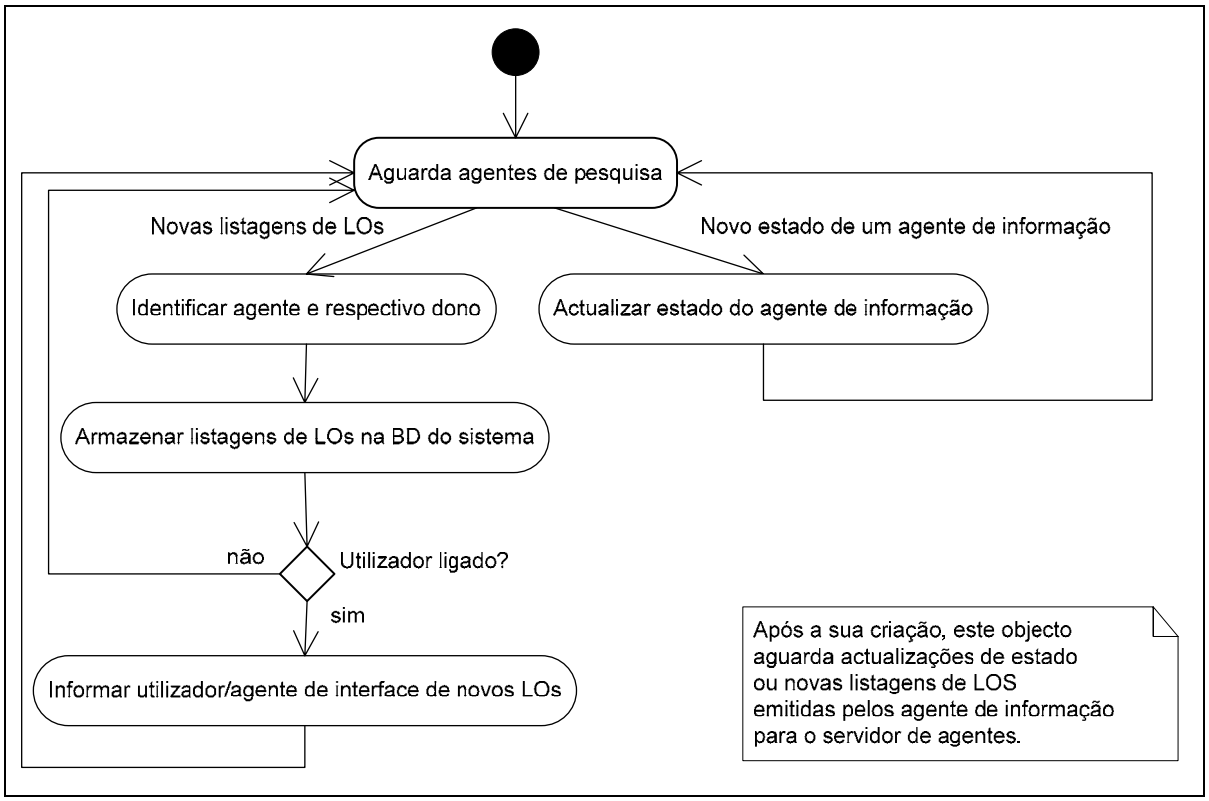


Figura 95 – Diagrama de actividade: objecto AgentLOServ

O envio de agentes de pesquisa para a rede accionado pelo gestor de persistência do servidor de agentes baseia-se no percurso previamente definido para esse agente. Quando o agente de pesquisa chega ao primeiro destino identifica-se perante o agente de informação dessa máquina remota. Caso o utilizador não tenha entretanto cancelado a pesquisa semântica, entrega o seu pedido ao agente de informação. Caso contrário termina a sua execução. O último destino do agente corresponde à origem, ou seja, ao servidor de agentes de onde partiu. Portanto, se o próximo destino não for o endereço do servidor de agentes, então migra para a máquina do próximo sistema remoto. Caso contrário regressa e informa o servidor de agentes da sua chegada. Quando o agente de pesquisa migra para uma máquina remota, é o gestor de persistência inerente ao agente de informação dessa máquina remota que se responsabiliza pelo seu reenvio, garantindo que é recebido pela próxima máquina remota.

Uma vez que não é da responsabilidade do agente móvel a devolução dos resultados, o agente ganha em mobilidade uma vez que não tem que transportar informação, a rede ganha em largura de banda uma vez que o processamento ocorre do lado dos servidores exteriores e o utilizador ganha em rapidez na obtenção de resultados já que pode ir visualizando a informação à medida que esta vai sendo enviada pelos agentes de informação, não tendo que aguardar pelo regresso do seu agente de pesquisa.

O diagrama de actividade do agente de pesquisa móvel é apresentado na Figura 96.

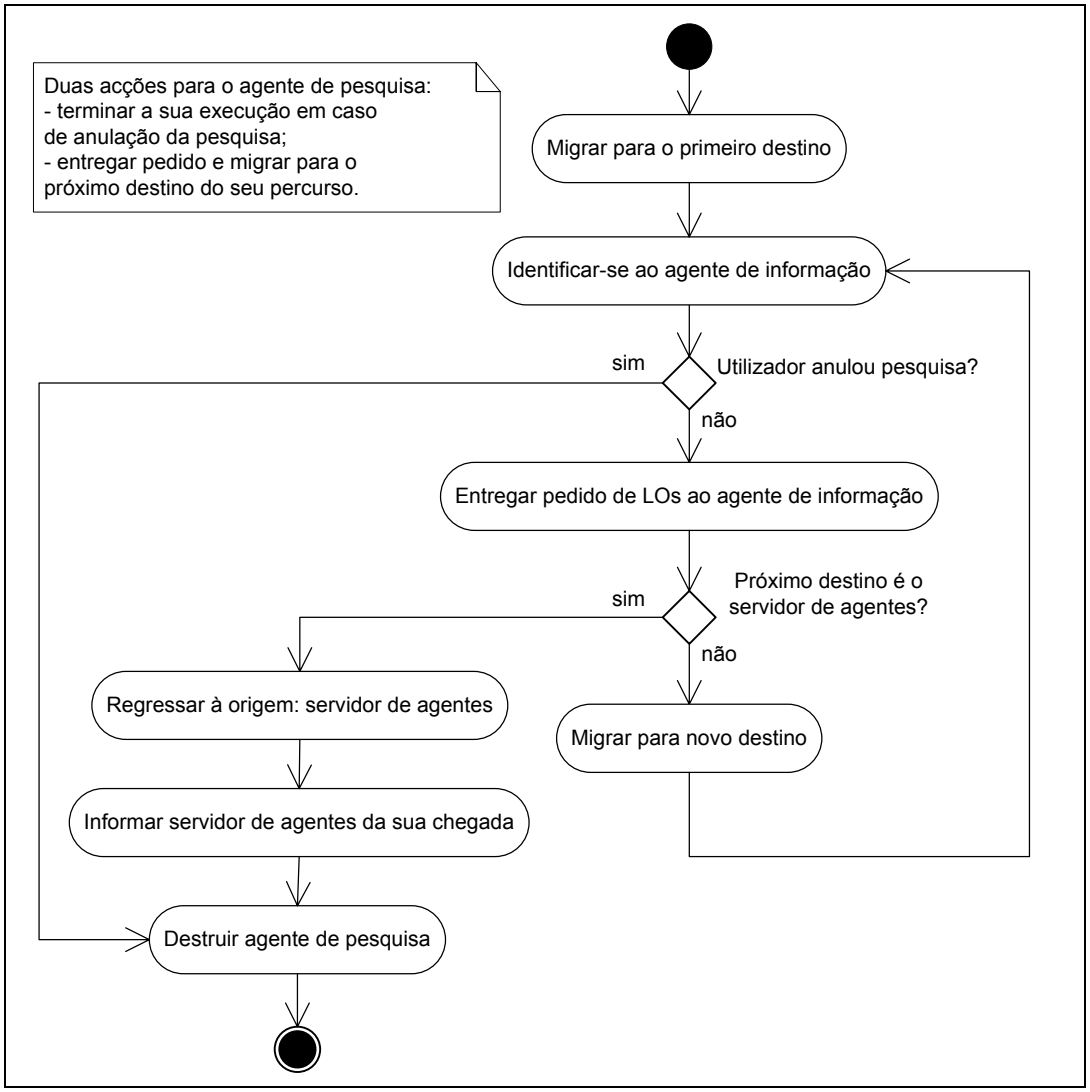


Figura 96 – Diagrama de actividade: agente de pesquisa móvel

6.5.3- Agente de Informação

Os agentes de informação são agentes estacionários executados nos servidores dos sistemas remotos registados no sistema. As funcionalidades deste componente do sistema WSE são:

- Pesquisar semanticamente metadados nas bases de dados e/ou nos metadados expressos RDF/XML (que incluem a expressividade em RDFS ou OWL) com base nas palavras-chave e outros parâmetros entregues pelo agente de pesquisa;

- Enviar para o servidor de agentes o resultado da pesquisa pedida pelo agente de pesquisa (título, descrição e URI);
- Reenvio de agentes de pesquisa para a próxima máquina remota, garantindo a sua recepção no novo destino.

Na Figura 97 é apresentada a arquitectura do agente de informação respeitante a um sistema remoto (por exemplo, um sistema de e-Learning Moodle), os seus componentes e as suas relações, bem como as ligações com os restantes componentes da arquitectura do servidor de agentes.

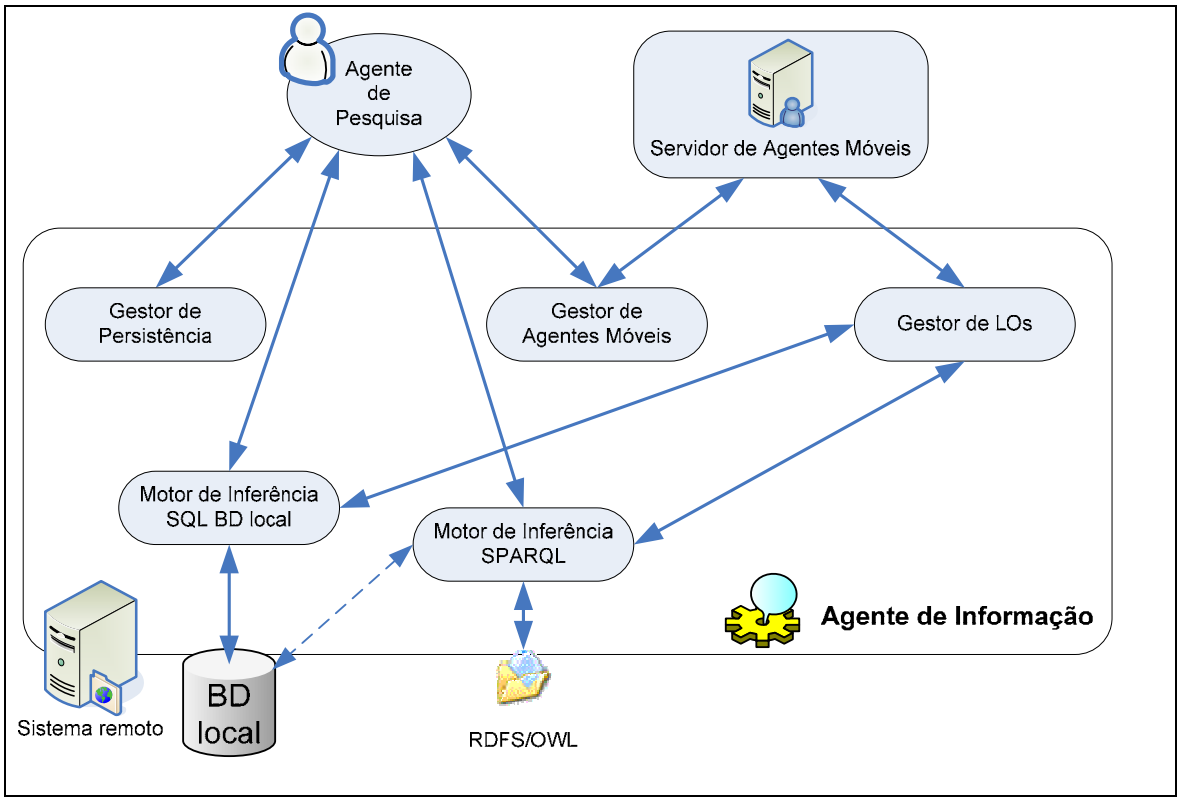


Figura 97 – Arquitectura do agente de informação para um sistema remoto

O **gestor de persistência** de um agente de informação é responsável pela recepção e execução dos agentes de pesquisa que chegam à máquina onde se encontra este agente de informação. Caso o servidor de agentes não tenha informado de que a pesquisa foi anulada, este gestor de persistência encarrega-se de enviar o respectivo agente de pesquisa para a próxima máquina remota definida no percurso do agente, garantindo que o mesmo é recebido pelo gestor de persistência correspondente dessa máquina. Caso contrário, o agente de pesquisa terá que ser destruído. Esta funcionalidade é da responsabilidade do gestor de agentes móveis.

O componente "gestor de agentes móveis" encarrega-se de manter actualizada a lista de agentes a destruir e o estado do servidor de agentes. Assim, se o servidor passar para o estado de desligado, sendo impossível receber os agentes que tentem regressar, este componente causa a destruição de todos os agentes de pesquisa que entretanto cheguem à máquina, mesmo que não constem da sua lista de agentes a destruir.

Se o agente não constar da lista de agentes a abater, o pedido de pesquisa é entregue a um dos motores de inferência ou a ambos de acordo com os parâmetros que acompanham as palavras-chave da pesquisa. Se o percurso inclui parâmetros para que este agente de informação seja simultaneamente capaz de interrogar a base de dados e de realizar inferências semânticas sobre os documentos RDF(S)/OWL, então o pedido é entregue ao **motor de inferência SQL para a BD local** e ao **motor de inferência semântico SPARQL**.

Logo que os resultados sejam devolvidos, o **gestor de LOs** tem a responsabilidade de os enviar ao servidor de agentes.

Para tal, o agente de informação é constituído por dois objectos que implementam duas ou três interfaces consoante o tipo de funcionalidades que o sistema remoto disponibiliza. Na Tabela 7 são apresentadas as classes, as interfaces e os componentes do agente de informação para os quais essas interfaces disponibilizam funcionalidades.

Objecto/Classe	Interface	Componentes
AMPServ	IAMPServbd	Motor de inferência SQL BD local Gestor de LOs
AMPServ	IAMPServrdf	Motor de inferência SPARQL Gestor de LOs
ServidorServ	IServidorServ	Gestor de Agentes Móveis

Tabela 7 – Classes e interfaces do agente de informação de um sistema remoto

Tal como representado no diagrama da Figura 85 – Diagrama de actividade do agente de informação (página 206), a execução de um agente de informação de um sistema remoto inicia-se com o estabelecimento da conexão à BD local e às pastas de metadados e ontologias e com a criação dos dois objectos que disponibilizam as funcionalidades aos agentes de pesquisa e ao servidor de agentes. De seguida, o agente de informação comunica ao servidor de agentes que está ligado e fica a aguardar a chegada de agentes de pesquisa. Caso venha a ser desligado antes de terminar a sua execução, deverá informar o servidor de agentes do seu novo estado e fechar as ligações à BD local e às pastas de documentos semânticos.

Para manter informação actualizada sobre o estado do servidor de agentes e dos agentes que necessitam de ser destruídos, o agente de informação recorre ao objecto "ServidorServ". Caso o agente de informação proceda à destruição de um agente de pesquisa, os restantes agentes de informação terão que ser informados do sucesso dessa actividade.

Para identificar os agentes que chegam à máquina remota e satisfazer os seus pedidos de pesquisa de LOs, o agente de informação usa o objecto "AMPServ". A Figura 98 apresenta as principais actividades deste objecto.

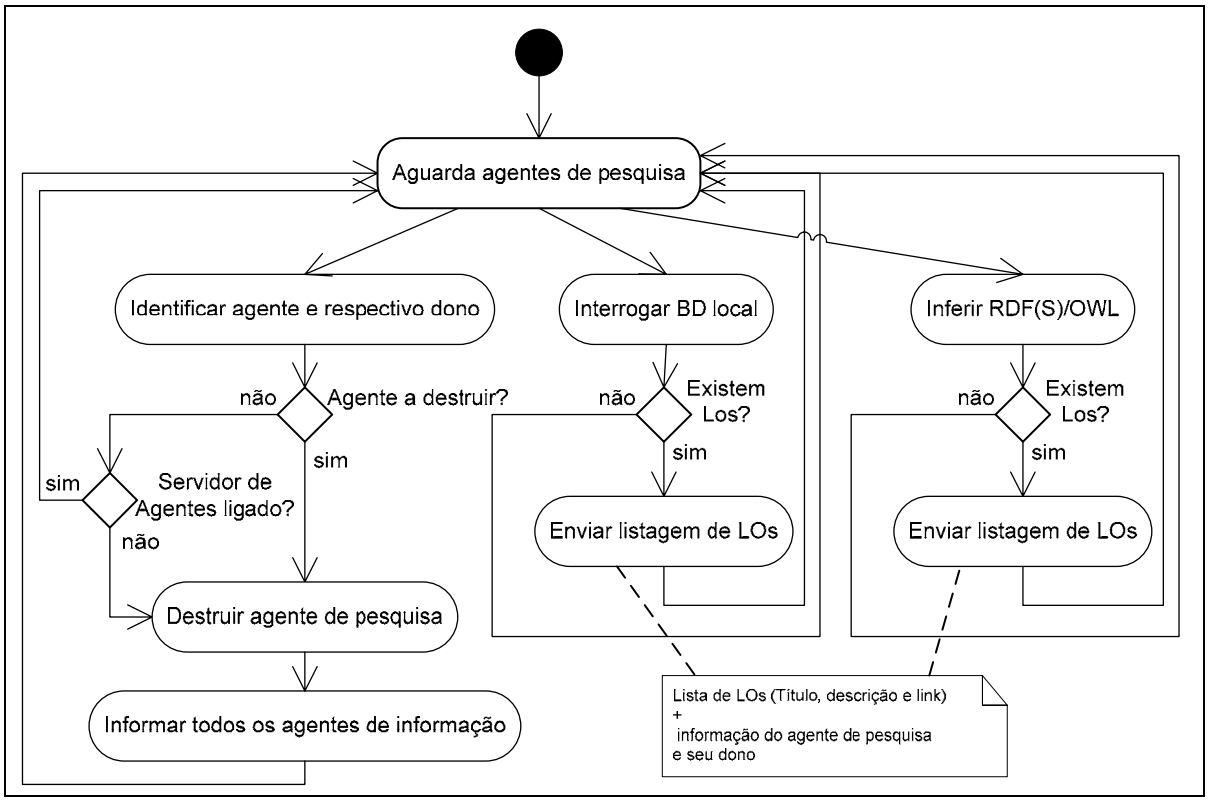


Figura 98 – Diagrama de actividade: objecto AMPServ

De acordo com a análise e especificação apresentadas ao longo deste capítulo, podem ser identificados diferentes tipos de mensagens trocadas entre os vários objectos do sistema:

- **Mensagens síncronas:** são usadas para invocar serviços ou funcionalidades de outro objecto. Estas mensagens são normalmente usadas quando um componente não precisa de realizar nenhum processamento durante o pedido, mantendo-se inactivo enquanto espera pela disponibilização do serviço.
- **Mensagens do tipo *one-way*:** mensagens enviadas num só sentido que são normalmente usadas quando o objecto emissor não requer qualquer resposta do objecto destinatário, continuando a sua execução imediatamente. Por exemplo, quando

o agente de pesquisa entrega o pedido ao agente de informação e segue o seu percurso, sem haver necessidade de esperar pela resposta, uma vez que esta será enviada directamente pelo agente de informação para o servidor de agentes.

- **Mensagens do tipo *multicast*:** são normalmente usadas quando é necessário enviar uma única mensagem para vários destinatários ou invocar a mesma mensagem em múltiplos objectos, optimizando a largura de banda e incrementando a confiabilidade das mensagens de rede. Por exemplo, quando o servidor de agentes informa do seu estado todos os agentes de informação.
- **Mensagens assíncronas:** são usadas para pedidos múltiplos que ocorrem simultaneamente. O objecto emissor continuará o seu processamento durante o pedido de envio da mensagem. Por exemplo: quando o servidor de agentes necessita de informar um utilizador da existência de novas listagens de LOs encontrados, quando o servidor de agentes precisa de informar um determinado grupo de agentes de informação da necessidade de destruir um agente de pesquisa específico, ou quando um agente de informação tem de informar um conjunto de agentes de informação da destruição de um agente de pesquisa específico.

6.6- Resumo

O objectivo deste capítulo era apresentar uma arquitectura para um sistema de recuperação de objectos de aprendizagem baseado nas tecnologias para a Web Semântica, para o e-Learning e para os agentes de software. Para tal, foram apresentados sucintamente os modelos que orientaram as actividades de planeamento e de análise e especificação de requisitos do sistema Web Semântica Educativa (WSE), incluindo a motivação para a realização destas actividades, resultante da reflexão sobre o impacto que a visão da Web Semântica pode originar no contexto educativo em geral e no e-Learning em particular.

Seguidamente, foram apresentadas a modelação do contexto e dos requisitos, a modelação do comportamento e da estrutura e a modelação da arquitectura.

A arquitectura proposta é composta essencialmente por quatro componentes: os "agentes de interface" para permitir aos utilizadores interagirem com o sistema WSE; os "agentes de pesquisa móveis" para transportar os pedidos de pesquisa de informação aos servidores remotos; os "agentes de informação" para recuperar os objectos de aprendizagem dos servidores remotos; e o "servidor de agentes" para gerir e controlar todos os restantes componentes do sistema.

Os componentes desta arquitectura organizam-se em três níveis: nível cliente (agentes de interface para comunicação entre os diferentes perfis de utilizador e a aplicação do servidor de agentes); nível intermédio (servidor de agentes que envia e gere os agentes de pesquisa móveis de acordo com os metadados e ontologias inerentes aos sistemas remotos) e nível fornecedor de LOs (agentes de informação que realizam consultas SQL nas bases de dados dos sistemas remotos ou inferência SPARQL nos documentos RDF(S) e OWL de acordo com os pedidos dos agentes de pesquisa).

A cada um destes níveis correspondem diferentes camadas de informação: a camada de consulta para parametrizar a pesquisa semântica no nível cliente; a camada de metadados, ontologias e mapas de tópicos para descrever os sistemas remotos e os seus cursos no nível intermédio; e a camada de objectos de aprendizagem e outros objectos de informação devidamente descritos com metadados para realizar a pesquisa no nível fornecedor de LOs.

Por conseguinte, esta arquitectura requer uma infra-estrutura distribuída onde agentes móveis e agentes estacionários se relacionem trocando mensagens e informação com vista a recuperar LOs de servidores remotos. O sistema que aqui se propõe não só impulsiona uma maior visibilidade para Web actual, uma vez que permite pesquisar BDs em servidores dispersos, como também constitui um contributo significativo rumo à Web Semântica educativa já que a pesquisa se baseia em duas camadas semânticas: uma baseada na anotação semântica inerente aos sistemas remotos, armazenada na BD do sistema e na ontologia do sistema; e a outra baseada na pesquisa semântica sobre os metadados armazenados nas BDs locais ou na inferência semântica sobre documentos RDF(S)/OWL armazenados num directório local de cada servidor remoto.

7- Desenvolvimento do Sistema WSE

No contexto do desenvolvimento de Sistemas de Informação para a Web podem ser consideradas diversas metodologias de engenharia de software e as respectivas técnicas, linguagens e ferramentas, para a análise, o projecto, a implementação, a instalação e a avaliação do sistema. A modelação relativa à análise e projecto ou especificação do sistema foi apresentada no capítulo anterior. Neste capítulo preocupamo-nos com as características de implementação, instalação e avaliação mais relevantes do protótipo do sistema proposto, de acordo com as prioridades que foi necessário definir.

Genericamente, são descritos os principais componentes do protótipo e as respectivas funcionalidades e as características de implementação e instalação mais relevantes, bem como as actividades de preparação dos sistemas externos ou remotos com vista à sua integração na infra-estrutura distribuída que propomos. Finalmente, são apresentados alguns dos principais resultados referentes à avaliação do protótipo e, em particular, dos módulos e do bloco de pesquisa semântica para um sistema de e-Learning particular, o Moodle.

7.1- Actividade de Desenvolvimento

Nesta última década, foram propostas diversas metodologias de engenharia de software orientadas por agentes (AOSE - *Agent Oriented Software Engineering*):

- Metodologias baseadas na teoria dos agentes (*methodologies based on agent theory*): fornecem uma modelação mais adequada para agentes do que as restantes metodologias, uma vez que se preocupam com as características peculiares dos mesmos. Para o sistema de agentes móveis pode usar-se, por exemplo, a metodologia mGAIA (*mobile Generic Architecture for Information Availability*) ou o ambiente de modelação JAMES (*Java-based Agent Modeling Environment for Simulation*), cobrindo as etapas de análise de requisitos, modelação, especificação, prototipagem, simulação, implementação e monitorização. Mas, para a modelação de agentes

inteligentes, a metodologia Prometheus parece-nos a mais adequada. Outros exemplos: GAIA, MESSAGE/UML, OPM/MAS (*Object-Process Methodology for Multi-Agent System*), ROADMAP (*Role-Oriented Analysis and Design for Multi-Agent Programming*), SODA e Tropos.

- Extensões das metodologias orientadas por objectos (*extensions of object-oriented methodologies*): uma vez que se baseiam em técnicas e linguagens já conhecidas, permitem acelerar a adopção dos agentes. Exemplos: MaSE (*Multi-agent System Engineering*), PASSI e Prometheus.
- Metodologias baseadas na engenharia do conhecimento (*methodologies based on knowledge engineering*): podem constituir uma boa opção na modelação de processos de aquisição de conhecimento. Exemplos: MAS-CommonKADS.
- Metodologias híbridas (*hybrid methodologies*): resultam da integração das características ou pontos fortes de algumas das metodologias anteriores com vista a eliminar ou atenuar algumas das suas limitações. Exemplos: *Skeleton methodology* e *Modular methodology*.

Cada uma destas metodologias tem diferentes pontos fortes e fracos e apresenta características específicas para suportar diferentes aspectos do domínio da aplicação a que se destina. Algumas das metodologias apresentadas recorrem a ferramentas, técnicas e linguagens de modelação tais como UML, a linguagem de modelação normalizada pelo OMG no âmbito da engenharia de software orientada por objectos, ou às suas mais recentes extensões AUML (*Agent Unified Modelling Language*) e AML (*Agent Modelling Language*). A AML é uma extensão da UML que usa os conceitos de AUML, MESSAGE, FIPA-S, OWL, etc.

Face ao exposto, a actividade de desenvolvimento do protótipo do sistema proposto foi orientada globalmente pela metodologia híbrida modular - *modular methodology* - uma vez que esta metodologia permitiu-nos seguir diversas metodologias e modelos consoante o componente ou subsistema que pretendíamos modelar, especificar e implementar. Ou seja, para a análise e projecto do sistema multiagente usámos essencialmente a metodologia MESSAGE, recorrendo a UML/AML para a modelação do contexto, dos requisitos, do comportamento, da estrutura e da arquitectura do sistema. Para orientar o desenvolvimento dos módulos nos sistemas de e-Learning e restantes sistemas remotos recorremos a metodologias orientadas aos objectos baseadas em modelos iterativos e incrementais que

permitiram envolver todos os intervenientes no processo com vista a alargar pouco a pouco a qualidade, o detalhe e o âmbito do sistema, sem termos que nos preocupar com a especificidade dos agentes.

Assim, esta parte do desenvolvimento seguiu o modelo em espiral (Boehm, 1988; Boehm, 2000) como uma extensão ou variante dos modelos iterativos e incrementais. De forma simplista, este modelo pode ser composto pelas seguintes fases: Estudo Prévio, Análise, Projecto, Implementação, Protótipo e Avaliação. A ideia principal é que cada uma destas fases pode ser revisitada tantas vezes quantas as que for necessário até ao final do desenvolvimento. A escolha deste modelo conferiu ao processo de desenvolvimento um carácter genérico e flexível, incremental e evolutivo, participativo e centrado no utilizador, baseado essencialmente em abordagens de prototipagem e na análise dos riscos envolvidos em cada situação ou momento do projecto.

A arquitectura apresentada no capítulo anterior permitiu-nos representar cada um dos componentes e funcionalidades com um nível de abstracção elevado. Ou seja, o componente "servidor de agentes" concentra-se na gestão e controlo dos restantes componentes do sistema, enquanto que cada um dos componentes "agentes de informação" preocupa-se com o processo de acesso à informação do seu servidor. Os agentes de interface abstraem o acesso ao sistema por parte dos utilizadores, enquanto que os agentes de pesquisa móveis abstraem todo o processo de localização e pesquisa de informação. Esta divisão permitiu-nos especificar e delimitar claramente as funcionalidades de cada um destes componentes.

As características da metodologia modular e do modelo em espiral permitiram a escolha da solução que melhor se adequava aos riscos envolvidos em cada situação e momento particulares, pelo que as fases de análise, especificação, implementação e avaliação dos protótipos do sistema apresentadas neste capítulo e no anterior como fases estáticas e bem definidas pouco correspondem à realidade do processo de desenvolvimento, já que cada nova iteração ou ciclo da espiral implicava alterações na análise e especificação e, consequentemente, implementações de protótipos cada vez mais avançados. Obviamente, a alteração de um módulo ou sub-sistema acabaria por originar um novo ciclo de desenvolvimento com alterações em praticamente todas as restantes partes do sistema.

Tal como referimos, este capítulo tem como principal propósito apresentar as fases de construção, instalação e de avaliação dos protótipos deste sistema.

A **construção do sistema** consiste essencialmente na elaboração de todo o software e ficheiros de dados do sistema, ou seja, das aplicações informáticas, da sua interligação, dos testes e da documentação. A construção do sistema pode ser também designada por etapa de implementação ou programação e permite gerar o código necessário ao funcionamento do sistema de acordo com a arquitectura especificada.

A **instalação e a manutenção do sistema** visa não só preparar e instalar o sistema no ambiente para o qual foi projectado, mas também acompanhar o funcionamento do mesmo, resolvendo eventuais problemas e melhorando algumas partes específicas.

Genericamente, os principais modelos (ou aproximações) de Sistemas de Informação para a Web podem ser classificados em: modelos centrados no servidor, modelos centrados no cliente e modelos suportados por infra-estruturas distribuídas.

Nesta perspectiva, podemos considerar que os agentes de interface seguem a filosofia dos modelos centrados no cliente, e, mais concretamente dos modelos baseados em código móvel, uma vez que a máquina virtual do cliente interpreta e executa o código (*applet* Java), que se encontra num ficheiro separado e independente do documento HTML.

O servidor de agentes móveis é claramente um sistema suportado por uma infra-estrutura distribuída, também conhecida por ORB e que disponibiliza os seguintes serviços básicos: criação de objectos que podem ser invocados remotamente, obtenção de referências para objectos remotos, envio de mensagens e invocação de métodos em objectos remotos, gestão de nomes e gestão de interfaces, entre outros.

Finalmente, os servidores externos ou remotos correspondem tipicamente a modelos centrados no servidor. Mas através dos agentes de informação estes servidores são integrados na infra-estrutura distribuída proposta. Para tal, foi necessário escolher uma plataforma de desenvolvimento que facultasse um ambiente de execução distribuído.

Assim, a construção do núcleo do sistema ou servidor de agentes móveis e seus agentes de interface e agentes de informação baseou-se na utilização de duas tecnologias: a linguagem de programação Java e a plataforma de desenvolvimento e execução de agentes Voyager.

O *Java Development Kit* (JDK) é um software desenvolvido pela *Sun Microsystems* que implementa um conjunto básico de ferramentas necessárias para escrever, testar e depurar aplicações e *applets* (SUN, 2004). As aplicações são programas autónomos que apenas necessitam do interpretador Java para serem executados. As *applets* são programas

especialmente criados para serem executados num Web *browser*, mas não possuem a autonomia das aplicações (Carriço e Carriço, 1998).

Em traços gerais, a distribuição de um programa Java é efectuada através de pacotes. Dentro de cada pacote encontram-se vários objectos e cada objecto pertence a uma classe. Os objectos possuem formas de comunicação que permitem o envio de mensagens. Para utilizar uma funcionalidade de um objecto bastará chamar o método correspondente, colocando os parâmetros na sequência correcta.

A linguagem de programação Java é uma linguagem orientada aos objectos e uma candidata natural à construção de aplicações em rede. As características do Java que mais se destacam no âmbito do desenvolvimento de um ambiente baseado em agentes móveis são (Oshima e Lange, 1998):

- Independência de plataforma – sendo uma linguagem compilada e interpretada proporciona portabilidade, isto é, ao compilar um programa escrito em Java (ficheiro "programa.java") é gerado um código intermédio (Java *bytecode* que resulta num ficheiro "programa.class"), que é interpretado na plataforma onde será executada a aplicação, pela máquina virtual de Java. Isto permite criar agentes móveis sem conhecimento prévio de qual é o tipo de computadores onde serão executados;
- Segurança na execução – os programas Java não podem violar a semântica básica da linguagem. Usando os diversos mecanismos de segurança desta linguagem é possível construir um ambiente imune a ataques de agentes móveis mal intencionados, uma vez que todo o código carregado por classes Java está sujeito a restrições de segurança;
- Carregamento dinâmico de classes – as classes são definidas durante a execução, fornecendo um espaço de endereçamento privado para cada agente, o que permite que as classes possam ser carregadas através da rede;
- Programação *multithread* – a execução de agentes no seu próprio processo (ou *thread* de execução) permite a implementação de agentes autónomos. Uma vez que o Java não implementa programação *multithread*, a interacção é realizada por meio de primitivas de sincronização;
- Serialização de objectos – esta característica é crucial para a implementação de agentes móveis, uma vez que permite que os objectos sejam empacotados com as informações necessárias para posterior reconstrução;

- Reflexão – o Java possui mecanismos para obter informações das classes carregadas, permitindo construir agentes com maior conhecimento de si próprios e dos outros agentes.

A plataforma que nos pareceu mais indicada para o desenvolvimento e execução dos agentes inteligentes móveis inerentes ao núcleo do sistema WSE foi a plataforma Java Voyager. Esta plataforma disponibiliza um conjunto de serviços e características que simplificam a construção de aplicações distribuídas. A escolha da Voyager deveu-se essencialmente a:

- Ser um ORB Java;
- Permitir fácil integração com a linguagem Java na construção de aplicações;
- Facilitar o desenvolvimento de aplicações distribuídas baseadas em agentes, com especial destaque para a mobilidade de código ou de agentes;
- Suportar diversas formas de distribuição de eventos/mensagens de comunicação;
- Possuir uma arquitectura em camadas flexível;
- Proporcionar a portabilidade e interoperabilidade necessárias;
- Garantir a segurança, a confiabilidade e a escalabilidade.

O ORB Voyager simplifica e uniformiza o acesso à grande maioria das normas e especificações comummente aceites, nomeadamente ao nível da comunicação, envio de mensagens (*messaging*), serviço de nomes para permitir o acesso aos objectos (*naming service*), serviço de directório que pode ser acedido e partilhado pelos clientes (*directory service*) e serviço para ligação de clientes e servidores que não usam Voyager (*gateway*).

O desenvolvimento com a plataforma Voyager exige que o programador tenha conhecimentos sólidos da linguagem de programação Java e que compreenda os conceitos da computação distribuída. Não obstante, a grande maioria das funcionalidades disponibilizadas pela Voyager são executadas em *background*, sendo completamente transparentes para o utilizador final. Referimo-nos a características tais como a execução remota de classes, a construção remota de objectos, o carregamento dinâmico de classes, a mobilidade de objectos ou mesmo as actividades dos agentes móveis autónomos e inteligentes, entre outras características.

Quanto à construção dos sistemas de e-Learning para testar a funcionalidade do sistema recorremos à plataforma Atutor (2004) e especialmente à plataforma Moodle (2004) não só por serem *open source*, o que as torna interessantes para instituições de ensino/formação que não pretendem adoptar uma filosofia comercial na disponibilização de conteúdos educativos, mas também porque nos pareceram as mais acessíveis para aplicar as tecnologias da Web Semântica aos conteúdos a disponibilizar (no que diz respeito à plataforma Moodle na secção 6.3 apresentaremos mais pormenorizadamente os principais motivos que justificaram a sua escolha). Pelas mesmas razões, optou-se pela construção de um Website ou Portal de conteúdos educativos de um professor ou investigador suportada pela plataforma *Mambo* e pela construção de um sistema de gestão de conteúdos com repositório de objectos de aprendizagem suportada pela plataforma *Joomla!*, nomeadamente no que diz respeito à criação da estrutura e à publicação de conteúdos educativos na respectiva base de dados. Finalmente, houve também a necessidade de desenvolver uma página educativa de um professor usando as tecnologias para a Web Semântica.

De destacar também o uso das linguagens de programação PHP (PHP, 2005) e do sistema de gestão de base de dados MySQL (MySQL, 2005a), quer na preparação dos sistemas remotos (sistemas de e-Learning, sistemas de gestão de conteúdos e Website do professor) para a Web Semântica, quer na implementação de alguns componentes do protótipo do sistema WSE (BD do sistema e utilitários).

Uma vez que o Java foi a linguagem de programação escolhida para o desenvolvimento do servidor de agentes, a solução mais natural seria recorrer a um ODBMS (*Object Oriented Database Management System*) para a BD do sistema. No entanto, os últimos ciclos do processo de desenvolvimento apontavam para a necessidade de que o servidor de agentes se comportasse como um sistema de e-Learning Moodle (que usa base de dados MySQL) integrando diversos sistemas de e-Learning Moodle distribuídos por diversas escolas secundárias. Por conseguinte, optou-se pelo MySQL usando uma ligação JDBC (*Java DataBase Connectivity*), uma vez que para este caso de estudo as vantagens oferecidas pelas bases de dados orientadas aos objectos não são assim tão significativas. JDBC não é mais do que uma API composta por um conjunto de classes e interfaces escritas em Java para enviar as instruções SQL para uma base de dados relacional. O *driver* JDBC usado neste protótipo foi o *Connector/J* (MySQL, 2005b) que é exclusivamente em Java e implementa a especificação 3.0 do JDBC, comunicando directamente com o servidor MySQL através do protocolo respectivo (MySQL, 2006).

Finalmente, não podemos deixar de referir também a utilização das linguagens ou tecnologias para o e-Learning (SCORM/IMS/IEEE-LOM/DCM) e para a Web Semântica (RDF/RDFS/OWL) devidamente explanadas nos capítulos II, III e IV.

7.2- Protótipo do Sistema WSE

O protótipo do sistema WSE para a pesquisa de objectos de aprendizagem baseado nas tecnologias para a Web Semântica, nas tecnologias para o e-Learning e nas tecnologias para os agentes é composto pelos seguintes componentes:

- Componente "agente de interface", que permite disponibilizar os serviços necessários a formadores e formandos, nomeadamente a interface para solicitar objectos de aprendizagem e disciplinas ou cursos de formação existentes em sistemas remotos registados no sistema WSE;
- Componente "servidor de agentes", que constitui o núcleo do sistema WSE e que torna transparente todo o processo de pesquisa aos utilizadores, enviando e gerindo os agentes de pesquisa móveis de acordo com as solicitações dos utilizadores;
- Componente "agente de pesquisa", que permite melhor desempenho na pesquisa ao evitar que os agentes móveis permaneçam no Servidor para o qual se deslocam, seguindo o seu percurso enquanto o agente de informação realiza a inferência;
- Componente "agente de informação", que permite a pesquisa de informação de acordo com o tipo de sistema a pesquisar (Sistema de e-Learning, repositório de aprendizagem ou Website educativo);
- Componente "utilitários para metadados, ontologias e mapas de tópicos", que permite complementar a anotação semântica de sistemas remotos registados e editar ontologias e mapas de tópicos com vista a orientar os agentes na pesquisa de conteúdos educativos dispersos e a apoiar a navegação semântica.

Tal como especificado no capítulo VI, o servidor de agentes possui diversos objectos servidores que disponibilizam serviços. A localização e utilização destes serviços por parte

dos restantes componentes do sistema são efectivadas usando o serviço de nomes disponibilizado pela infra-estrutura Voyager através de invocações síncronas. No entanto, para fornecer a funcionalidade e a flexibilidade que um sistema distribuído deste tipo necessita, as mensagens do tipo síncrono não são suficientes. Portanto, a comunicação entre os diversos componentes do sistema exige também o recurso ao *multicast* de eventos ou mensagens e às mensagens *one-way*.

A arquitectura Voyager permite que as aplicações sejam tanto cliente como servidor, suportando comunicação com outras aplicações Voyager, CORBA, SOAP, DCOM e RMI. Portanto, são disponibilizadas diversas implementações do serviço de nomes: *CORBA naming service*, *CORBA IOR resolution*, *JNDI naming service*, *RMI registry naming service*, *Microsoft active directory* e *Voyager federated directory service*. Este último foi o serviço de nomes e de directório usado no âmbito do protótipo que aqui se apresenta.

Este serviço de nomes permite a resolução de nomes em objectos, permitindo localizar os objectos remotos através do seu nome. Desta forma, um objecto remoto é representado por um *proxy* desse objecto que implementa as mesmas interfaces que as do seu homónimo remoto, permitindo que os objectos clientes usem os serviços disponibilizados pelos objectos servidores.

Tanto na comunicação dos agentes de interface e dos agentes de informação com o servidor de agentes (comunicação remota), como na comunicação dos agentes de pesquisa móveis com os agentes de informação e com o servidor de agentes (comunicação local), a disponibilização das funcionalidades é feita pelo serviço de nomes e de directório referido. Por conseguinte, a infra-estrutura Voyager terá de ser instalada e configurada não só na máquina do servidor de agentes, mas também em todas as máquinas onde são executados os agentes de informação.

Assim, quando o servidor de agentes ou os agentes de informação (ou outra aplicação Voyager) são executados, exportam automaticamente um objecto “Directory” que será usado pela classe “Namespace” para o registo e localização de objectos servidores, através das primitivas “rebind” e “lookup”, respectivamente.

Tal como vimos no capítulo anterior, o servidor de agentes cria o objecto “ClientServ” para os agentes de interface, o objecto “AgentMServ” para os agentes de pesquisa móveis, e o “AgentLOServ” para os agentes de informação. Por sua vez, os agentes de informação criam o objecto “AMPServ” que disponibiliza serviços aos agentes de pesquisa móveis que chegam

à máquina onde esse agente de informação está em execução. Aquando da sua criação, cada objecto é registado no serviço de nomes e de directório e é-lhe associado um determinado nome. As linhas de código Java abaixo, inerentes ao servidor de agentes, exemplificam a criação do *proxy* "iamps" relativo ao objecto "servidorwse.AgentMServ", registado no servidor de nomes através da primitiva "Namespace.rebind" com o nome "MyAgentMServer", para que os agentes de pesquisa móveis através da interface "IAgentMServ" possam localizar e aceder aos serviços deste objecto.

```
// Construir o serviço para os Agentes Móveis de Pesquisa
iamps = (IAgentMServ) Factory.create("servidorwse.AgentMServ");
Namespace.rebind("MyAgentMServer", iamps);
System.out.println("Serviço de Agentes Moveis de Pesquisa criado...");
```

Do lado dos agentes de informação, as linhas de código seguintes exemplificam a criação dos *proxies* "ampsbd" e "ampsrdf" do objecto "loserver.AMPServ" no agente de informação, registados no directório local através da primitiva "Namespace.rebind" com o nome "AMPServerbd" e "AMPServerrdf", para que os agentes de pesquisa móveis, através da interface "IAMPServbd" ou "IAMPServrdf", possam localizar e aceder aos serviços desse agente de informação.

```
// construir o serviço para receber os Agentes Móveis de Pesquisa
ampsbd = (IAMPServbd) Factory.create("loserver.AMPServ",url+":9001");
Namespace.rebind( "AMPServerbd", ampsbd );
System.out.println("Serviço SQL BD para Agentes de Pesquisa criado...");
...
ampsrdf = (IAMPServrdf) Factory.create("loserver.AMPServ",url+":9001");
Namespace.rebind( "AMPServerrdf", ampsrdf );
System.out.println("Serviço SPARQL para Agentes de Pesquisa criado...");
```

O *proxy* "iamps" representa uma instância local, enquanto que os proxies "ampsbd" e "ampsrdf" representam instâncias remotas.

A localização dos serviços disponibilizados pelos agentes de informação é feita na máquina para a qual migra um determinado agente de pesquisa móvel, através da primitiva "Namespace.lookup". O parâmetro desta primitiva é construído com base no percurso do agente (itinerário que o agente transporta consigo) e nos nomes de registo dos serviços disponibilizados pelo agente de informação ("AMPServerbd" e "AMPServerrdf"), tal como se pode observar no método relativo ao agente de pesquisa móvel (classe "AgentMP"):

```
private void invocar() {

...

//Se a pesquisa é básica ou o parâmetro = pesquisa SQL na BD
IAMPServbd servicobd =
(IAMPServbd)Namespace.lookup((String)percurso_.elementAt(local_)+"/AMPServerbd");

...

//Caso contrário, a pesquisa é realizada com o motor de inferência SPARQL
IAMPServrdf servicordf =
(IAMPServrdf)Namespace.lookup((String)percurso_.elementAt(local_)+"/AMPServerrdf");
}
```

Desta forma, o agente de pesquisa móvel obtém a interface da aplicação remota para poder aceder aos métodos da aplicação que se encontram na máquina remota, de acordo com os parâmetros que transporta consigo. Após ter cumprido o seu percurso, esse mesmo agente de pesquisa terá de localizar o serviço registado com o nome "MyAgentMPServer" que permite informar o servidor de agentes do seu regresso, tal como se pode observar no método abaixo, inerente ao agente de pesquisa móvel (classe "AgentMP"):

```
public void regresso() {
...

ams = (IAgentMServ)Namespace.lookup(servidorwse+"/MyAgentMServer");

...
}
```

Para além do acesso a serviços remotos, o servidor de agentes e os agentes de informação têm necessidade de recorrer ao *multicast* de mensagens para determinados componentes do sistema. Mas o aumento de objectos destinatários dessas mensagens pode afectar a eficiência do serviço. Para optimizar o processo de replicação de mensagens, a confiabilidade das mensagens e a largura de banda da rede, a Voyager disponibiliza uma arquitectura de sub-espaços (*Voyager Subspaces*). Assumindo que o espaço distribuído corresponde ao conjunto de máquinas virtuais e que o sub-espaço contém apenas uma única máquina virtual, então o espaço do sistema WSE é um conjunto de sub-espaços composto pela máquina do servidor de agentes e todas as máquinas nas quais foram instalados e configurados agentes de informação.

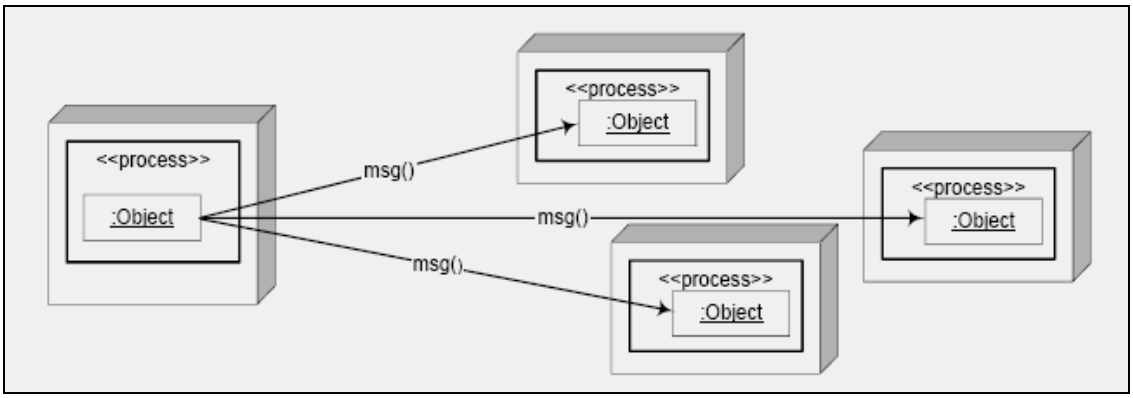


Figura 99 – *Multicasting* de mensagens na plataforma Voyager

Uma mensagem ou um evento enviado através de um *proxy multicast* para um sub-espaço é copiado para todos os sub-espaços vizinhos antes de ser entregue aos objectos do sub-espaço local, o que permite uma distribuição simultânea da mensagem ou do evento a todos os objectos existentes no espaço.

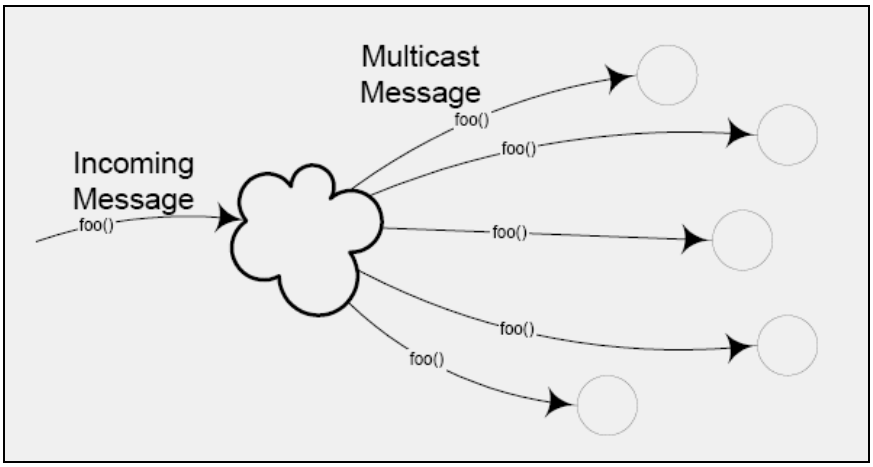


Figura 100 – *Multicasting* de mensagens para um sub-espaço Voyager

Na última versão da Voyager, basta usar "Subspace subspace = new Subspace();" para criar o sub-espaço. Para adicionar membros locais ao sub-espaço usa-se "subspace.add(new String());", "subspace.add(new Vector());" ou "subspace.add(*Hello*());", enquanto que para adicionar membros remotos é necessário localizar previamente o objecto em questão (por exemplo: *Object hello = Namespace.lookup("9000/HelloWorld");*).

A criação do sub-espaço do servidor de agentes do sistema WSE e o respectivo registo no serviço de directório local com o nome "SubSpaceWSE" pode ser observado nas linhas de código abaixo:

```
//Construir o Subspace
sswse = (ISubspace) Factory.create("com.recursionsw.ve.space.Subspace");
Namespace.rebind("SubSpaceWSE", sswse);
System.out.println("Subspace criado...");
```

No caso dos sub-espaços intrínsecos aos agentes de informação, o processo é idêntico. Após criar esses sub-espaços, torna-se necessário localizar o sub-espaço do servidor de agentes para interligar os dois sub-espaços através da primitiva "connect":

```
//Construir o Subspace do agente de informação
slo = (ISubspace) Factory.create("com.recursionsw.ve.space.Subspace");
Namespace.rebind("SubSpaceLO", slo);
System.out.println("Subspace LO criado...");

//Lookup do subspace do servidor de agentes
ISubspace sswse = (ISubspace)Namespace.lookup(servidorwse+"/SubSpaceWSE");

sswse.connect(slo);
```

As ligações do sub-espaço do servidor de agentes com cada um dos sub-espaços dos agentes de informação resultam num espaço de máquinas virtuais com topologia em estrela.

7.2.1- Agente de Interface

No âmbito da arquitectura foram identificados três perfis de utilizadores (formando, formador e administrador de sistema remoto), para além do gestor do sistema WSE e do perfil de visitante ou convidado (*guest*). Os acessos destes utilizadores são efectuados através de clientes Web. Todavia, o gestor do sistema WSE tem ainda a possibilidade de aceder localmente ao servidor de agentes e cada um dos administradores de sistemas remotos pode aceder localmente ao agente de informação correspondente para ligar ou desligar o serviço.

Tal como referimos no capítulo anterior, o agente de interface é carregado no *browser* a partir do servidor onde se encontra em execução o servidor de agentes móveis. Estes agentes de interface correspondem a *applets* Java.

Uma vez que apenas é possível abrir uma ligação para o mesmo servidor Web, o acesso à arquitectura distribuída terá que ser assegurado pela activação do servidor Voyager

via HTTP através do servidor Web. Face às questões de segurança das *applets*, podemos usar o servidor de agentes como um *hub* ou concentrador de objectos da nossa aplicação.

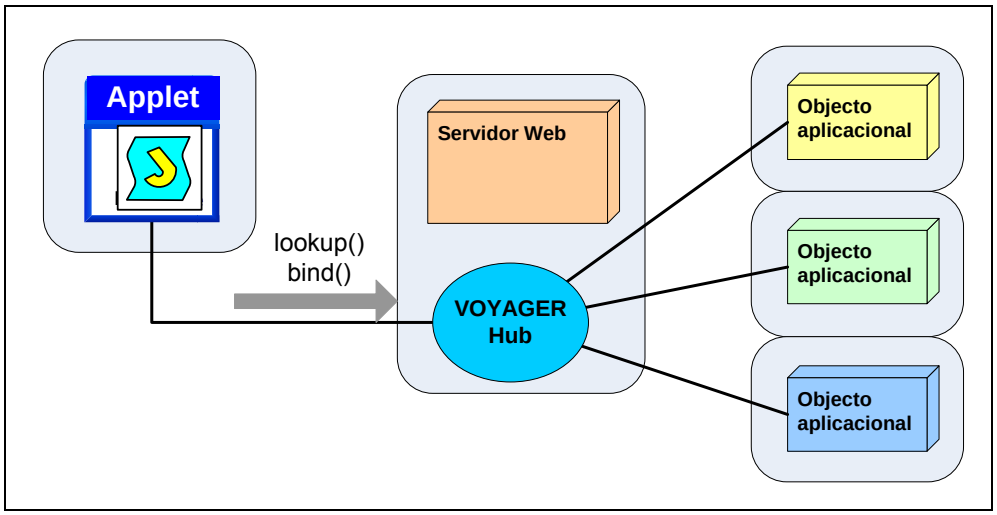


Figura 101 – Acesso de uma *applet* à arquitectura distribuída do servidor de agentes

Desta forma, asseguramos que a máquina do cliente não necessite de instalar e configurar a plataforma Voyager para aceder às funcionalidades do servidor de agentes.

Quando o utilizador acede ao serviço utilizando o *browser* Web, a *applet* embebida numa página HTML é responsável primeiramente por autenticar o utilizador. Caso o perfil do utilizador seja incorrecto, o utilizador é convidado a seguir uma de quatro opções: 1) Introduzir um perfil correcto; 2) Usar o perfil de utilizador visitante ou convidado (*guest*); 3) Solicitar a inscrição como formando ou formador; 4) Abandonar o acesso ao sistema através do botão "Cancelar".

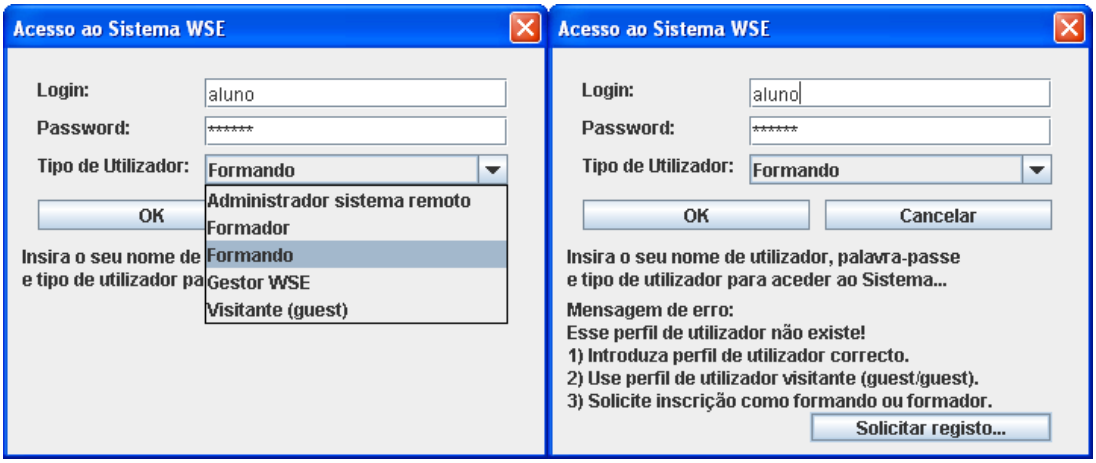


Figura 102 – Autenticação de um utilizador

Caso contrário, o utilizador é redireccionado de acordo com o seu perfil para o serviço correspondente. Para tal, a *applet* é responsável pela obtenção do *proxy* do objecto que lhe disponibilizará as funcionalidades que requisitar. Tal como vimos anteriormente, este *proxy* é obtido através do serviço de nomes e de directórios usando a primitiva *lookup*. A linha de código abaixo pertence à *applet* destinada aos formandos e permite disponibilizar as funcionalidades do serviço "MyClientAServer".

```
aluno =
(IAlunoServ) Namespace.lookup(Routing.getRouterAddress() + "/MyClientAServer");
```

As interfaces gráficas de pesquisa permitem que os utilizadores indiquem que tipo de LOs e de disciplinas ou cursos de formação desejam procurar. Esta camada de interface para consultas disponibiliza interfaces de pesquisa básica e interfaces de pesquisa avançada. Tanto na pesquisa de LOs como na pesquisa de disciplinas ou cursos de formação, as interfaces de pesquisa básica permitem especificar os temas ou assuntos (palavras-chave) e o tipo de sistemas remotos a pesquisar (sistemas de e-Learning, repositórios de objectos de aprendizagem, sistemas de gestão de conteúdos e outros servidores Web educativos). Caso a pesquisa incida sobre LOs é possível também indicar o tipo de LOs a pesquisar: apresentação (ppt, pdf, etc); manual, sebenta ou documento similar (html, pdf, ppt, doc, etc); lição; pacote (SCORM ou IMS-CP); ficha de trabalho; avaliação (formativa, sumativa, etc); questionário, inquérito ou sondagem; fórum; wiki; blog; entre outros. Caso a pesquisa incida sobre as disciplinas ou cursos de formação é possível indicar também o nível de aprendizagem (indefinido; nível 0 – curso completo; nível 1 – iniciação; nível 2 – intermédio; nível 3 – avançado; todos os níveis). As interfaces gráficas de pesquisa avançada permitem refinar esses parâmetros de pesquisa e acrescentar outros em conformidade com a especificação de metadados e o tipo de pesquisa (disciplinas ou apenas LOs), possibilitando indicar o nível semântico sobre o qual decorrerá a pesquisa (pesquisa baseada nos metadados armazenados na base de dados do sistema remoto ou pesquisa baseada nos documentos RDF/XML, semanticamente mais expressivos, armazenados num directório do sistema remoto). Como referimos anteriormente, a pesquisa básica fornece um primeiro nível de metadados existente no próprio sistema WSE e referente à anotação semântica dos sistemas remotos feita pelo gestor da WSE e complementada pelos formadores ou administradores dos sistemas remotos, enquanto que a pesquisa avançada permite adicionar mais elementos de filtragem relativos a este primeiro nível e especificar um segundo nível de metadados referente aos elementos de

metadados existentes nos sistemas remotos. Na Figura 103 podemos observar a interface de pesquisa básica, bem como os menus disponíveis para um utilizador com perfil de formando.

Figura 103 – Agente de interface: pesquisa básica

De referir que, quando um utilizador se desliga do serviço através da opção “Sair” do menu “Servidor”, é-lhe apresentada uma janela de confirmação que o avisa de que está a abandonar a aplicação e, como tal, todos os agentes de pesquisa que ainda não regressaram serão destruídos. O utilizador poderá confirmar ou não a saída. Para evitar esta situação, o utilizador pode usar a opção Logout do menu “Formando”, mantendo os agentes de pesquisa em circulação para que, quando volte a aceder ao serviço, tenha ao seu dispor os resultados das pesquisas que submeteu. Esta característica só por si justifica a necessidade de registo dos utilizadores.

A interface do formador e do gestor LOs (administrador de sistemas remotos) é idêntica à do formando. No entanto, o menu de opções do formador e do gestor LOs disponibiliza a anotação semântica e a submissão de planos de cursos no formato XTM, para além de ligações a módulos experimentais de edição de ontologias e mapas de tópicos.

7.2.2- Servidor de Agentes

As classes possuem um propósito específico e normalmente relacionam-se com outras classes, formando assim um sistema completo e funcional. Ao codificar uma classe, podemos separar os seus métodos em dois tipos: os de acesso público e os de acesso privado. Os métodos privados realizam tarefas ou processam informação que diz respeito ao funcionamento interno da classe em questão. Mas, se um método é declarado público, então pode ser invocado por outros objectos.

Quando um objecto se relaciona com outro, ele não acede aos seus recursos ou funcionalidades directamente, mas sim através da sua interface. Com as interfaces podemos definir o que é feito, sem termos que definir como é que vai ser feito.

Uma classe pode implementar várias interfaces. Por exemplo: a classe "ClientServ" implementa as interfaces "IAlunoServ", "IProfServ" e "IAdminseServ". Por sua vez, uma interface também pode ser implementada por várias classes. É o caso das interfaces "IProfServ" e "IAdminseServ" para as quais foi necessário especificar uma outra classe "AnotaServ" para implementar as funcionalidades inerentes à anotação semântica, para além da classe "ClientServ" que implementa as funcionalidades de comunicação entre o agente de interface e o servidor de agentes.

Como vimos anteriormente, esse conjunto de funcionalidades reside no servidor de agentes. Este componente não é mais do que uma aplicação de Java que permite gerir todo o sistema, desde os clientes Web, aos agentes de informação, passando pelos agentes de pesquisa móveis.

A administração do servidor de agentes fica a cargo do gestor do sistema WSE (designado na interface por Administrador) através das funcionalidades fornecidas pelo objecto "GestorServ".

As principais funcionalidades deste componente permitem a gestão de utilizadores e de sistemas externos ou remotos que aderem ao sistema. Quando os utilizadores solicitam o registo no sistema WSE, o gestor terá que validar o perfil que solicitam e activar o registo, a menos que a autenticação dos utilizadores seja efectuada pelo servidor LDAP do IPB. Esta funcionalidade permite ler os atributos do utilizador no LDAP e preencher os valores pedidos pelo sistema WSE (na versão actual do protótipo apenas o tipo de utilizador e o nome de utilizador e respectiva palavra-passe são registados). Neste caso, se o nome de utilizador e palavra-passe

forem válidos, o sistema WSE cria um novo registo de utilizador na DB do sistema. Nas seguintes vezes em que o utilizador entrar, só será verificado o nome de utilizador e correspondente palavra-passe. A gestão de utilizadores é apresentada na Figura 104.

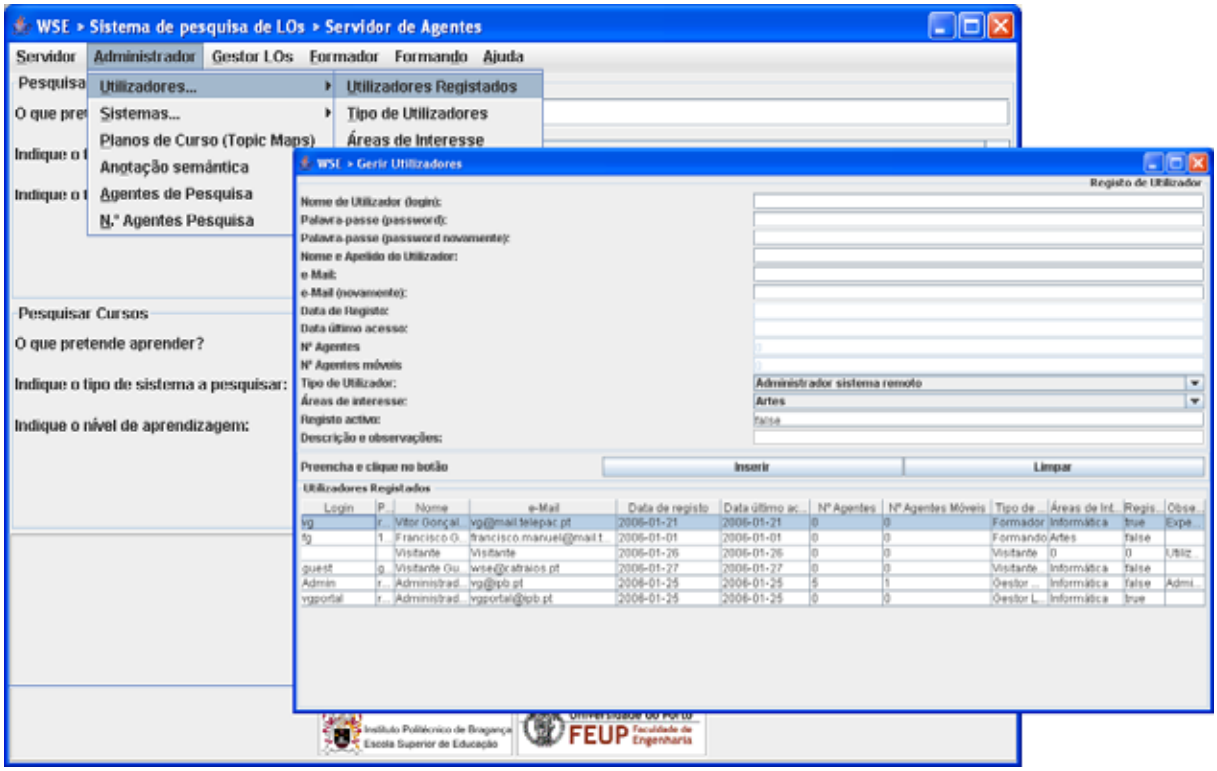


Figura 104 – Servidor de agentes: gestão de utilizadores

A actualização do registo de um utilizador permite definir o número máximo de agentes móveis permitido para esse utilizador. Este valor é definido pelo gestor do sistema e depende essencialmente do número médio de utilizadores activos, já que tem como principal objectivo optimizar o processo de pesquisa de objectos de aprendizagem. Para uma rápida actualização desse valor, o menu "Administrador" disponibiliza uma opção que permite o acesso a uma tabela editável que lista o número máximo de agentes e número de agentes de pesquisa em circulação por utilizador.

A eliminação de um utilizador implica, para além da remoção do registo desse utilizador, a remoção de todos os registos da tabela de agentes e da tabela de LOs referentes a eventuais pesquisas submetidas por esse utilizador.

Ao efectuar o registo no sistema WSE, um administrador de um sistema candidato pode submeter o pedido de adesão do sistema juntamente com o seu registo de utilizador. Com base nessas informações o gestor WSE procede ao seu registo na tabela de sistemas. Para cada sistema registado será necessário o URL ou o endereço IP da máquina, o tipo de

sistema e o tipo de LOs que disponibiliza, o nome e tipo de BD local e/ou a localização da pasta de metadados, bem como as formas de interagir com esses contentores de informação. Estas informações podem ser actualizadas à posteriori pelo administrador do agente de informação.

A tabela de sistemas registados permite ainda a actualização automática do estado do agente de informação correspondente, de acordo com as mensagens que chegam ao servidor de agentes vindas desse agente de informação. A este agente de informação podem ser associados metadados através da anotação semântica disponível tanto para o administrador do sistema WSE (Administrador), como para o administrador do sistema remoto (o menu é representado por Gestor LOs na interface), tal como se pode verificar na Figura 105.

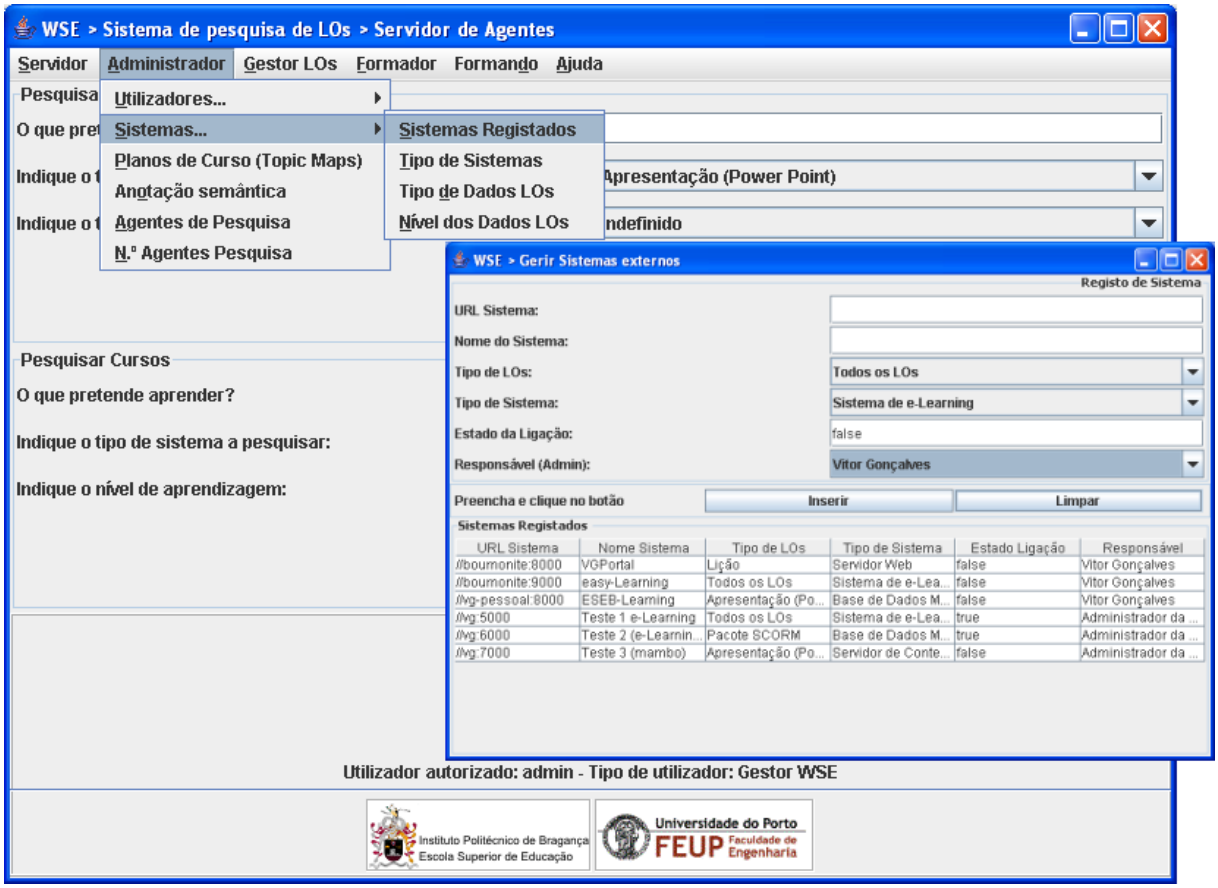


Figura 105 – Servidor de agentes: gestão de sistemas externos

Para suportar estas duas funcionalidades, o gestor do sistema WSE poderá ter necessidade de actualizar informações relativas ao tipo de utilizadores e às suas áreas de interesse, ao tipo de sistemas, ao tipo de dados dos LOs ou ao nível de aprendizagem. Informações estas que são úteis para a localização de informação uma vez que é com elas que

o utilizador define os principais parâmetros para solicitar a criação de agentes móveis de pesquisa.

Uma outra funcionalidade disponibilizada pelo servidor de agentes é a anotação semântica de sistemas. Este componente tem como finalidade complementar as informações da tabela de sistemas registados, criando uma camada de informação através da adição de metadados aos locais onde residem os recursos educativos. Esta funcionalidade será abordada no âmbito do componente "utilitários para metadados, ontologias e mapas de tópicos".

Com o objectivo de monitorizar o estado dos agentes de pesquisa e o seu ciclo de vida, este menu disponibiliza também uma listagem dos agentes de pesquisa por utilizador permitindo averiguar se os agentes estão em migração ou já regressaram ao sistema. Os pormenores relativos ao ciclo de vida dos agentes de pesquisa móveis podem ser encontrados na secção que se segue.

7.2.3- Agente de Pesquisa

Um agente de pesquisa não é mais do que um objecto de software situado num ambiente de execução (sistema WSE baseado na plataforma Voyager). Os agentes podem mover-se autonomamente entre as aplicações ou componentes e continuam a sua execução quando chegam a um novo local. Genericamente, a Voyager fornece-lhe a autonomia e mobilidade necessárias para a execução das tarefas especificadas.

A autonomia é-lhe atribuída por omissão aquando da sua criação. Para criar o agente de pesquisa baseado no objecto *agent facet* usa-se o método *AgentFacet.of()*. Para terminar a sua execução, é necessário que a autonomia seja configurada como falsa através do método *setAutonomous(false)*. Assim, o agente fica apto a ser destruído.

A mobilidade é-lhe atribuída com um dos seguintes métodos:

- *moveTo(object)*;
- *moveTo(url)*.

Quanto à inteligência dos agentes, esse atributo só poderá ser configurado pelo programador. O grau de inteligência dependerá da base de conhecimento (metadados e ontologias) e do poder de inferência que ele lhe conseguir atribuir.

O ciclo de vida de um agente de pesquisa pode ser resumido a três fases: criação do agente de pesquisa; execução do agente de pesquisa; e destruição do agente de pesquisa. A Figura 106 ilustra de forma geral estas fases.

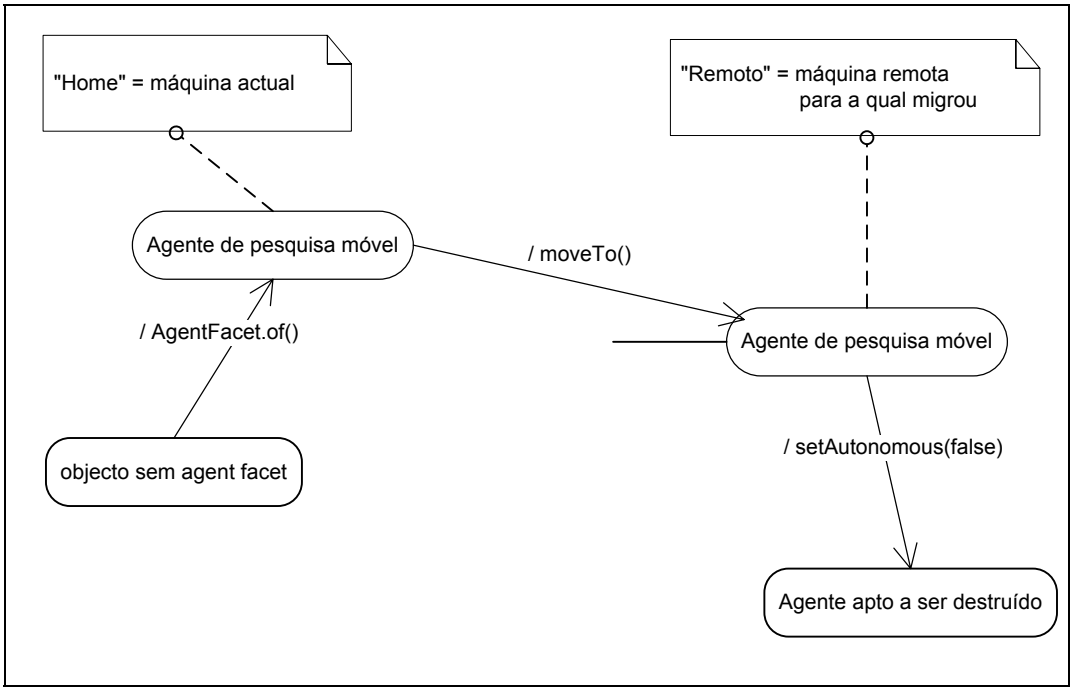


Figura 106 – Ciclo de vida genérico de um agente de pesquisa

Quando os utilizadores especificam as palavras-chave, o tipo de sistemas a pesquisar e outros parâmetros relativos ao tipo de objectos de aprendizagem ou ao tipo de cursos de formação que desejam procurar, o agente de interface correspondente invoca o método "registaragentelos" ou o método "registaragentecursos" do objecto "ClientServ". Este método solicita a criação do agente de pesquisa e a BD do sistema é actualizada com vista a posteriormente identificar o seu dono e a informação que eventualmente chegue ao sistema. No entanto, tal como referido anteriormente, caso exista um agente de pesquisa em migração análogo ao especificado, o agente não é criado e o utilizador é notificado através de uma janela de aviso.

A **criação do agente de pesquisa** no âmbito do servidor de agentes traduz-se na criação de uma instância do objecto "AgentMP" e na invocação do método que lhe fornece autonomia e mobilidade. Para que o agente saiba para onde deve migrar, é-lhe entregue o seu percurso. O percurso corresponde a um "vector", cujos elementos representam os endereços das máquinas que deve visitar, construído com base nos parâmetros de pesquisa entregues pelo agente de interface do utilizador, nas anotações semânticas dos sistemas remotos

registado no sistema e no estado dos agentes de informação (só agentes de informação ligados ou em execução é que constam do percurso do agente).

A **execução do agente de pesquisa** manifesta-se após obter a *facet* de agente autónomo através do método "of" da classe "AgentFacet" e a mobilidade através do método "moveTo", definido na interface "IAgentFacet". Refira-se que, nas versões anteriores à Voyager Edge, a classe "AgentFacet" e a respectiva interface "IAgentFacet" designavam-se simplesmente "Agent" e "IAgent", respectivamente.

O método "of" adiciona ao objecto "AgentMP" a *facet* que, após ser obtida, permite invocar o método "moveTo" com os parâmetros referentes ao endereço da máquina virtual para a qual o agente de pesquisa se deve mover e o nome do método a ser invocado quando chegar ao destino:

```
//Método invocado pelo servidor para lhe fornecer autonomia e mobilidade e
entregar-lhe o seu percurso
public void local(Vector v) {
. . .
try {
AgentFacet.of(this).moveTo((“tcp:”+(String)percurso_.elementAt(local_)),“remoto”);
    } catch( Exception exception) {
    . . .
    }
}
```

Chegado ao seu primeiro destino, o passo seguinte é identificar-se e entregar o pedido ao agente de informação. E fica pronto para migrar para a próxima máquina remota. Mas, antes de migrar, deverá verificar se a próxima máquina é o servidor de onde partiu:

```
public void remoto() {
   invocar(); // actividades a efectuar na máquina para onde se deslocou
   local_++; //incrementa a variavel para saber para onde ir a seguir
   //se o próximo destino é a máquina do servidor de onde partiu
   if( ((String)percurso_.elementAt(local_)).equals(AgentFacet.of(this).getHome()))
        { try {
        AgentFacet.of( this ).moveTo( (String)itenerario_.elementAt(local_),
"regresso" ); //migra para o servidor e executa o método "regresso"
        }catch( Exception exception ) {System.err.println( exception );}
        } else //Caso contrário, o próximo destino não é o servidor
        {
        try { //migra para o novo destino e executa novamente este método("remoto")
AgentFacet.of(this).moveTo(("tcp:"+(String)percurso_.elementAt(local_)),"remoto");
            }catch(Exception exception) {System.err.println(exception);}
        }
    }
```

Quando o agente está numa máquina remota, executa um conjunto de tarefas definidas no método "invocar()". Para tal, obtém a interface da aplicação remota para poder aceder ao método da aplicação que se encontra na máquina remota. O método "invoke" da classe

"OneWay" permite que o agente de informação procure informação sobre os objectos de aprendizagem na sua BD local ou nos seus documentos RDF/OWL, de acordo com o serviço solicitado, enquanto o agente de pesquisa segue o seu percurso:

```
//Método que fornece as actividades a serem efectuadas no RDF da máquina remota
//Para as actividades (servicobd) a serem realizadas na BD o código é análogo
public void invocar()
{

try {
IAMPServrdf servicordf =
(IAMPServrdf)Namespace.lookup((String)percurso_.elementAt(local_)+"/AMPServerrdf");
     boolean ret = servicordf.identAg(ident_);
     if(!ret)
      {
     OneWay.invoke(servicordf,"pesqrdf",newObject[]{pesq_, new Integer(ident_)});
     Thread.sleep(1000);
      }  else {
  terminar(); }
  } catch( Exception e) {
  . . .
 }
. . .
}
```

Para além dos métodos que permitem concretizar a migração do agente de pesquisa, a interface "IAgentFacet" também define os métodos para a sua destruição. Uma vez que este sistema se baseia na linguagem de programação Java (JAVA, 2004), para que um agente seja destruído é necessário retirar-lhe a autonomia. Para tal, é indispensável fazer com que seja reclamado pelo *garbage collector*. Assim, **a destruição de um agente de pesquisa** resulta da invocação do método "setAutonomous", que provoca a conclusão da sua execução ao forçar que seja reclamado pelo *garbage collector*. O método responsável por permitir que este mecanismo reclame o objecto AgentMP apresenta-se no código seguinte:

```
private void terminar()
   {
       Agent.of(this).setAutonomous(false);
   }
```

7.2.4- Agente de Informação

Os agentes de informação são aplicações desenvolvidas em Java que permitem que os agentes de pesquisa móveis solicitem a pesquisa de objectos de aprendizagem em bases de dados e/ou em documentos RDFS/OWL.

Os administradores dos sistemas remotos podem ligar ou desligar o serviço Voyager inerente ao agente de informação, utilizando a aplicação instalada no servidor correspondente. Eventuais alterações nas configurações da BD local e da pasta de metadados devem ser realizadas através do menu "Gestor LOs" do servidor de agentes, tal como a anotação semântica dos sistemas remotos registados no sistema WSE.

Estas funcionalidades relativas aos sistemas remotos são invocadas do servidor de agentes pelos administradores desses sistemas usando os agentes de interface. A anotação semântica de sistemas remotos registados no sistema WSE através de metadados e ontologias permite optimizar o mecanismo de localização, filtragem e recuperação de informação. Ou seja, o percurso a criar para o agente de pesquisa será menor e a informação que o mesmo transporta sobre o alvo da pesquisa será mais pormenorizada.

Para além de receber os agentes de pesquisa e atender os seus pedidos, ao agente de informação compete a actualização constante do seu estado de execução e, com base no estado do servidor de agentes, a destruição de eventuais agentes de pesquisa que entretanto se lhe apresentem.

Um agente de informação é a interface entre o agente de pesquisa móvel e a BD do sistema remoto (por exemplo, a BD *moodledata* de um sistema de e-Learning Moodle) ou a pasta de metadados (por exemplo, */raiz_moodle/moodledata/local_metadados*).

Por conseguinte, os parâmetros submetidos através do agente de interface, que permitiram definir o percurso do agente de pesquisa móvel e que o acompanharam até aos seus destinos, são os dados de entrada das classes responsáveis por realizar a pesquisa tanto na BD local como na pasta de metadados local do agente de informação.

Genericamente, os parâmetros recebidos são interligados com os campos correspondentes da(s) tabela(s) de metadados na BD local. Com base nesse alinhamento, a pesquisa é realizada através de SQL pelo método responsável pela recuperação de metadados dos LOs.

Uma pequena parte do código relativo à pesquisa de LOs baseados nos metadados armazenados na BD local do sistema remoto é apresentada abaixo para exemplificar o processo:

```
public Array pesqbd(Array parpesq) {
    Array result = null;
    ResultSet resultSet = null;

...

    try { //estabelecer ligação à BD local
       jdbc=new MyDBLOCon();
       jdbc.init();
       Connection conn=jdbc.getMyConnection();
       stmt= conn.createStatement();

       String query = "SELECT title, identifier, description, relation FROM
       metadatadc WHERE type=\"" + tipodados + "\" AND title =\"" + pesq + "\" AND
       subject =\"" + pesq + "\" ... "\"ORDER BY title";

       resultSet = stmt.executeQuery(query);

      ...

       stmt.close();
       }
    } catch ( SQLException sqlex ) {
            System.out.println("Erro ao procurar LOs na BD!");
            sqlex.printStackTrace();
  }
...
```

No caso da pesquisa baseada nos documentos RDF, a solução encontrada baseia-se na *framework* Jena (*HP Labs Bristol*). Tal como referimos anteriormente, esta infra-estrutura de Java *open source* fornece um ambiente de programação para aplicações destinadas à WS, nomeadamente RDF, RDFS e OWL, para além da linguagem SPARQL e um motor de inferência baseado em regras. No âmbito desta parte do desenvolvimento, vamos focar a atenção nestes últimos componentes que permitem consultar ou inferir documentos RDF(S)/OWL. Para executar consultas em SPARQL e RDQL, Jena disponibiliza o motor de consultas ARQ.

O primeiro passo é criar um modelo para manter os ficheiros de metadados na memória. A consulta ou pergunta é criada a partir da *string* “querystring” utilizando o método “QueryFactory”.

As linhas de código seguintes representam a parte mais relevante do motor de consulta “servicordf”:

```
  import com.hp.hpl.jena.query.*; //Biblioteca ARQ API
  String[] metadados; //array de nomes dos ficheiros de metadados
  //Criar modelo na memória e preenchê-lo com metadados
  Model model = ModelFactory.createMemModelMaker().createModel();
  for (int i = 0; i < metadados.length; i++) {
  model.read(metadados[i], null);
  }
...
//Questionar ou consultar o modelo de metadados
  String queryString = "
    PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
    SELECT ?x
    WHERE { ?x rdfs:label ?v .
    FILTER regex(?v, '.$pesquisa.', "i") } "; //Filtro: parâmetros de pesquisa
  Query query = QueryFactory.create(queryString); //executar consulta
  QueryExecution qexec = QueryExecutionFactory.create(query, model);
  try {
    ResultSet results = qexec.execSelect();
    for ( ; results.hasNext() ; ) {
  ...
  } finally { qexec.close() ; }
```

A *framework* Jena suporta MySQL e o driver JDBC (MySQL Connector/J). Por conseguinte, substituir o serviço "servicobd" por outro com base nesta tecnologia será uma funcionalidade a implementar num futuro próximo, desde que a infra-estrutura distribuída atinja a maturidade suficiente e garanta que a informação actualmente armazenada nas BD dos agentes de informação seja totalmente representada no formato RDF/OWL.

De referir que as listagens de código foram apresentadas de forma simplificada para atenuar a complexidade dos processos descritos.

Podemos afirmar que os agentes de informação são aplicações executadas em *background* nas diversas máquinas existentes na rede e aderentes ao sistema WSE. Para tal, o administrador de cada um dos sistemas externos apenas terá que executar uma *batch file* (ficheiro com extensão bat) ou adicioná-la aos serviços do servidor. No caso do agente de informação subjacente ao sistema easy-Learning da ESE-IPB o ficheiro "easy.bat" corresponde ao seguinte comando:

```
java -Dclass_based_proxies=true loserver.ServidorLO //easy-learning.ipb.pt:9001
```

A execução deste comando e, consequentemente, do programa java, permitirá ligar o servidor Voyager para activar esse agente de informação. O extracto de código responsável pela ligação do servidor Voyager deste agente de informação corresponde às seguintes linhas:

```
String host = (InetAddress.getLocalHost()).getHostName();
url= "//"+ host;
ClassManager.enableResourceServer();
Voyager.startup( url+":9001" );  // inicia Voyager na url:port 9001
System.out.println("Voyager ligado: "+url+":9001");
```

Com este componente o ambiente distribuído fica completo, tal como se pode verificar na Figura 107.

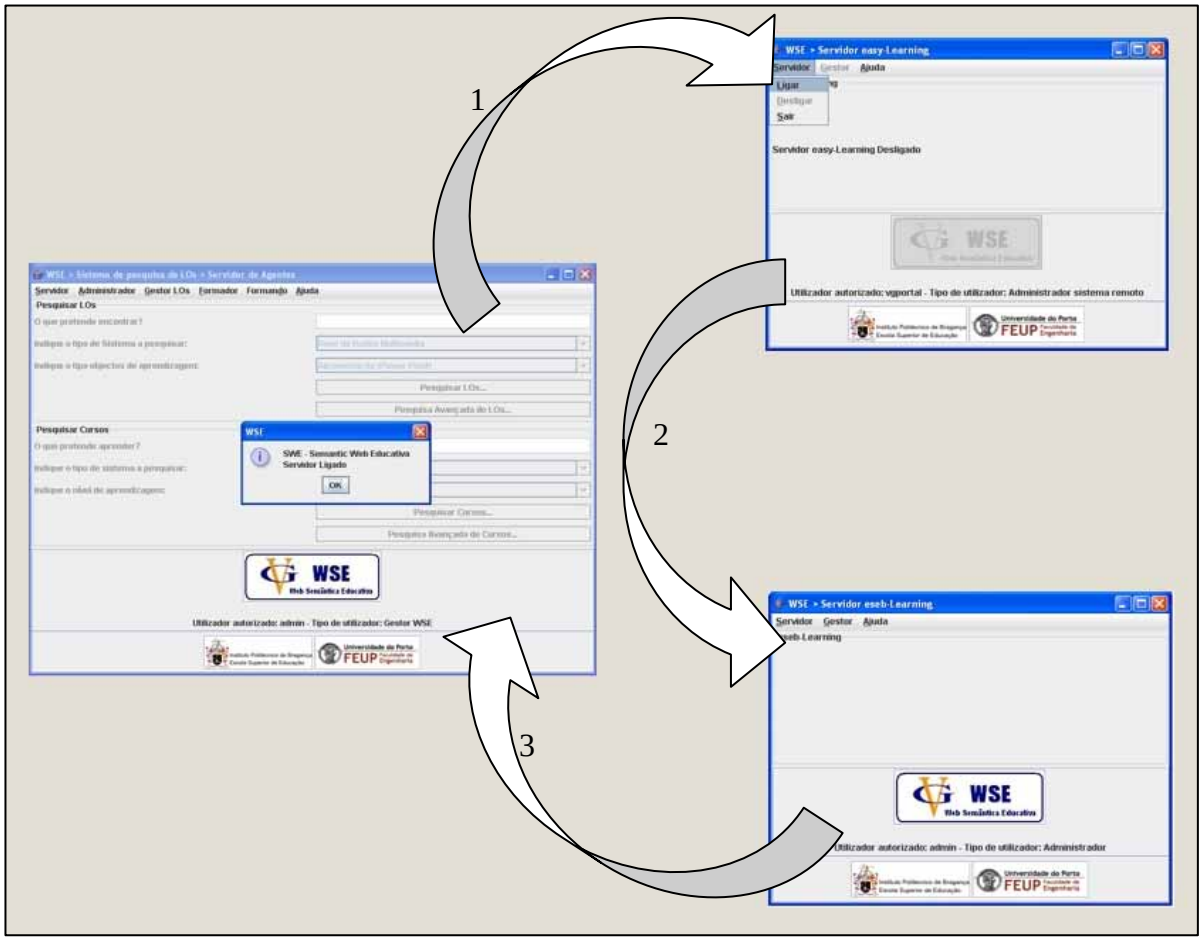


Figura 107 – Aplicação do servidor de agentes e aplicações dos agentes de informação

Do lado esquerdo da figura encontra-se a aplicação do servidor de agentes que acabou de ser iniciada. Do lado direito da figura podemos visualizar duas aplicações para dois agentes de informação (easy-Learning para representar um sistema de e-Learning Moodle e eseb-Learning para representar um sistema de e-Learning Atutor). Estas aplicações são iniciadas automaticamente como serviços aquando do arranque dos sistemas operativos respectivos. No entanto, caso seja necessário, o menu Servidor permite ligar ou desligar as aplicações. Como se pode observar na figura, o servidor de agentes Voyager informa que acabou de ser ligado, o servidor easy-Learning Voyager está a ser ligado e o servidor eseb-Learning Voyager já se

encontra ligado. Se os servidores Voyager estiverem desligados, não será possível receber ou enviar agentes de pesquisa, praticamente todas as opções encontrar-se-ão inactivas e na janela principal é visualizada uma mensagem alusiva a esse facto. Caso os três componentes estejam ligados e a funcionar devidamente, os agentes de pesquisa podem migrar entre os servidores, por exemplo de acordo com a sequência indicada na Figura 107.

7.2.5- Utilitários para Metadados, Ontologias e Mapas de Tópicos

O servidor de agentes fornece os formulários necessários para a gestão da BD do sistema, incluindo a descrição, através de metadados, dos sistemas remotos. Por sua vez, cada um dos agentes de informação possui os formulários necessários para a descrição dos objectos de aprendizagem que disponibiliza, tal como veremos nas secções seguintes.

Além da interoperabilidade sintáctica e estrutural fornecida pelo XML, XML *Schema*, RDF e RDF *Schema*, a WS necessita das linguagens para ontologias de modo a aumentar a expressividade semântica nas descrições dos LOs e promover a interoperabilidade semântica entre aplicações. Para tal, é necessário utilizar uma linguagem ontológica compatível com as linguagens referidas inerentes às camadas inferiores da arquitectura da WS, com sintaxe rica para representar o conhecimento e regras para permitir a inferência de novos dados, e de fácil compreensão e extensão.

Neste sentido, para além da linguagem RDFS, este trabalho recorreu à linguagem OWL, uma vez que é a norma recomendada para a WS com o objectivo de satisfazer os requisitos supracitados. Tal como vimos, a linguagem OWL fornece suporte às descrições RDF, abstracções de classes, generalização, agregação, relações de transitividade, simetria e detecção de inconsistências.

De entre as ferramentas disponíveis para a criação e edição de ontologias OWL, usámos essencialmente a ferramenta Protégé. Não obstante, também foram usadas as ferramentas *TopBraid Composer*, *Semantic Works*, *OntoEdit* e, pontualmente, o bloco de notas para edição dos documentos RDF(S). Embora tenham sido avaliadas outras ferramentas no âmbito da construção da ontologia da página Web de um professor, a detecção de alguns problemas de interoperabilidade inerente à edição dos documentos RDFS ou OWL obtidos motivaram a preferência pela Protégé, para além do facto das ferramentas *Semantic Works* e

OntoEdit serem comerciais, sendo pouco prático instalar constantemente as versões de demonstração dessas ferramentas.

No âmbito do Sistema WSE foram criadas três ontologias de domínio: ontologia geral para o sistema WSE em OWL, ontologia para os sistemas de e-Learning em OWL e ontologia para uma página pessoal de um professor em RDFS. Um resumo do documento ontológico OWL para o sistema WSE é apresentado no anexo H.

O desenvolvimento de ontologias pode ser orientado por diversas metodologias. No âmbito do projecto OntoWeb e projectos similares foram comparadas algumas das principais metodologias: Cyc method, Uschold and King's method, CoMMA, Grüninger and Fox's methodology, KACTUS method, METHONTOLOGY, SENSUS method, $(KA)^2$ e On-To-Knowledge Methodology (Fernández-López, 2002; Almeida et al., 2005). Embora não exista uma metodologia única e exacta para o desenvolvimento de ontologias, podem ser identificadas etapas comuns. No contexto particular desta tese, a construção das ontologias seguiu um conjunto de orientações que podemos resumir nas seguintes etapas:

1) Determinar o domínio da ontologia e enumerar os conceitos relevantes;
2) Definir as classes e as hierarquias;
3) Definir as propriedades e as relações;
4) Definir restrições;
5) Criar as instâncias.

A ontologia geral para o sistema WSE permitiu modelar o domínio dos sistemas externos ou remotos que se registam no sistema. Tal como foi referido anteriormente, a definição do percurso do agente de pesquisa baseia-se nesta ontologia. A anotação semântica dos LOs (normalmente os cursos ou objectos de aprendizagem de elevada granularidade e os recursos ou objectos de aprendizagem de menor granularidade) incluídos em cada um dos sistemas remotos permitirá complementar a anotação semântica dos sistemas registados e, consequentemente, tornar o percurso do agente de pesquisa mais preciso e a recuperação de informação mais célere. Note-se que este ganho em celeridade resultará não de um incremento na velocidade de inferência nos agentes de informação, mas sim de um provável menor número de servidores a visitar pelo agente de pesquisa, já que a anotação semântica dos LOs permitirá refinar ainda mais o percurso desse agente.

O modelo representado na Figura 108 representa, num nível de abstração elevado, o domínio da ontologia, as classes, as subclasses e as relações.

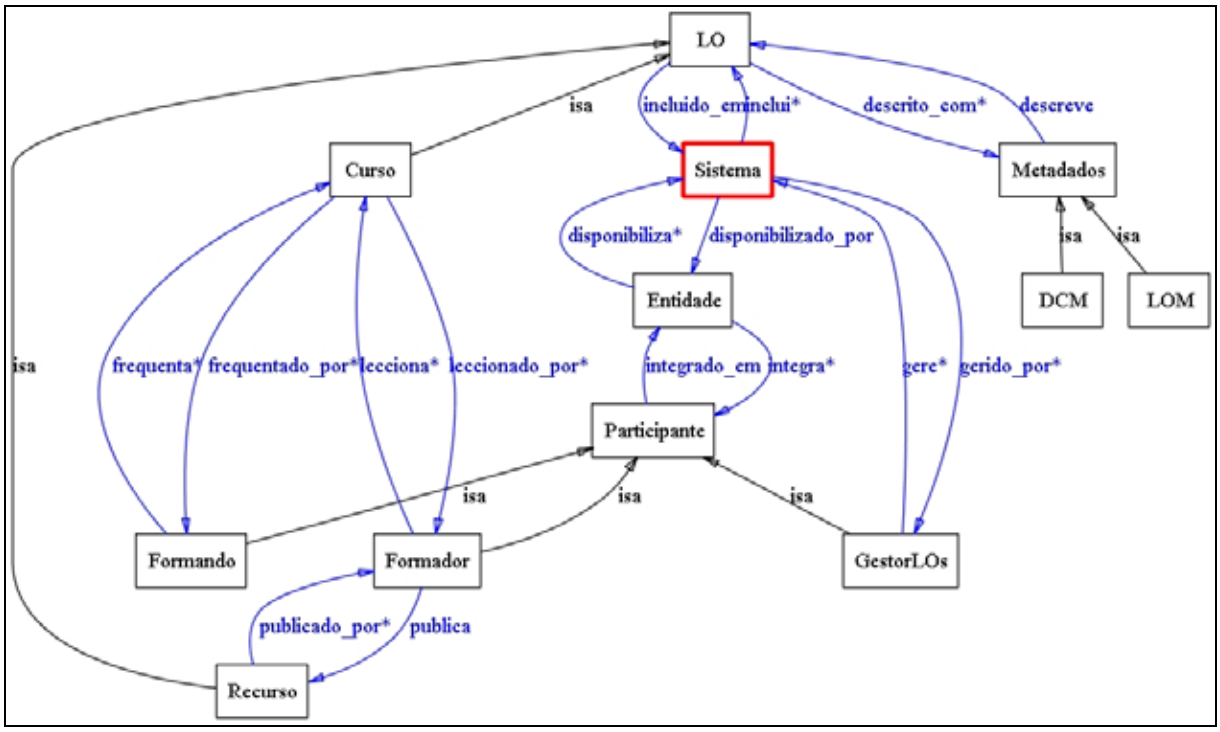


Figura 108 – Modelo da ontologia geral para o Sistema WSE

No que diz respeito à ontologia para os sistemas de e-Learning, o domínio da ontologia e as principais classes e relações são ilustradas através do modelo gráfico representado na Figura 109.

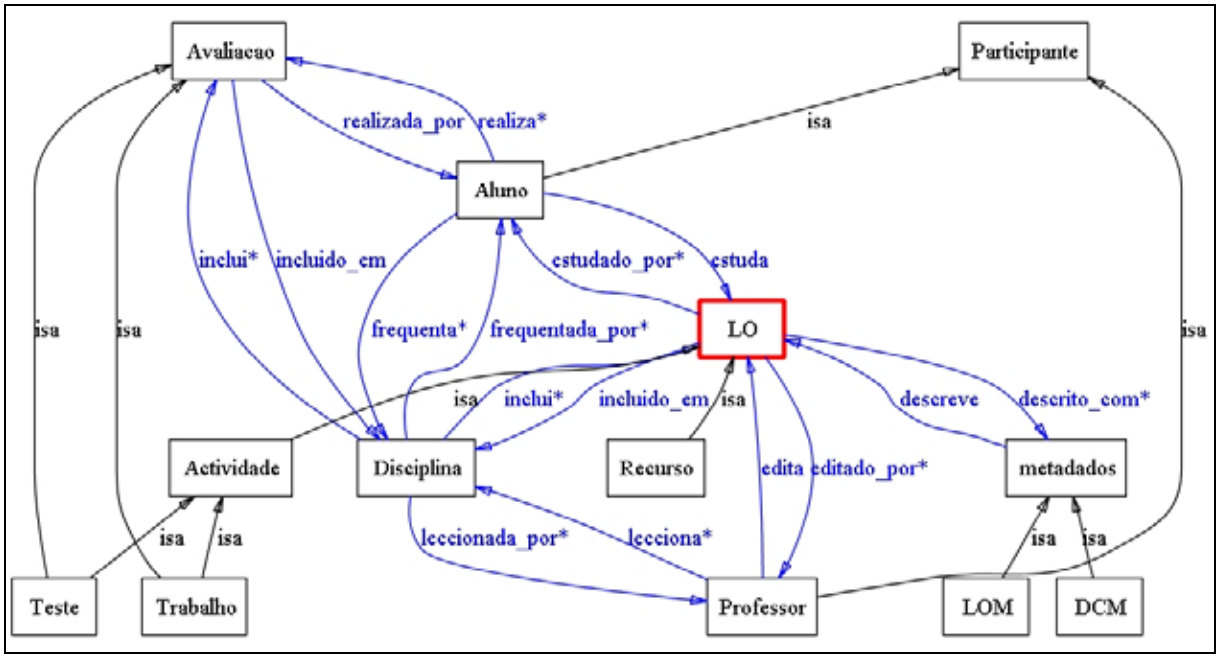


Figura 109 – Modelo da ontologia para os sistemas de e-Learning

A relação “isa” mostra as principais relações entre classes e subclasses. As classes que são descritas para uma relação, ou seja, as classes às quais uma relação está ligada, determinam o domínio.

As classes representadas na Figura 108 e na Figura 109 são subclasses da classe *Thing* (super-classe de todas as classes de uma ontologia). O projecto destas duas ontologias foi desenvolvido com recurso à ferramenta Protégé. A Figura 110 corresponde ao projecto da ontologia para os sistemas de e-Learning.

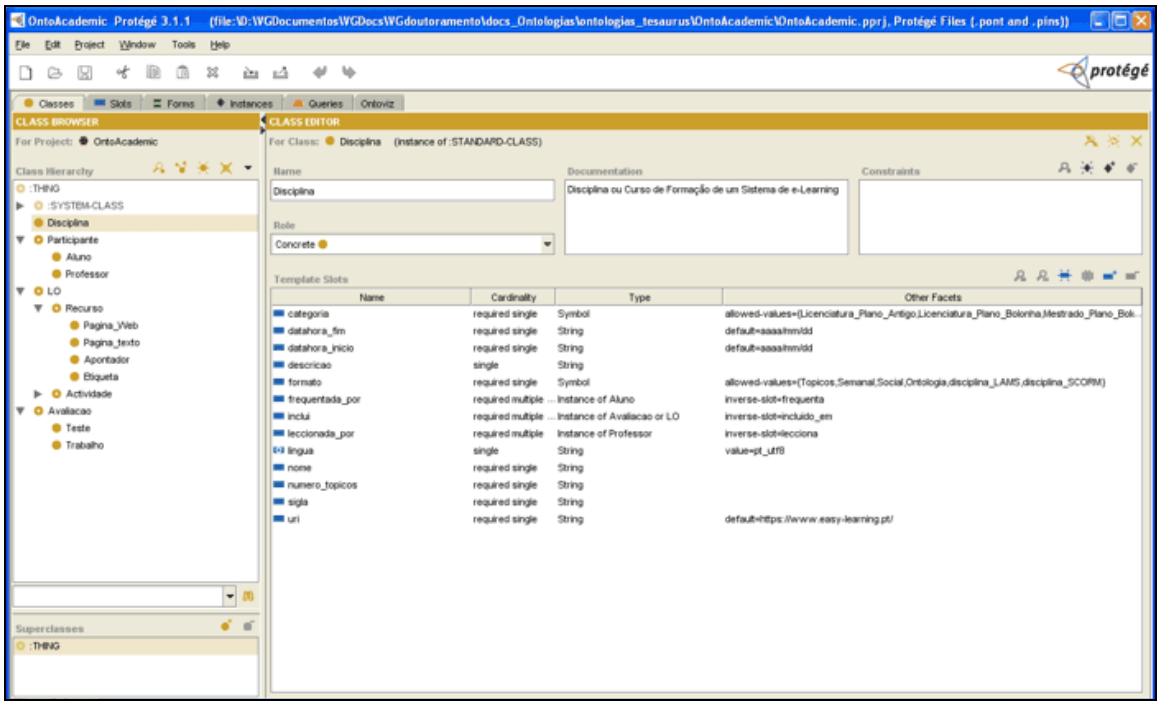


Figura 110 – Protégé: ontologia para os sistemas de e-Learning

Depois de criar as classes e subclasses, foram definidas as propriedades e as relações. De salientar que uma classe herda as propriedades que tiverem sido definidas para as suas superclasses. A Figura 111 ilustra a interface para a definição de uma relação.

Figura 111 – Protégé: criação de uma propriedade

De seguida, procedeu-se à definição de restrições das relações ou propriedades. As restrições (*facets*) descrevem características restritivas para as classes, nomeadamente através do tipo de valor da relação (*string*, número, booleano, etc), da faixa de valores possíveis para a relação (*range*), das relações definidas para uma classe (*domain*) e da cardinalidade da relação. Por exemplo, na ontologia para o sistema WSE definimos que um sistema remoto deve incluir pelo menos três LOs (cursos ou recursos) e que apenas pode ser classificado com uma categoria. Isto significa que, caso um servidor aloje dois ou mais sistemas, cada um deles deverá ser registado autonomamente na BD do sistema e, consequentemente, corresponder a uma instância diferente na ontologia. De referir que propriedades tais como "tipo_los" de objectos de aprendizagem e "tipo" de avaliação correspondem a listagens de valores permitidos.

A Tabela 8 apresenta a especificação das classes, subclasses e respectivas propriedades e relações usadas no projecto da ontologia para o Sistema WSE.

Classes	Subclasses	Propriedades e relações
Sistema		url_ip categoria datahora_registo descricao disponibilizado_por gerido_por inclui nome sigla
Participante		areas_interesse datahora_registo descricao email integrado_em login nome
	Formando	frequenta
	Formador	lecciona publica
	GestorLOs	gere
LO		datahora descricao descrito_com incluido_em lingua nome uri_lo tipo_los
	Curso	frequentado_por leccionado_por nivel_aprendizagem
	Recurso	publicado_por
Entidade		integra disponibiliza nome uri
Metadados		uri_metadado descreve
	DCM	Propriedades correspondem aos elementos das especificações Dublin Core Metadata e IEEE-LOM respectivamente.
	LOM	

Tabela 8 – Classes, subclasses e atributos da ontologia para o Sistema WSE

A Tabela 9 apresenta a especificação das classes, subclasses e respectivas propriedades e relações usadas no projecto da ontologia para os sistemas de e-Learning.

Classes	Subclasses	Propriedades e relações
Disciplina		categoria datahora_fim datahora_inicio descricao formato frequentada_por inclui leccionada_por lingua nome sigla uri
Participante		descricao e-mail fotografia lingua login nome uri_foaf
	Aluno	estuda frequenta realiza
	Professor	edita lecciona
LO		descricao descrito_com nome uri_lo
	Actividade Chat Forum Glossario Inquerito Licao Livro Programa_da_disciplina Questionario Sondagem Sumario_da_aula Teste Trabalho Wiki Workshop	datahora_fim datahora_inicio …
	Recurso Apontador Etiqueta Pagina_texto Pagina_Web	…
Avaliacao		datahora_fim datahora_inicio descricao incluida_em nota realizada_por tipo
	Teste	questoes
	Trabalho	tema
Metadados		uri_metadado descreve
	DCM	Propriedades correspondem aos elementos das especificações.
	LOM	

Tabela 9 – Classes, subclasses e atributos da ontologia para os sistemas de e-Learning

Criadas as classes, as subclasses, os atributos, as relações e as restrições, a fase seguinte corresponde à criação das instâncias de cada classe da ontologia. Para tal, o Protégé disponibiliza um editor de formulários que permite criar previamente formulários para suportar a criação e edição de instâncias. A Figura 112 mostra a edição de uma instância da classe "disciplina".

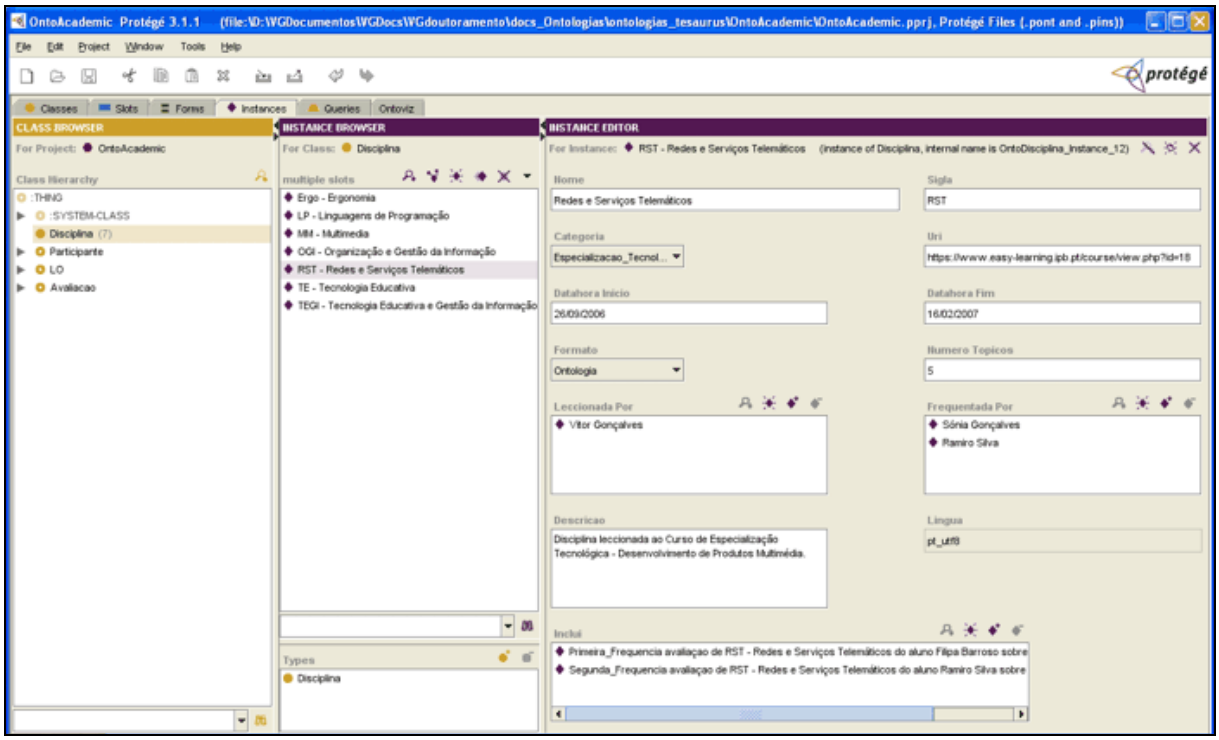


Figura 112 – Protégé: editor de instâncias da ontologia para os sistemas de e-Learning

Finalmente, podemos usar o Protégé para realizar consultas baseadas nestas ontologias através da funcionalidade "queries". As ontologias apresentadas permitem a realização de inferências, tais como: Quais os objectos de aprendizagem sobre um determinado tema ou assunto? Quais os objectos de aprendizagem mais recentes? Quais os instrumentos de avaliação que incidem sobre determinado tema? Quais as disciplinas frequentadas por um determinado aluno? Quais as disciplinas leccionadas por um determinado professor? Quais os objectos de aprendizagem editados por um professor? Qual a avaliação de um determinado aluno numa determinada disciplina? Quais os objectos de aprendizagem mais recentes? Quais os cursos, disciplinas ou objectos de aprendizagem de acesso livre em determinada instituição? Não obstante, os resultados de algumas destas questões apenas poderão ser visualizados pelos utilizadores com autorização para tal.

A funcionalidade "queries" foi usada não só para optimizar as ontologias, mas também para avaliar o grau de satisfação dos utilizadores na obtenção semântica de informação.

Embora a ferramenta Protégé tenha sido extramente útil na concepção e desenvolvimento das ontologias necessárias para avaliar e validar o sistema WSE, é desejável que a edição das mesmas seja efectuada colaborativamente pelos utilizadores do sistema com permissão para tal. Por conseguinte, a solução mais natural para a construção dos módulos de anotação semântica e de edição de ontologias passa pela utilização da API para ontologias OWL da *framework* Jena, por oferecer total compatibilidade com o RDF, liderar o suporte à linguagem OWL e permitir realizar inferências sobre RDF(S) e ontologias OWL.

Para a construção do módulo de mapas de tópicos que permitirá descrever e-cursos remotos e apoiar a navegação semântica, o desenvolvimento tem vindo a ser baseado na API TM4J, uma vez que este conjunto de classes Java fornece as interfaces e implementações para a importação, manipulação e exportação de mapas de tópicos descritos em conformidade com a DTD XTM. Para a navegação nos mapas de tópicos submetidos pelos formadores, nomeadamente os documentos no formato XTM, tem sido usado o *Ontopia Omnigator*.

O desenvolvimento destes módulos ainda se encontra numa fase embrionária, uma vez que na identificação e especificação dos requisitos estes módulos obtiveram prioridade baixa.

O sistema WSE, nomeadamente os componentes agentes de interface, servidor de agentes móveis, agentes de informação e utilitários, foram desenvolvidos na linguagem Java através de IDE NetBeans.

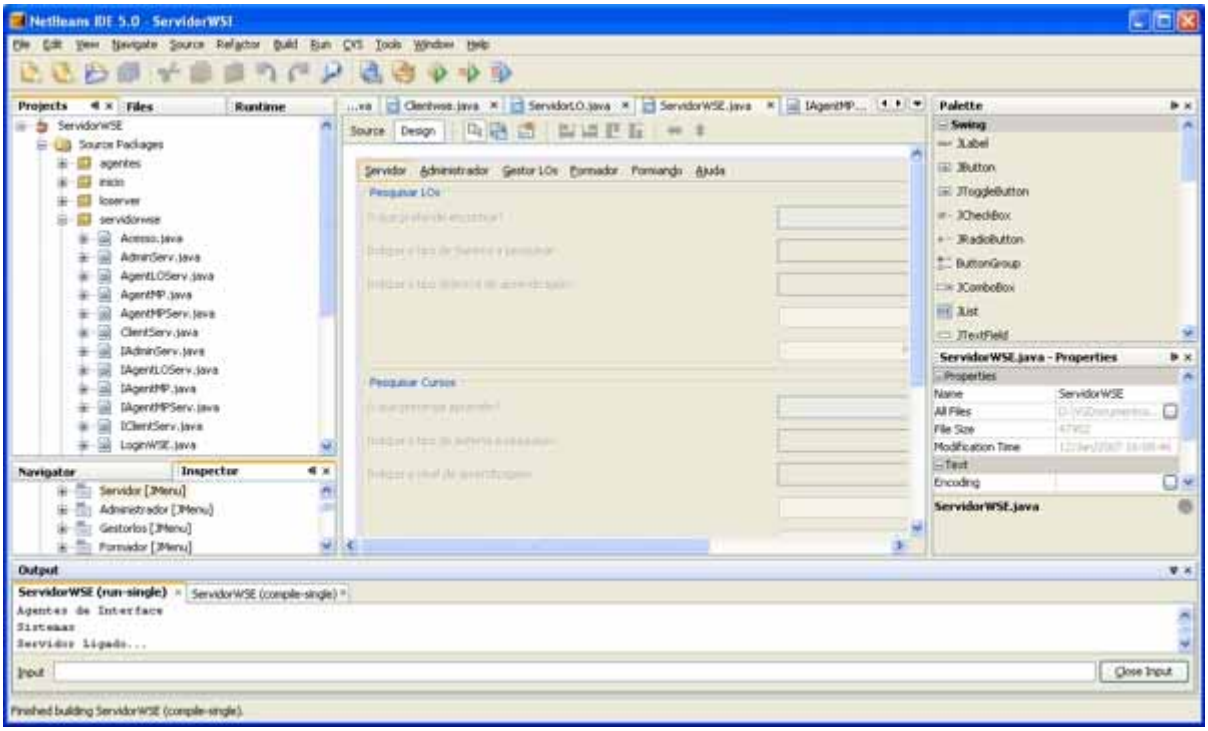


Figura 113 – Actividade de programação do projecto WSE no IDE Netbeans

Para a construção dos objectos deste projecto foram usadas as bibliotecas *Voyager Edge*, *Jena Framework API*, *ARC - Application API Jena*, *MySQL Connector/J*, para além das bibliotecas *JDK*, tal como se pode verificar na Figura 114. De referir ainda que o ambiente de execução do IDE Netbeans permitiu a criação e manutenção da BD do Sistema através do *driver JDBC Connector/J*. Para tal, o tipo das tabelas da BD MySQL usado para esta arquitectura em Java foi o "INNODB" por ser a única que suportava transacções. Pontualmente foi também usado o *phpMyAdmin* através do *browser* Web para gerir a BD do sistema.

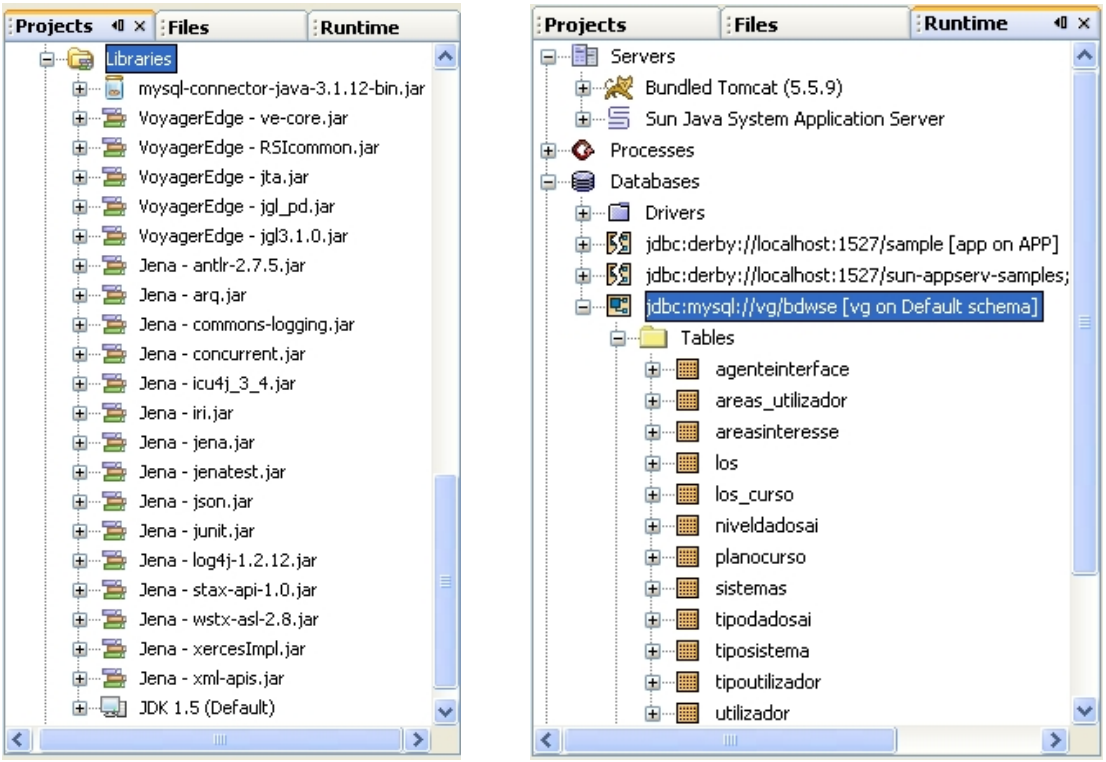


Figura 114 – Bibliotecas e ambiente de execução do IDE Netbeans

7.3- Sistema de e-Learning Moodle

Com vista a integrar as tecnologias para a Web Semântica com as tecnologias para o e-Learning foi necessário avaliar algumas plataformas de e-Learning tanto para o Ensino Superior, como para o Ensino Básico e Secundário.

Uma vez que a arquitectura do sistema WSE proposta no capítulo anterior foi implementada experimentalmente no contexto da Escola Superior de Educação do Instituto Politécnico de Bragança (ESE-IPB), o primeiro passo passou pela avaliação do Sistema de Gestão de Conteúdos Académicos e e-Learning (plataforma Web de ensino proprietária desenvolvida no âmbito da iniciativa *Campus Virtuais*) que tinha acabado de ser disponibilizada à comunidade educativa do IPB no ano lectivo de 2004/2005. Esta solução de e-Learning facilita a gestão das disciplinas dos diversos cursos, permitindo que o docente disponibilize os acetatos, as sebentas, as fichas de disciplina, a bibliografia, entre outros documentos, mas não permite a importação de pacotes SCORM, IMS-CP e IMS-LD, nem a anotação semântica dos documentos importados para a disciplina.

Assim, haveria necessidade de ajustar o sistema de e-Learning actual face à necessidade de usufruir de recursos de aprendizagem devidamente descritos através de metadados. Estrategicamente esta solução tornou-se inviável em tempo útil, para além de ser impraticável uma vez que iria originar alterações não programadas à actividade de desenvolvimento desse sistema.

Uma das plataformas de e-Learning que mais interesse tem despertado nos últimos anos é o Moodle *Course Management System*. Tal como em muitas outras instituições, esta foi a opção de e-Learning e de b-Learning mais natural e viável tanto para a ESE-IPB como para as escolas do Ensino Básico e Secundário. Não obstante, outros sistemas foram instalados para testes comparativos, nomeadamente a plataforma *open source* ATutor.

Antes de passarmos à apresentação das actividades de desenvolvimento levadas a cabo para preparar o sistema de e-Learning Moodle com vista à sua integração no sistema WSE, convém mencionar os parâmetros principais que influenciaram a escolha desta plataforma:

- É um software *Open Source*, permitindo o seu uso e modificação de acordo com as necessidades (respeitando a licença GNU);
- Baseia-se no Construtivismo, nomeadamente no Construcionismo Social;

- Disponibiliza um ambiente intuitivo e agradável caracterizado pela flexibilidade e facilidade na navegação;
- Dispõe de um conjunto de recursos e actividades suficientes para a criação *on-line* de LOs, evitando que os professores ou formadores se limitem a disponibilizar apenas documentos nos formatos pdf, doc, ppt ou zip;
- Permite a importação de pacotes SCORM e IMS-CP permitindo tirar partido das vantagens inerentes à (re)utilização de LOs;
- Permite a importação e execução de pacotes SCORM desenvolvidos em editores, tais como: Editor eXe, Editor RELOAD SCORM e Editor RELOAD *Learning Design*;
- Estava prevista a inclusão de *Learning Design* através de um motor de execução de projectos de aprendizagem, tal como se veio a verificar com a integração do sistema LAMS (*Learning Activity Management System*);
- É um ambiente modular, permitindo o desenvolvimento e a adição de novos módulos e blocos com relativa facilidade, principalmente depois de conhecer e compreender as regras e convenções basilares do desenvolvimento destes dois tipos de componentes.

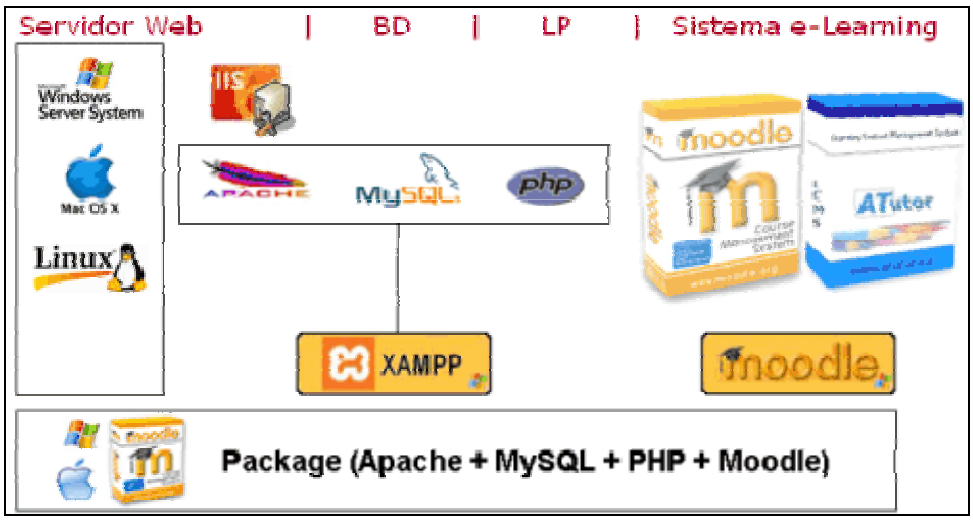


Figura 115 – Diferentes alternativas de instalação e configuração do Moodle e Atutor

Em termos de instalação e configuração do Moodle e do ATutor, existem várias hipóteses: usar um pacote de instalação completo (solução pouco flexível); usar Xampp (pacote de software que inclui apache, MySQL e PHP) e instalar o Moodle e Atutor na pasta pública do servidor Web; ou instalar todos os componentes separadamente. Esta última solução foi a escolhida para a instalação e configuração do servidor uma vez que permite uma

estrutura mais flexível e robusta. Não obstante, inicialmente usou-se a segunda alternativa baseada no software Xampp.

Quanto à preparação do sistema de e-Learning Moodle para facilitar a sua integração com o sistema WSE e satisfazer os requisitos mais prementes dos potenciais utilizadores da ESE-IPB, a solução mais natural e viável, isto é, evitando alterar o núcleo do Moodle, passou pelo desenvolvimento de:

- Dois módulos para a geração (semi)automática de metadados inerentes aos LOs, em conformidade com as principais especificações de metadados – módulo de metadados LOM e módulo de metadados DCM. A decisão de desenvolver dois módulos justifica-se essencialmente com a satisfação dos seguintes requisitos:
  - Permitir que os administradores dos sistemas de e-Learning instalem a especificação que melhor se ajuste às necessidades dos utilizadores ou ambas;
  - Permitir que os formadores possam escolher a norma que melhor se ajusta ao LO que pretendem anotar;
  - Evitar que a falta de uma dessas duas normas fosse motivo para não descrever LOs.
- Dois módulos para a descrição das disciplinas através de metadados: módulo do programa da disciplina ou curso de formação e módulo de sumários das aulas devidamente integrados com os módulos de metadados.
- Bloco de pesquisa semântica baseado em metadados e ontologias que permite recuperar informação baseada nos metadados associados aos LOs, nos metadados associados às disciplinas ou cursos de formação e em ontologias (desde que os professores editem os documentos RDF(S) gerados automaticamente). Assim, este bloco disponibiliza duas formas diferentes de pesquisa: uma baseada nos metadados armazenados na base de dados e outra baseada nos documentos RDF(S) e OWL.

O desenvolvimento de um módulo ou de um bloco para o Moodle deve seguir um conjunto de passos e regras de modo a assegurar que, aquando da sua instalação num sistema de e-Learning Moodle, o grupo de ficheiros que o compõem seja reconhecido como um módulo ou como um bloco. De entre as várias regras de programação que um módulo e um bloco devem respeitar, destacamos (Moodle, 2004):

- O conjunto de ficheiros do módulo ou do bloco deverá ser organizado numa pasta ou directório, cujo nome deverá ser uma única palavra sem espaços, números ou

caracteres especiais. No caso de um módulo, a pasta deverá localizar-se na pasta "mod" da instalação Moodle. No caso de um bloco, a pasta deverá localizar-se na pasta "blocks" da instalação Moodle;

- Os ficheiros de código deverão ter a extensão ".php", enquanto que os ficheiros de modelo deverão ter a extensão ".html";
- As funções usadas deverão corresponder, sempre que possível, às funções declaradas nos ficheiros da pasta "lib" da instalação Moodle, reutilizando o código desenvolvido. Recomenda-se portanto que os acessos à base de dados se baseiem nas funções que constam do ficheiro "../lib/datalib.php" e que os componentes de um formulário Web usem as funções que constam do ficheiro "../lib/weblib.php", entre outras funções ou variáveis gerais definidas no ficheiro "../lib/moodlelib.php", facilitando a normalização e compreensão das linhas de código por terceiros. As funções específicas de um módulo deverão ser definidas num ficheiro com o nome e extensão "lib.php" a guardadas no directório desse módulo;
- Para facilitar a tradução dos textos para outras línguas, deverão ser usadas funções predefinidas do Moodle (nomeadamente "get_string()" e "print_string()"). Estas funções permitem localizar as palavras-chave no ficheiro do pacote linguístico correspondente situado no directório "/lang" do módulo. Obrigatoriamente, este ficheiro deverá ter o nome do módulo ou o nome do bloco seguido da extensão ".php";
- Para filtrar e limpar os textos dentro do Moodle, nomeadamente os que são provenientes dos utilizadores, deve-se recorrer à função "format_text()";
- O nome do bloco ou do módulo e o seu plural são definidos no correspondente ficheiro do pacote linguístico através da atribuição do nome desejado às variáveis "blocktitle", "modulename" e "modulenames";
- Normalmente, todos os ficheiros de código devem incluir o ficheiro "config.php" do Moodle usando *require_once("../../config.php")* uma vez que é necessário para identificar o utilizador através das funções "require_login()", "isadmin()", "isteacher()", "iscreator()" ou "isstudent()";
- Os ficheiros de ajuda devem ser colocados no directório adequado. No caso do pacote linguístico português para a codificação UTF8: *lang/pt_utf8/help*.

A criação de módulos para o sistema de e-Learning Moodle pode ser orientada por um módulo modelo desenvolvido pelos programadores do Moodle e disponibilizado no Website oficial do Moodle (*http://download.moodle.org/download.php/modules/NEWMODULE.zip*). A estrutura deste módulo base inclui os seguintes ficheiros obrigatórios:

- ***mod.html*** – ficheiro que permite a exibição do formulário inicial para criar e modificar uma instância do módulo (objecto resultante do módulo, por exemplo: ficha de sumário N.º 1, formulário do plano da disciplina X, ficha da especificação de metadados Y, inquérito Z, etc);
- ***version.php*** – ficheiro que indica a versão do módulo e outra informação relevante para posteriormente se realizarem actualizações ou *upgrades* do módulo;
- ***icon.gif*** – imagem representativa do módulo (ícone de tamanho 16x16 *pixeis*);
- ***index.php*** – página que lista todas as instâncias do módulo existentes;
- ***view.php*** – ficheiro que permite ver uma instância do módulo. Normalmente, é usado para visualizar os recursos ou actividades listadas pelo ficheiro "index.php" inerente à página inicial de um curso ou para visualizar uma determinada instância após a sua criação ou actualização;
- ***config.html*** – ficheiro opcional responsável por actualizar as configurações globais para os módulos ou para definir as configurações específicas do módulo;
- ***lib.php*** – ficheiro responsável por definir as funções específicas do módulo. Para evitar eventuais conflitos, usou-se como convenção que o nome de todas as funções específicas do módulo deveria começar com o nome do módulo em minúsculas (*nomemodulo*) e que o nome das constantes específicas do módulo deveria iniciar-se com o nome do módulo em maiúsculas (*NOMEMODULO*). Assim, no mínimo deverão estar definidas as seguintes funções:
    - ***nomemodulo_add_instance()*** – código para criar/adicionar uma nova instância;
    - ***nomemodulo_update_instance()*** – código para actualizar uma instância;
    - ***nomemodulo_modulo_delete_instance()*** – código para eliminar uma instância;

- ***nomemodulo_user_outline()*** – código para listar um resumo das intervenções ou contributos do utilizador activo numa instância particular;
- ***nomemodulo_user_complete()*** – código que fornece os detalhes das intervenções do utilizador activo numa instância particular;
- Entre outras funções disponíveis, mas de implementação opcional, destacam-se:
  - ***nomemodulo_delete_course()*** – código para limpar devidamente os restantes registos depois de ter eliminado todas as instâncias do módulo;
  - ***nomemodulo_process_options()*** – código para pré-processar os dados de um formulário a partir das configurações do módulo.

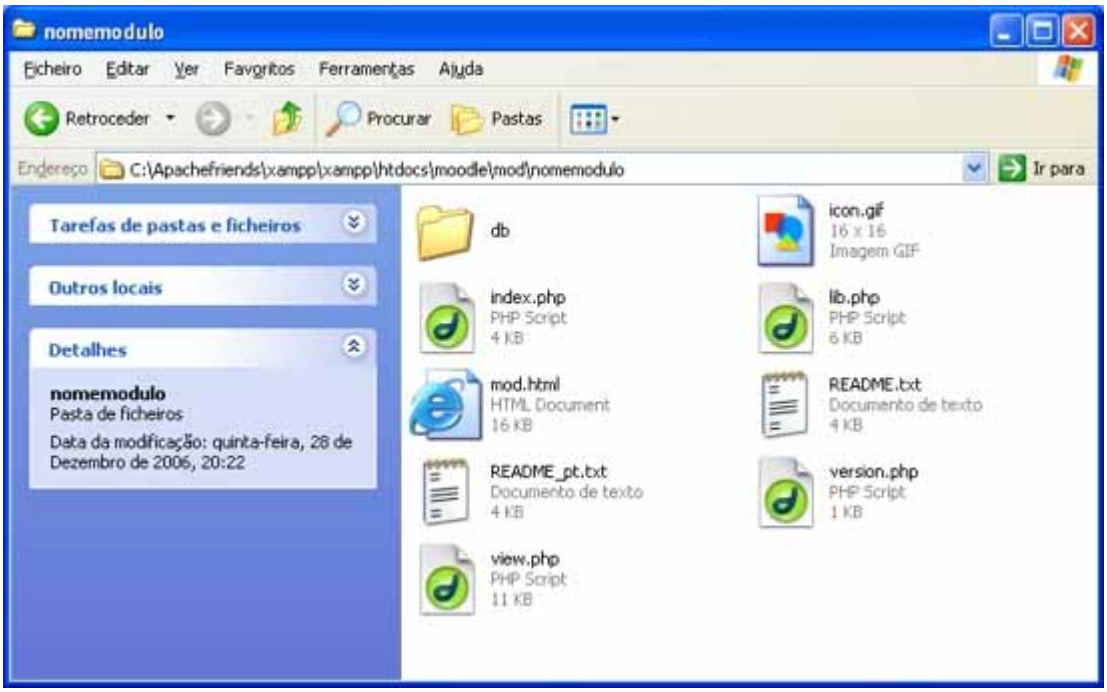


Figura 116 – Directório do módulo modelo

Caso o módulo ou bloco necessitem de utilizar a base de dados do Moodle (ou mesmo outra base de dados), no directório do módulo deve existir uma sub-pasta com o nome "db" (*../mod/nomemodulo/db/*). Essa sub-pasta deverá incluir um ficheiro "mysql.php" e/ou "postgres.php" e também um ficheiro "mysql.sql" e/ou "postgres.sql" consoante o SGBD sobre o qual o sistema de e-Learning Moodle está a funcionar. O ficheiro "mysql.php" ou "postgres.php" é utilizado para eventuais actualizações na base de dados caso já exista uma versão anterior do mesmo módulo ou do bloco instalada no sistema. O ficheiro "mysql.sql" ou

"postgres.sql" contem as instruções SQL que permitem criar ou actualizar as tabelas da base de dados que o módulo ou bloco irá usar. Os nomes das tabelas deverão iniciar-se com "prefix_" seguido de um nome elucidativo do conteúdo que armazenarão (prefix_nomemodulo). O valor de "prefix" será posteriormente atribuído pelo Moodle aquando da execução do código SQL. Por exemplo, a tabela principal deverá ter o mesmo nome que o módulo (*prefix_nomemodulo*) e conter no mínimo os seguintes campos:

- *id* – identificador auto-incremental do módulo como índice primário;
- *course* – para armazenar o identificador do curso ao qual a instância pertence;
- *name* – para armazenar o nome completo de cada instância do módulo.

```
CREATE TABLE IF NOT EXISTS prefix_nomemodulo (
  id INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  course INT(10) UNSIGNED NOT NULL DEFAULT '0',
  name VARCHAR(250) NOT NULL DEFAULT '',
  PRIMARY KEY (id),
  KEY course (course)
);
```

Para além disso, o ficheiro "mysql.sql" ou "postgres.sql" deverá incluir no mínimo as seguintes instruções SQL para actualizar a tabela "prefix_log_display":

```
INSERT INTO prefix_log_display (module, action, mtable, field) VALUES
('nomemodulo', 'view', 'nomemodulo', 'name');
INSERT INTO prefix_log_display (module, action, mtable, field) VALUES
('nomemodulo', 'update', 'nomemodulo', 'name');
INSERT prefix_log_display (module, action, mtable, field) VALUES
('nomemodulo', 'add', 'nomemodulo', 'name');
```

Estas instruções permitem que as instâncias do módulo sejam adicionadas, visualizadas e actualizadas usando o nome completo de cada instância.

Finalmente é conveniente que o directório do módulo inclua pelo menos um ficheiro "readme.txt" com informações sucintas sobre a finalidade e os autores do módulo, as revisões e actualizações que o módulo for sofrendo e os requisitos de instalação.

A instalação de um módulo num sistema de e-Learning Moodle é relativamente simples. Normalmente, os módulos são disponibilizados em formato zip. Para instalar os módulos desenvolvidos no âmbito deste projecto de investigação basta seguir os seguintes passos:

1) Verificar se a instalação do Moodle corresponde à versão 1.6.x ou superior (embora tenha sido testado com sucesso em versões anteriores, recomenda-se a instalação nesta versão do Moodle com vista a tirar proveito de todas as características e funcionalidades dos módulos uma vez que os mesmos foram optimizados para esta versão e para os requisitos da codificação UTF8).
2) Copiar o ficheiro comprimido do módulo para a pasta "mod" da instalação Moodle e descomprimi-lo.
3) Os quatro módulos desenvolvidos para o Moodle foram traduzidos para português, inglês e francês. Para disponibilizar os módulos nas três línguas é necessário copiar os ficheiros "metadatalom.php", "metadatadc.php", "lessonsummary.php" e "program.php" para a pasta do pacote linguístico correspondente (no caso do pacote linguístico português copiar para "moodledata/lang/pt_utf8/").
4) Por fim, usar a conta de administrador para aceder com o *browser* à área de configuração *http://seu_moodle.pt/admin*. As tabelas necessárias serão criadas e a tabela "prefix_log_display" ajustada ou actualizada. Caso seja gerado algum erro, as tabelas podem ser criadas ou actualizadas copiando e executando em SQL as linhas de código do ficheiro "db/mysql.txt" através da aplicação *phpMyAdmin*.

Tal como para o desenvolvimento de módulos para o Moodle, também o desenvolvimento de blocos deve seguir determinadas regras. Para além das regras e convenções já referidas, destacam-se as seguintes características:

- Normalmente na pasta de um bloco ("blocks/nomebloco/") é obrigatória a presença de apenas um ficheiro: "block_nomebloco.php" (ficheiro onde é definida a classe do bloco e as respectivas subclasses e métodos). Opcionalmente, podem surgir os ficheiros "styles.php" (para definir os estilos específicos a usar na apresentação do bloco, isto é, tipos e tamanhos de letra, cores, etc), "config_instance.html" (para definir as configurações de cada instância, isto é, para expressar como deve ser exibida cada instância do bloco) e "config_global.html" (para definir as configurações do bloco, isto é, para expressar como devem ser exibidas todas as instâncias do bloco), entre outros ficheiros mais específicos do bloco.
- Um bloco define a classe principal do bloco como extensão da classe "block_base", a partir da qual se derivam todas as outras classes do bloco e se descreve cada um dos

métodos que podem ser sobrescritos pelos programadores para atribuir funcionalidades ao bloco.

```
class block_nomebloco extends block_base {
   function init() {
      $this->title = get_string('blocktitle', 'block_search_metadata');
      $this->version = 2006100100;
   }

   Function metodo2() {
   ...
   }
   ...
}
```

A primeira linha do ficheiro “block_nomebloco.php” define a classe do bloco, através da forma normalizada que se apresenta no código acima. De seguida, deve ser atribuído um pequeno método: “init”. Este método é essencial em todos os blocos e tem como objectivo principal definir as duas variáveis desta classe: “$this->title” (título que aparece no cabeçalho ou barra de título do bloco) e “$this->version” (versão do bloco: necessária para eventuais actualizações do bloco ou para o caso do bloco recorrer a tabelas da base de dados).

Definido o método inicial, o programador pode usar e subscrever outros métodos de acordo com as funcionalidades que pretenda disponibilizar.

A instalação de um bloco num sistema de e-Learning Moodle é tão simples quanto a instalação de um módulo. Normalmente, os blocos são disponibilizados em formato zip. Para instalar o bloco desenvolvido no âmbito deste projecto de investigação basta seguir os seguintes passos:

1) Verificar se algum dos módulos de metadados referidos se encontra instalado no sistema de e-Learning Moodle. Caso contrário instalar, pelo menos, um dos módulos, descomprimindo o ficheiro correspondente do módulo na pasta “mod”;

2) Copiar o ficheiro comprimido do bloco para a pasta “blocks” da instalação Moodle e descomprimi-lo;

3) O bloco desenvolvido para o Moodle foi traduzido para português, inglês e francês. Para disponibilizar os módulos nas três línguas é necessário copiar os ficheiros “block_search_metadata.php” e “allmetadata.php” para a pasta do pacote linguístico correspondente (no caso do pacote linguístico português copiar para “moodledata/lang/pt_utf8/”);

4) Usar a conta de administrador para aceder com o *browser* à página de configuração dos blocos. O bloco será instalado automaticamente e o administrador informado do sucesso da instalação.

7.3.1- Módulos de Metadados para os LOs

Os módulos de metadados desenvolvidos no âmbito deste projecto de investigação permitem descrever os recursos e actividades (LOs de uma disciplina ou curso de formação Moodle) de um sistema de e-Learning Moodle através de metadados, de acordo com as especificações referidas anteriormente no capítulo III.

Com vista a aliviar o professor do processo tedioso de preencher todos os campos de metadados e evitar eventuais erros de anotação por falta de familiarização com as especificações, exigia-se que a descrição dos LOs através de metadados fosse automatizada tanto quanto possível.

Neste sentido, a solução passou pela criação de dois módulos para anotação (semi)automática de metadados. Um em conformidade com a especificação DCM e o outro de acordo com a especificação IEEE-LOM (e IMS-LRM para LOM), deixando a decisão da escolha para o professor. Embora a norma LOM seja a mais adequada para descrever LOs, alguns dos recursos e actividades podem ser anotados com DCM não só porque o grau de complexidade é menor, mas também porque no caso de artigos, posters, manuais ou referências documentais poderá ser a especificação mais adequada.

No âmbito da especificação *Dublin Core Metadata*, o módulo de metadados DCMES subdividiu-se em três esquemas ou níveis de detalhe: *Simple Dublin Core Metadata*, *Qualified Dublin Core Metadata* e, a título experimental, *Qualified Dublin Core Metadata* integrando alguns elementos de IEEE-LOM. No esquema *Simple* DC destaca-se o seguinte:

- Um registo *simple* DC é composto por uma ou mais propriedades e respectivos valores associados;
- Cada propriedade é um atributo do recurso ou actividade que está a ser descrito;
- Cada propriedade corresponde a um dos 15 elementos do DCMES;

- As propriedades podem ser repetidas;
- Cada valor é do tipo *string* literal;
- Cada valor pode ter uma língua associada (por exemplo: pt);
- Uma boa prática para interligar o recurso aos metadados *simple* DC é usar o URI do recurso como valor do elemento *identifier*.

O esquema *Qualified* DC pode ser visto como uma extensão do *Simple* DC. Para além destas características, no contexto do esquema *Qualified* DC destaca-se ainda o seguinte:

- Cada propriedade pode ser qualquer elemento pertencente a um dos 15 *core elements* ou a um dos outros 7 elementos recomendados pelo DCMI, bem como um dos elementos de refinamento (*element refinements*);
- Cada valor pode ter um esquema de codificação associado (*associated encoding scheme*);
- Cada esquema de codificação tem um nome (por exemplo: UDC - *Universal Decimal Classification*).

O processo de adicionar metadados aos recursos ou actividades pelos professores ou formadores não implica a necessidade de aprendizagem ao nível da utilização uma vez que a interface segue os mesmos passos de qualquer outro módulo de actividades.

Assim, o primeiro passo para anotar os objectos de aprendizagem (recursos ou actividades), corresponde à escolha da actividade a adicionar: "adicionar uma actividade > Metadado LOM (ou Metadado DCMES)". O segundo passo é a escolha do objecto de aprendizagem a descrever. Após clicar no botão "próximo", o utilizador será informado de que os metadados foram gerados automaticamente e de que será redireccionado para a página de edição dessa instância de metadados. A maioria dos elementos é gerada com base na informação textual dos LOs ou outras informações existentes na base de dados ou nas configurações da plataforma do Moodle.

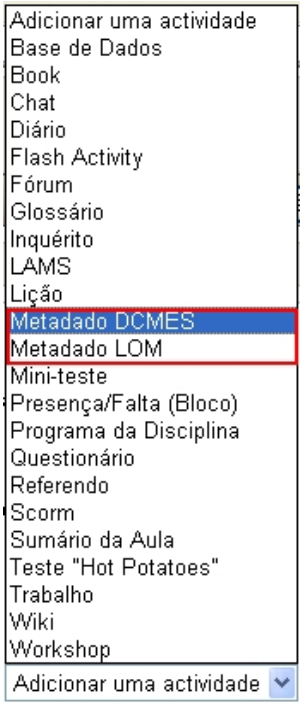


Figura 117 – Adicionar metadados a LOs no Moodle

Finalmente, o terceiro passo corresponde à gravação desse registo de metadados na base de dados e à geração dos documentos XML e RDF/XML, bem como à correspondente visualização dessas instâncias pelo utilizador.

Portanto, a adicionar metadados aos objectos de aprendizagem está simplesmente à distância de 3 cliques, tal como se pode observar na sequência de imagens da Figura 118.

Aquando da geração (semi)automática de metadados, os metadados em XML e RDF/XML são guardados na directoria de cada curso, para além de serem guardados na pasta acessível ao sistema WSE. Caso não exista, a directória "*dir_curso/metadata*" será automaticamente criada. Optamos por guardar os ficheiros na pasta do curso para permitir que o professor possa aceder aos metadados sempre que necessitar através do correspondente menu "Ficheiros".

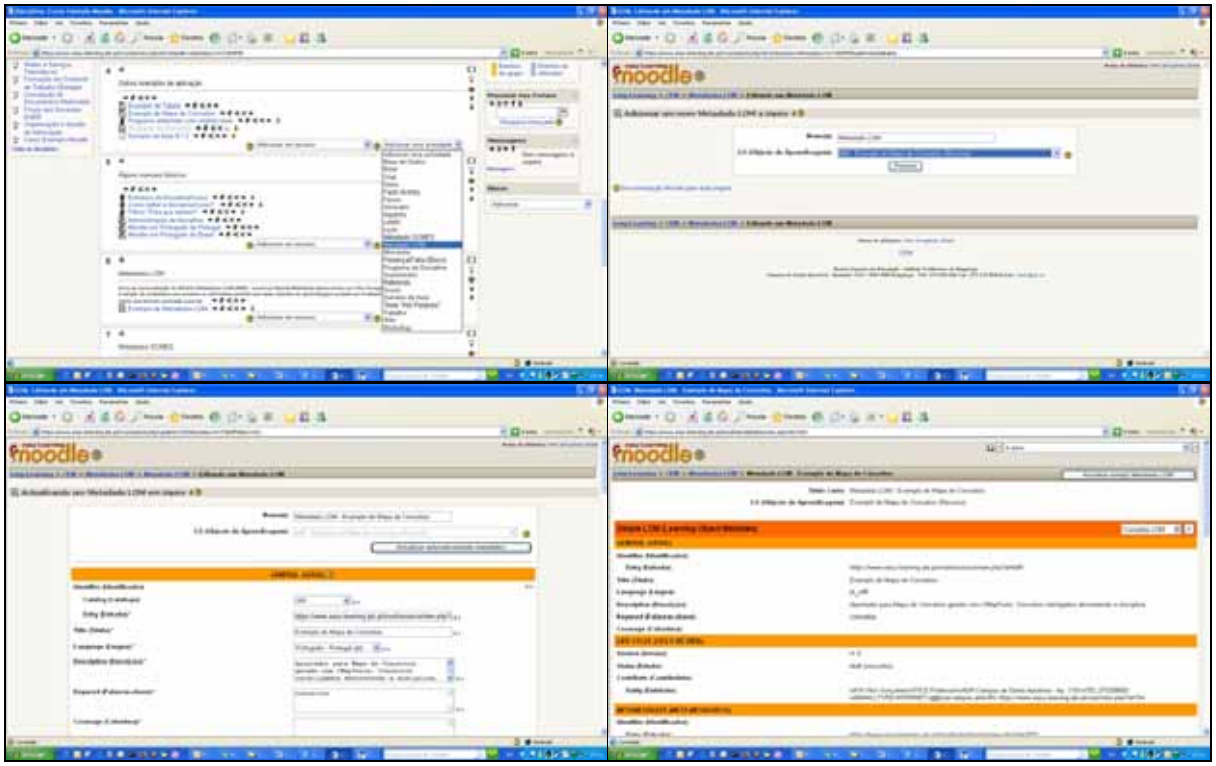


Figura 118 – Processo de adição de metadados LOM a um recurso

Tal como referimos, um dos requisitos para que a descrição dos LOs de um curso tivesse sucesso passava pela automatização do processo de criação de metadados tanto quanto possível. Para tal, alguns campos são preenchidos automaticamente com informações existentes nos LOs, na disciplina ou no curso de formação, ou no próprio Moodle.

No caso do módulo de metadados DCMES, dos 15 elementos do *Simple DC*, treze são gerados automaticamente, e dos 22 elementos e 33 qualificadores do *Qualified DC*, 30 são

gerados automaticamente. Destes, os formadores do caso de estudo identificaram 11 elementos sobre os quais ocasionalmente têm necessidade de proceder a alterações (✍), tal como se pode observar no anexo I. No caso do módulo de metadados LOM, dos 35 (sub)elementos que compõem o *Simple* LOM (elementos e sub-elementos seleccionados para formar um perfil ou esquema de metadados menos complexo para descrever os LOs do Moodle), 28 foram gerados automaticamente, e dos 58 elementos e sub-elementos do *Complete* LOM (versão completa da especificação IEEE-LOM 1484.12.3/D2-2004), normalmente 46 elementos são gerados automaticamente (este valor poderá ser ligeiramente inferior caso o tipo de LOs derive de um módulo de actividade novo). Destes, apenas 9 elementos foram identificados pelos formadores deste caso de estudo como elementos em que pontualmente têm necessidade de proceder a alterações (✍), tal como se pode observar no anexo J. Por conseguinte, entre 37 e 46 elementos são gerados automaticamente dependendo do tipo de recurso ou actividade a anotar. Os restantes elementos poderão ser preenchidos pelo professor para aprimorar a localização e recuperação de LOs. Mais importante do que saber o número de elementos que são gerados automaticamente, é saber quais e como foram gerados, pelo que a leitura dos anexos I e J é indispensável para complementar esta descrição.

Quando as características de um LO são alteradas, a geração de metadados é automaticamente despoletada. No entanto, com excepção dos módulos do programa e dos sumários, esta opção só está disponível para os sistemas de testes (e não nas distribuições para instalar destinadas a outros sistemas Moodle) uma vez que houve necessidade de alterar o código dos ficheiros do Moodle Core.

Para a geração de palavras-chave foi usado um algoritmo simples baseado numa classe responsável pela contagem do número de termos ou conceitos existente no LO, optimizado com a eliminação de todos os advérbios, pronomes pessoais ou outras palavras introduzidas pelo administrador do sistema e com a definição de termos preferidos (conjunto de termos ou conceitos relevantes baseados no Tesauro Europeu de Educação editados pelo administrador do sistema). Tal como mencionado nos vários estudos referenciados anteriormente, este tipo de extracção de metadados é ineficaz. Pelo que, munir a extracção de palavras-chave de um algoritmo mais eficiente e eficaz constitui um dos trabalhos futuros.

A título de exemplo apresentamos a codificação de quatro elementos de metadados. Antes de mais, refira-se que escolhemos os menos extensos para evitar expor aqui muitas linhas de código PHP. O primeiro refere-se ao elemento *Meta-Metadata Contribute Entity*. A

exibição dos dados desse elemento, visualizada na Figura 119, corresponde essencialmente ao seguinte código PHP:

```
$form->MetaMetadata_Contribute_Entity = '\nFN:' . $USER->firstname . ' ' .
    $USER->lastname . '\nTITLE:Professor' . '\nADR:' . $USER->address . '\nTEL:' .
    $USER->phone1 . '\nEMAIL\;TYPE=INTERNET:' . $USER->email . '\nURL:' .
    $CFG->wwwroot . '/user/view.php?id=' . $USER->id  . '\n';
```

O segundo e o terceiro pertencem à extracção do formato e tamanho do ficheiro do LO relativa aos elementos *Technical Format* e *Technical Size*:

```
$url_resource = $form->Technical_Location;
$ary_header = get_headers($url_resource, 1);
$filesize = $ary_header['Content-Length'];
$type = $ary_header['Content-Type'];
$form->Technical_Format = $type;
$form->Technical_Size = $filesize;
```

Finalmente, o quarto refere-se à inferência do *token* para o elemento *LifeCycle* Status tomando por base a visibilidade do recurso ou actividade (LO) e a verificação do estado do LO:

```
<?php //consultar os LOs de uma disciplina
$lines = get_array_of_activities($course->id);
foreach ($lines as $key => $line) {
             $cmlo[$key] = $line->cm;
             $modlo[$key] = $line->mod;
             $namelo[$key] = trim(strip_tags(urldecode($line->name)));
             $vislo[$key] = $line->visible; //LO visível
             }
…
//Atribuir às variáveis a string correspondente do pacote linguístico usado e criar o array
$draft = get_string('draft','metadatalom');
$final = get_string('final','metadatalom');
$revised = get_string('revised','metadatalom');
$unavailable = get_string('unavailable','metadatalom');
$oplcs = array ($draft=>$draft, $final=>$final, $revised=>$revised,
$unavailable=>$unavailable, ''=>$empty, 'newvalue'=>$newvalue);
…
//Seleccionar automaticamente o token de acordo com o estado de visibilidade do LO, excepto se
o utilizador decidir editá-lo para inserir um novo valor:
if ($form->LifeCycle_Status == 'newvalue') {
       $form->LifeCycle_Status = $draft; ?>
<input type="text" name="LifeCycle_Status" size="30" value="<?php p($form-
>LifeCycle_Status) ?>">
//Se campo e se recurso ou actividade tiver sido alterado desde a última criação de
metadados: revisto
<?php } else if (($form->LifeCycle_Status == NULL) & (Verify_status_lo($form-
>resource) == true)) {
       $form->LifeCycle_Status = $revised;
       $form->LifeCycle_Status = metadatalom_cut_final($form->LifeCycle_Status);
       choose_from_menu($opeieur, "LifeCycle_Status", $form->LifeCycle_Status,
$form->LifeCycle_Status, '', '0', $return=false, $disabled=false, $tabindex=0);
```

continua…

```
                                                                    …continuação

//Senão, se campo vazio e LO oculto: indisponível
} elseif (($form->LifeCycle_Status == NULL) & ($vislo[$form->resource] == 0)) {
        $form->LifeCycle_Status = $unavailable;
        $form->LifeCycle_Status = metadatalom_cut_final($form->LifeCycle_Status);

        choose_from_menu($opeieur, "LifeCycle_Status", $form->LifeCycle_Status,
$form->LifeCycle_Status, '', '0', $return=false, $disabled=false, $tabindex=0);
//Caso contrário, LO visível: final
} else {
        $form->LifeCycle_Status = $final;
        $form->LifeCycle_Status = metadatalom_cut_final($form->LifeCycle_Status);
        choose_from_menu($opeieur, "LifeCycle_Status", $form->LifeCycle_Status,
$form->LifeCycle_Status, '', '0', $return=false, $disabled=false, $tabindex=0);
} ?>
```

Uma vez que alguns elementos da norma aceitam mais do que um valor, os campos correspondentes podem ser duplicados utilizando duplo ponto e vírgula (“;; ”). É também disponibilizada a opção “Inserir novo valor” para transformar uma caixa de combinação em caixa de texto e permitir a adição de outros termos, para além dos recomendados nos vocabulários de cada uma das especificações, ou a repetição de valores para um determinado elemento.

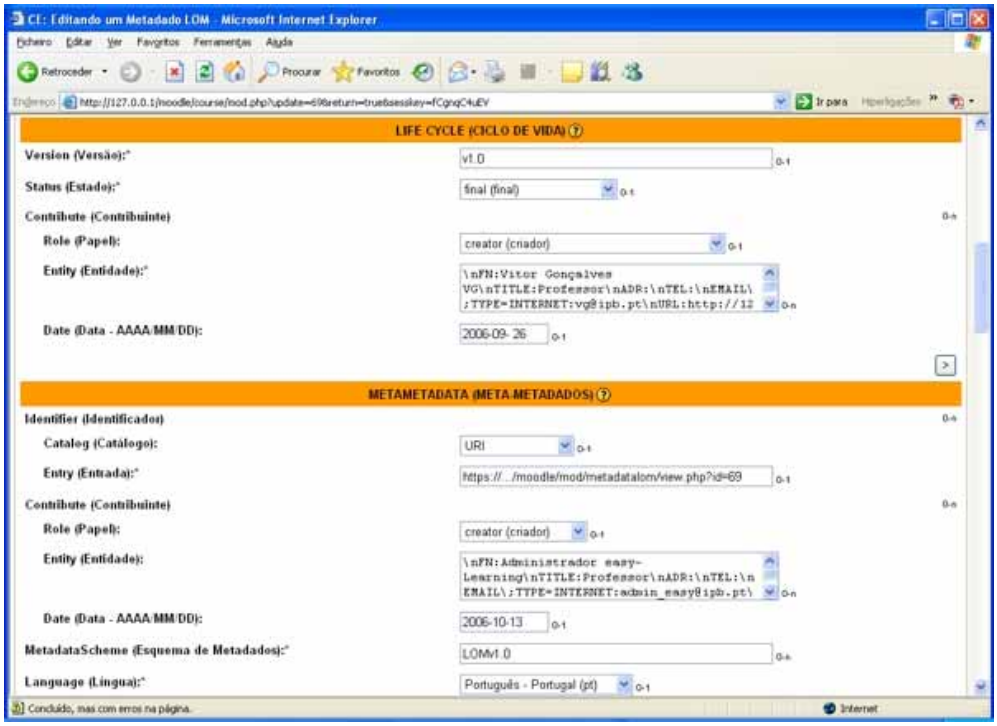


Figura 119 – Metadados LOM adicionados automaticamente a um recurso

No contexto do módulo DCMES, o utilizador tem ao seu dispor três vistas sobre os metadados gerados: *Simple DC*, *Qualified DC* e *Qualified DC* com alguns elementos LOM (mais concretamente os elementos da categoria *Educational*). No contexto do módulo LOM, o

utilizador tem ao seu dispor as seguintes três vistas de apresentação dos metadados: *Simple LOM* (conjunto de elementos seleccionado a partir da norma IEEE-LOM pelos alunos e professores no âmbito da especificação de requisitos), *Complete LOM* (conjunto de elementos completo em conformidade com a especificação IEEE-LOM), *IMS-LRM to LOM* (*Learning Resource Meta-data to IEEE-LOM*).

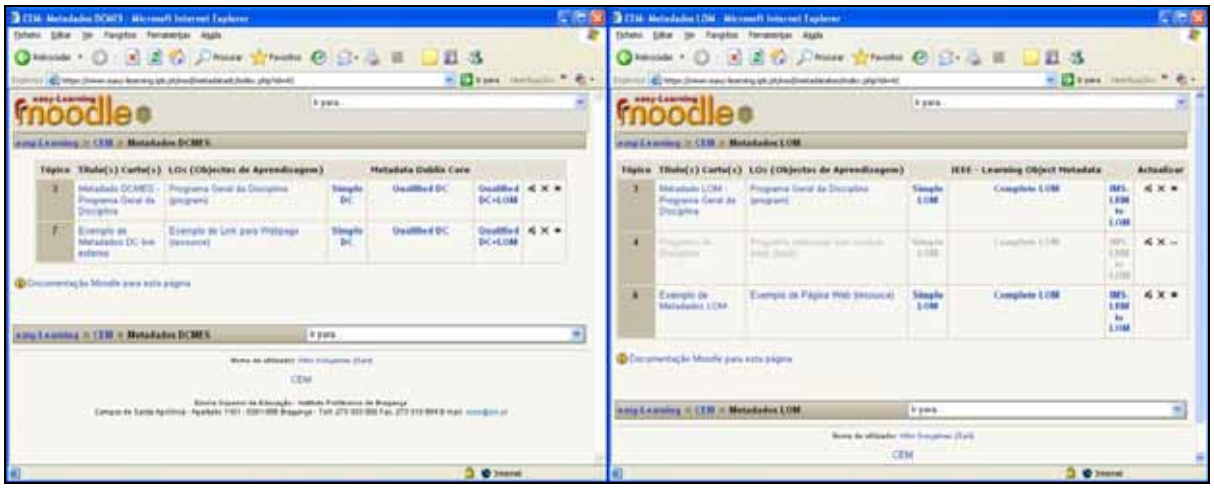


Figura 120 – Vistas disponibilizadas pelos índices dos módulos de metadados

Estas vistas foram identificadas pelos alunos como úteis para obterem detalhes sobre os conteúdos apresentados nos recursos ou actividades. Neste sentido, decidiu-se deixar ao critério dos professores das disciplinas a sua visualização ou não pelos alunos. Por omissão, os metadados são visualizados pelos alunos como um complemento aos recursos que descrevem. Mas, a visualização dos metadados no formato XML ou no formato RDF/XML apenas está disponível para os professores, para além de estarem acessíveis aos agentes de informação e ao motor de inferência SPARQL.

Estes módulos cobrem as principais características das especificações IEEE-LOM, IMS-LOM e DC e foram optimizados para a instituição ESE-IPB. Por exemplo, alguns esquemas de codificação ou catálogos de classificação foram escolhidos por omissão por se tratarem de sistemas de classificação já em uso na ESE-IPB no âmbito da catalogação e indexação de documentos (referimo-nos concretamente ao sistema de classificação décimal universal).

Admitindo que a solução da integração de metadados nos recursos do Moodle através de um módulo possa ter algumas limitações, nomeadamente em termos de usabilidade, parece-nos a alternativa mais viável e rápida para aqueles que necessitam de uma solução de pesquisa através de metadados sem ter preocupações ao nível da estrutura e do núcleo do Moodle (*Core Moodle*) ou mesmo de futuras actualizações desta plataforma *open source*.

De referir que alguns dos botões para geração dos metadados em documentos XML e RDF/XML existentes na vista de formador poderiam ser eliminados. Mas para validar o protótipo no âmbito do processo de geração de metadados decidiu-se mantê-los na interface dos professores com o objectivo de ilustrar as vantagens de uns esquemas em relação aos outros e de permitir que os próprios intervenientes no processo de avaliação do protótipo compreendessem as fases de geração. Contudo, na versão final esses botões são redundantes, uma vez que o módulo gera os metadados em XML e RDF/XML paralelamente à criação/actualização de registos na base de dados.

No que diz respeito à geração dos metadados no formato XML e RDF/XML, a mesma é realizada através de uma classe PHP que gera um documento XML e um documento RDF/XML a partir dos dados armazenados nas tabelas da BD Moodle. Face a algumas limitações pontuais a última versão desta funcionalidade passou a ser suportada pelo RAP (*RDF API for PHP*), software *open source* que disponibiliza funcionalidades para analisar gramaticalmente, procurar, manipular, serializar e disponibilizar modelos de RDF.

7.3.2- Módulos de Metadados para as Disciplinas

Nas primeiras avaliações dos protótipos dos módulos de metadados (LOM e DCMES) realizadas tanto com os professores e os alunos do Ensino Básico e Secundário, como com os do Ensino Superior, concluímos que os módulos para descrição dos LOs com metadados eram pouco utilizados, nomeadamente em recursos e actividades (LOs) já existentes. Houve então necessidade de associar metadados aos cursos. Com base na análise e especificação dos requisitos, a solução que nos pareceu mais viável passou pela criação de mais dois módulos: módulo do programa da disciplina ou curso de formação e módulo de sumários das aulas devidamente integrados com os módulos de metadados existentes. Por um lado, esta solução permitiu-nos associar metadados às disciplinas ou cursos de formação através dos programas e dos sumários (LOs que possuem os conceitos e termos que melhor expressam os conteúdos existentes numa disciplina ou curso de formação) e, por outro, permitiu-nos dinamizar a recuperação de LOs do sistema de e-Learning uma vez que todos os cursos passaram a ter os metadados mínimos para devolver informação.

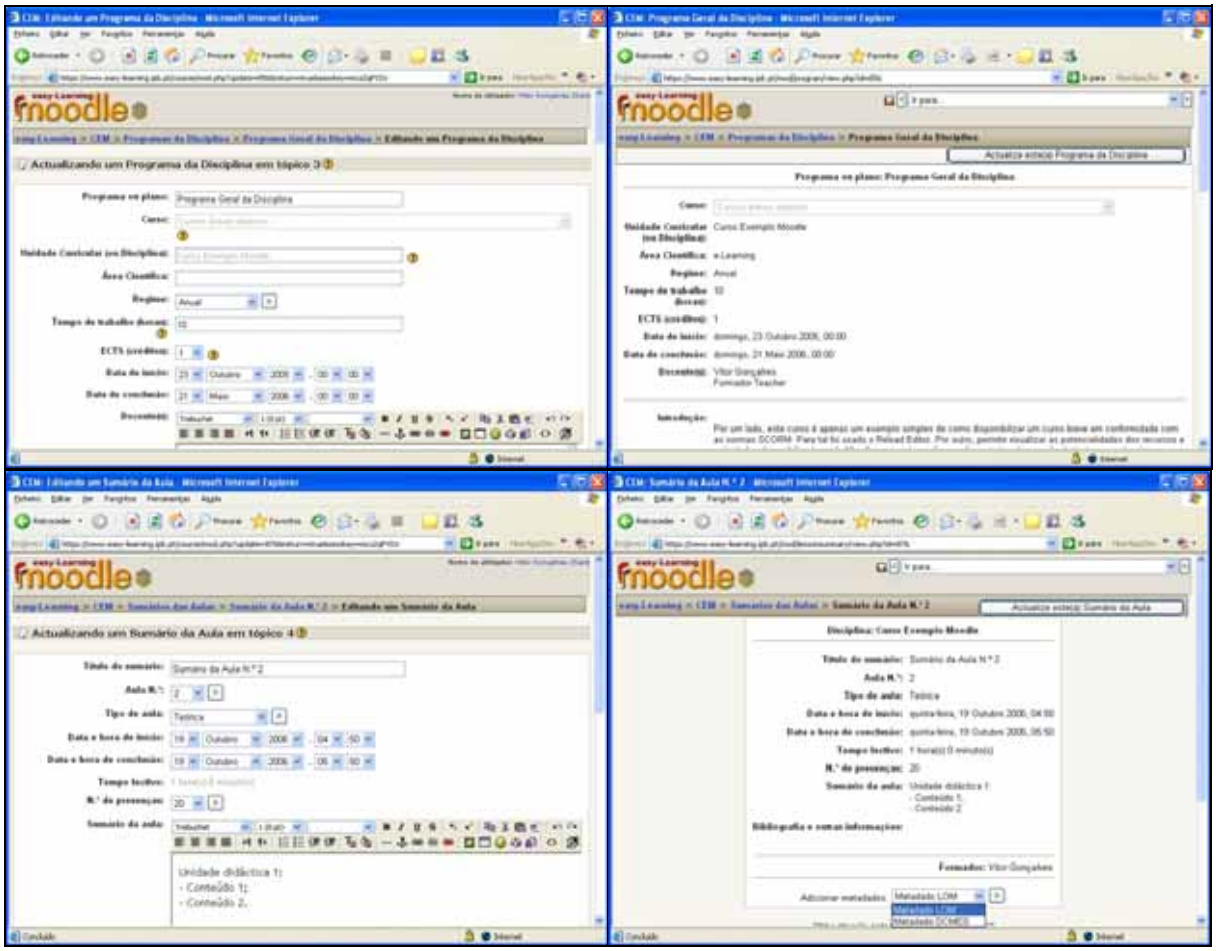


Figura 121 – Módulos de programas e de sumários com metadados

Uma vez que o Moodle não dispunha de nenhum módulo para a publicação normalizada de programas das disciplinas e de sumários das aulas, estes dois módulos permitiram também solucionar estes dois requisitos. Ambos os módulos foram especificados de acordo com as especificidades das disciplinas ou unidades curriculares dos novos planos de Bolonha. Após adicionar um programa da disciplina ou um sumário de aula, o professor é convidado a escolher a especificação de metadados (DCM ou LOM) com a qual deseja descrever automaticamente a instância do módulo programa ou do módulo sumários, tal como se pode observar na Figura 121.

Tal como referido anteriormente, a escolha da especificação fica a cargo do professor uma vez que o esquema DCM exige um menor esforço na compreensão dos seus elementos, para além de poder ser mais adequado para descrever artigos, posters, manuais ou referências documentais. Quando a proposta candidata para que os esquemas DCM e LOM sejam usados conjuntamente em instâncias de metadados DCM constituir uma especificação recomendada, este cenário pode vir a ser alterado, passando a geração automática a basear-se apenas nessa especificação.

7.3.2- Bloco de Pesquisa Semântica

O bloco de pesquisa semântica permite não só recuperar informação baseada nos metadados associados aos LOs e nos metadados associados às disciplinas ou cursos de formação armazenados na base de dados, como também recuperar informação baseada nos documentos RDF/XML gerados automaticamente ou mesmo nesses documentos depois de editados pelo professor ou formador para incluirem mais expressividade em OWL. No entanto, esta última opção é para já utópica uma vez que exigiria que o docente dominasse minimamente a linguagem OWL ou uma das ferramentas que permitem a criação e edição de ontologias OWL.

No que diz respeito à pesquisa baseada em metadados, este bloco disponibiliza duas formas diferentes de pesquisa: uma baseada nos metadados armazenados na base de dados e outra baseada nos metadados expressos nos documentos XML e RDF/XML e na ontologia.

A necessidade de manter estes dois tipos de pesquisa deve-se essencialmente ao facto de nem sempre os docentes adicionarem metadados a todos os recursos e actividades.

A título de exemplo, são apresentadas a seguir as principais linhas de código responsáveis pela pesquisa SQL de recursos ou actividades baseada em metadados DCM armazenados na BD Moodle:

```
function dcmetadata_search_metadata($searchterms, $courseid, $searchcourseid,
$page=0, $recordsperpage=50, &$totalcountdc, $extrasql='')

$searchstring = str_replace("\\\"","\"",$searchstring); //Resolve problema de plicas e aspas
$parser = new search_parser();
$lexer = new search_lexer($parser);

if ($lexer->parse($searchstring)) {//indicação dos elementos a usar na função/classe metasearch_generate_SQL
   $parsearray = $parser->get_parsed_array();
   $metasearch =
   metasearch_generate_SQL($parsearray,'d.title','d.creator','d.subject',
   'd.description','d.publisher','d.contributor','d.date','d.type','d.format',
   'd.identifier','d.source','d.language','d.relation','d.coverage','d.rights',
   'd.userid','d.timemodified');
//Interroga a BD Moodle
$selectsql = "{$CFG->prefix}metadatadc d
             WHERE ($metasearch)
                              $selectcourse $extrasql";
//Devolve o resultado da pesquisa
return get_records_sql("SELECT d.* FROM
             $selectsql ORDER BY d.course, d.timemodified DESC $limit");
```

O bloco de pesquisa está disponível em duas modalidades complementares: pesquisa básica e pesquisa avançada. A pesquisa básica corresponde apenas à pesquisa baseada nos

metadados armazenados na base de dados. Assim, permite pesquisar todos os recursos e actividades dentro de um curso, nos cursos do utilizador ou em todos os cursos, desde que os recursos tenham sido previamente anotados com metadados LOM ou metadados DCM. A pesquisa básica suporta as várias opções listadas abaixo:

- ***procurar estas palavras***: para fazer a pesquisa básica por uma ou mais palavras em qualquer parte dos recursos, basta digitá-las separadas por espaços. Todas as palavras com mais de dois caracteres podem ser usadas.
- ***+procurar +estas palavras***: o exemplo anterior encontraria também a palavra "destas" pois contém "estas". Para forçar a procura de uma palavra exacta, use o sinal mais (+).
- ***+procura –motor***: use o sinal menos (-) caso não queira que determinadas palavras exactas sejam incluídas na procura.
- ***"motor de procura"***: para procurar uma frase específica, use a frase entre aspas duplas.
- ***user:Vitor***: para procurar recursos de um utilizador específico, use o prefixo "user:" seguido do nome desse utilizador.
- ***userid:4***: para procurar recursos de um utilizador específico, também pode usar o identificador desse utilizador, usando o prefixo "userid:" seguido do id do utilizador.
- ***subject:avaliação***: para procurar uma palavra contida no título, assunto ou descrição, use o prefixo "subject:" antes da palavra.

Figura 122 – Bloco de pesquisa: pesquisa básica

É possível combinar as opções acima para especificar uma pesquisa com maior precisão através da modalidade de pesquisa avançada. A pesquisa avançada permite-nos aperfeiçoar os requisitos da pesquisa baseada nos metadados armazenados na base de dados ou, em alternativa, nos documentos RDF(S) e eventualmente nos documentos OWL. Para aceder à pesquisa avançada, pressione o botão de busca (>) sem digitar nada no campo ou caixa de *input* de texto.

No contexto da pesquisa avançada baseada em metadados armazenados na base de dados, para além das opções referidas acima, destacam-se as seguintes características:

- ***Metadados***: a pesquisa poderá incidir em ambas as especificações de metadados ou apenas numa delas (Metadados LOM ou Metadados DCMES). Escolhida a especificação é possível filtrar de acordo com o elemento de metadados.
- ***Cursos***: a pesquisa poderá incidir em todos os cursos, nos cursos do utilizador, num curso específico ou apenas no curso onde o utilizador se encontra.
- ***Tipo LOs***: a pesquisa poderá incidir num tipo específico de metadados, desde que os metadados do recurso ou actividade façam referência a esse tipo.

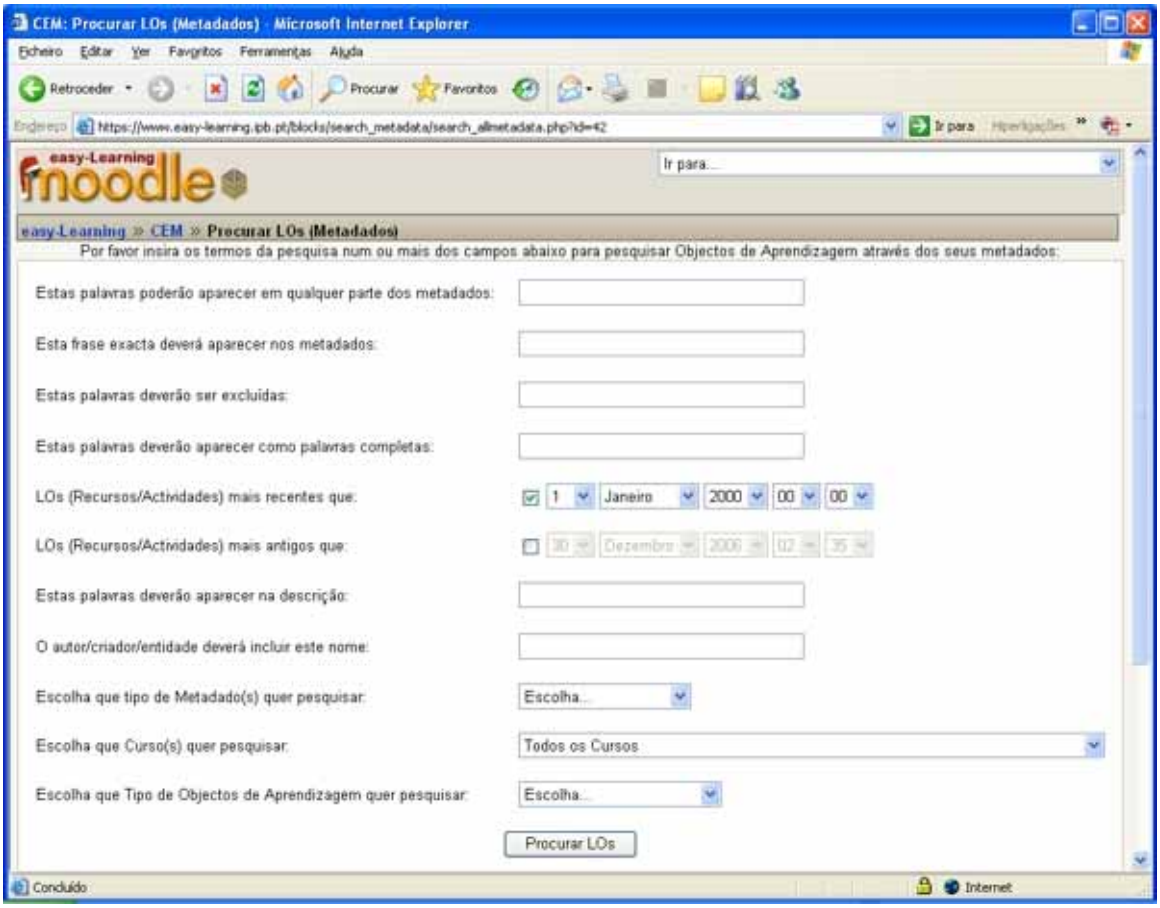


Figura 123 – Bloco de pesquisa: pesquisa avançada

A pesquisa avançada baseada nos metadados e ontologia armazenados em documentos RDF(S) e em documentos OWL pode ser usada accionando a opção Pesquisa SPARQL existente no fundo dessa página.

Uma vez que a pesquisa baseada em metadados apenas fornece pesquisas ao nível sintáctico e estrutural, para usufruir da pesquisa semântica é necessário incrementar a

expressividade semântica através das linguagens RDFS/OWL. Esta componente do bloco de pesquisa semântica baseou-se na *RDF API for PHP*, que inclui o motor de consulta SPARQL e RDQL e respectivas bibliotecas. As linhas de código seguintes demonstram a utilização do SPARQL na consulta dos documentos RDF(S) gerados pelos módulos de metadados.

```
// 1) Incluir as classes do RAP
define("RDFAPI_INCLUDE_DIR", "./../rdfapi-php/api/");
include(RDFAPI_INCLUDE_DIR . "RdfAPI.php");

// 2) Criar um novo MemModel e carregar os documentos na memória
$los = ModelFactory::getDefaultModel();

//$tlos - Array de documentos RDF/XML
foreach($tlos as $i=>$tlo) {
$los->load($tlos[$i]);
}
//$pesquisa - String com palavras-chave fornecidas pelo utilizador

//Exemplo de string de consulta gerada de acordo com os parâmetros do utilizador
$querystring = '
PREFIX dc <http://purl.org/dc/elements/1.1/>
SELECT ?title
WHERE { ?x dc:title ?title .
FILTER regex(?title, '.$pesquisa.', "i")}';

//Executar a consulta SPARQL e visualizar os resultados
$result = $los->sparqlQuery($querystring);
foreach($result as $line){
  $value = $line['?title'];
    if($value != "")
      echo $value->toString()."<br>";
    else
      echo "undbound<br>";
}
SPARQLEngine::writeQueryResultAsHtmlTable($result);
$result2 = $los->sparqlQuery($querystring, 'xml');

//Exemplo de string de consulta gerada de acordo com os parâmetros do utilizador
$querystring = '
PREFIX dc <http://purl.org/dc/elements/1.1/>
PREFIX rdf <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

CONSTRUCT { ?x dc:title ?title}
WHERE { ?x dc:title ?title
FILTER regex(?title, '.$pesquisa.', "i")}';

//Executar a consulta SPARQL e visualizar os resultados
$result = $los->sparqlQuery($querystring);

$result->writeAsHtmlTable();
```

Complementarmente, a pesquisa semântica pode ser aperfeiçoada através de uma interface baseada no servidor *SPARQL Server Joseki* (servidor que suporta o protocolo SPARQL e a linguagem *SPARQL RDF Query Language*).

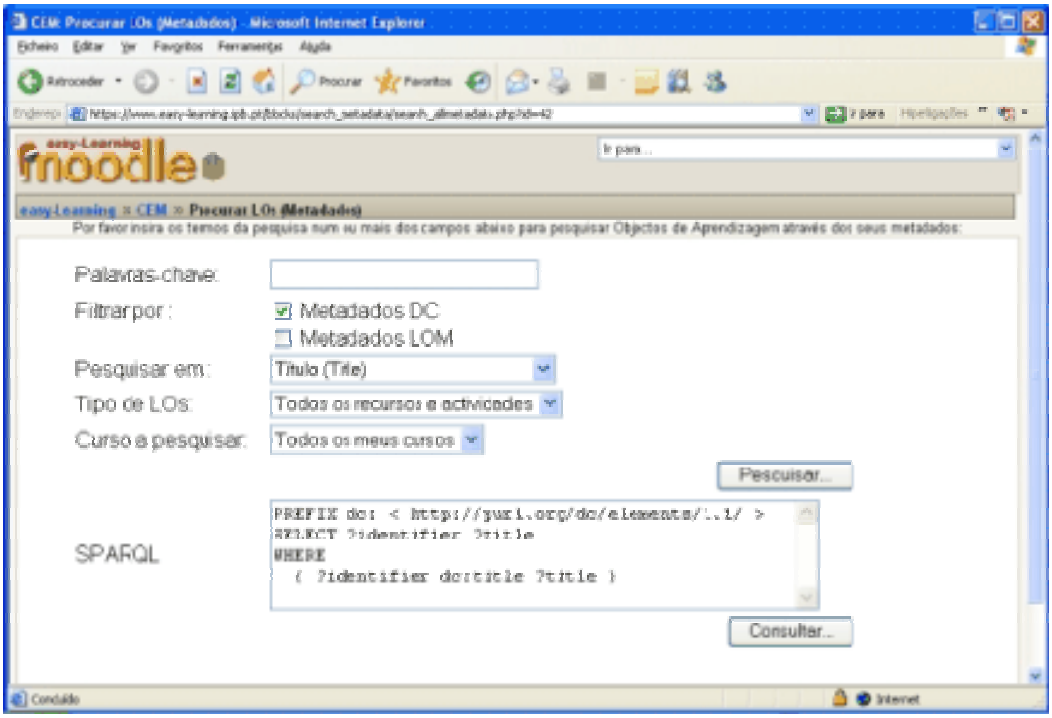


Figura 124 – Bloco de pesquisa: pesquisa semântica

De salientar que este bloco de pesquisa é independente do núcleo do sistema WSE e teve como objectivo imediato solucionar a falta de um mecanismo de pesquisa para localizar todos os objectos de aprendizagem do sistema de e-Learning Moodle. Para além disso, o desenvolvimento deste componente teve também como objectivo servir de base de comparação para avaliar o desempenho do servidor de agentes de pesquisa do sistema WSE, cujos resultados apresentamos na secção 7.5 – Avaliação do Protótipo do Sistema WSE.

7.4- Website do Professor para a Web Semântica

Esta secção tem como finalidade descrever sucintamente a construção de uma Página Web do professor desenvolvida totalmente com tecnologias para a Web Semântica, demonstrando as potencialidades das linguagens, tecnologias e ferramentas no âmbito dos metadados e ontologias e das linguagens de consulta.

O primeiro passo no desenvolvimento de um Website para a Web Semântica foi a anotação das páginas do Website pessoal (*http://www.ipb.pt/~vg/index.htm*), nomeadamente

no que diz respeito à descrição com metadados. Essa página Web pessoal foi anotada no cabeçalho HTML das páginas Web com metadados Dublin Core.

```
<html>
<head>
<!--WebPage anotada com metadados Dublin Core-->
<link rel="schema.DC" href="http://dublincore.org/documents/1999/07/02/dces/#">
<meta name="DC.Title" content="Página Web pessoal do Docente">
<meta name="DC.Creator" content="Vitor Gonçalves">
<meta name="DC.Subject" content="Tecnologia Educativa; Gestão do Conhecimento; Web
Semântica">
<meta name="DC.Description" content="Página pessoal do Docente Vitor Gonçalves do
Departamento de Tecnologia Educativa e Gestão da Informação da ESE-IPB
exemplificando a anotação de páginas Web com metadados Dublin Core">
<meta name="DC.Publisher" content="Escola Superior de Educação - Instituto
Politécnico de Bragança">
<meta name="DC.Contributor" content="Vitor Gonçalves; Francisco Gonçalves">
<meta name="DC.Date" scheme="W3CDTF" content="2004-04-25">
<meta name="DC.Type" scheme="DCMIType" content="Collection">
<meta name="DC.Format" content="text/html">
<meta name="DC.Language" scheme="RFC1766" content="pt">
<meta name="DC.Identifier" content="http://www.ipb.pt/~vg/index.htm">
<meta http-equiv="Content-Type" content="text/html; charset=windows-1252">
<meta http-equiv="Content-Language" content="en-us">
<title>Vitor Gonçalves Home Page</title>
</head>
</title>
```

Uma vez que os motores de pesquisa atribuem cada vez mais menos relevância às *meta tags*, esta página acabou por ser substituída por uma nova anotada com SHOE (*Simple HTML Ontology Extension*) através da ferramenta *Knowledge Annotator* para ser usada por agentes da Web Semântica. Nesta versão foi usada a ontologia SHOE *university* 1.0 (mais concretamente a versão *cs-dept-ontology* do *Computer Science Department Ontology - University of Maryland*) para anotar o ficheiro "index.htm", ou seja, a página principal do Website. Posteriormente, esta página foi complementada com a descrição FOAF do autor. A primeira versão do ficheiro "foaf.rdf" foi gerada com recurso à aplicação *Javascript FOAF-a-matic*.

Com recurso à ferramenta *SHOE Knowledge Annotator* foram gerados predicados para anotar por exemplo a página principal do Website (*index.htm*). Com base nesses predicados, foram geradas as *tags* para adicionar ao ficheiro *index.htm*. Estas *tags* são transparentes para os humanos, mas legíveis para os agentes de software. Mais detalhes podem ser visualizados em *http://www.vgportal.ipb.pt/vg/vgold/sumario_shoe.htm*.

Finalmente, a *RDFhomepage* (*http://www.vgportal.ipb.pt/vg/*) para a Web Semântica veio substituir a versão *SHOEhomepage* (*http://www.vgportal.ipb.pt/vg/vgold/*) anotada com recurso à ferramenta *SHOE Knowledge Annotator*. Esta nova versão baseou-se no projecto *rdfhomepage* (Schwarz, 2005) e o seu desenvolvimento ocorreu paralelamente ao desse

projecto contribuindo activamente para a implementação e avaliação de alguns componentes de software. Genericamente, através da API RAP (*RDF API for PHP*), o conjunto de ficheiros PHP de *RDFhomepage* pode ler diversas fontes RDF, tais como ficheiros FOAF ou ficheiros BibTex (a base de dados BibTex é convertida em RDF através do tradutor ou *parser* "bibtex2rdf") e, posteriormente, a informação pode ser relacionada e apresentada em formato HTML. Assim, tanto a perspectiva dos humanos como a das máquinas (agentes de software) é assegurada, tal como ilustrado na Figura 125.

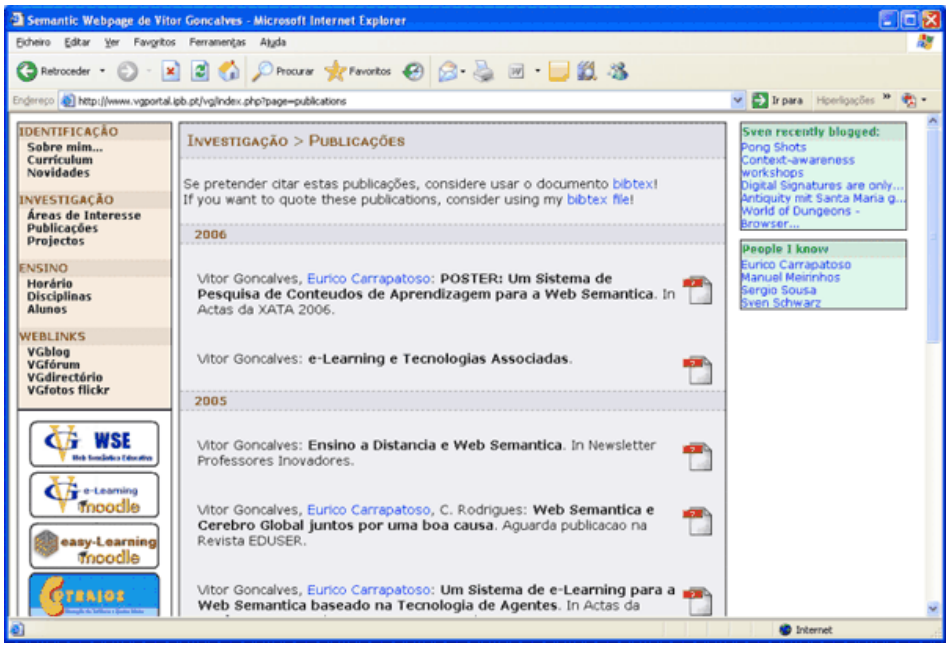


Figura 125 – Página do professor para a Web Semântica

Foi também criado um repositório de fotografias com o intuito de usar o "flickr2rdf". Esta aplicação usa o "Flickr API" para extrair metadados e gerar uma descrição RDF. Em suma, "flick2rdf" converte as *tags* de metadados de uma imagem *flickr* para uma descrição RDF FOAF (*Friend of a Friend*).

Tal como referimos, um dos objectivos do desenvolvimento desta página Web semântica foi a sua integração com o sistema WSE de modo a validar o protótipo. A título de exemplo apresentamos o resultado de uma inferência SPARQL submetida ao servidor de agentes por um utilizador com perfil de formando através do seu agente de interface. Esse formando pretende consultar artigos acerca do tema "Web Semântica". Aquando da recepção do pedido, o servidor de agentes criou um agente de pesquisa móvel, cujo itinerário continha o servidor Web de conteúdos educativos VGportal onde está alojada a página do professor (*http://www.vgportal.ipb.pt/vg/*). Quando o agente de pesquisa chega a esse servidor Web,

identifica-se ao respectivo agente de informação, entrega-lhe o pedido de pesquisa e segue o seu percurso. Com base nos parâmetros entregues, o agente de informação encarrega-se de realizar a pesquisa através do motor de inferência SPARQL e devolve os resultados ao servidor de agentes que, por sua vez, os entrega ao agente de interface do utilizador que os solicitou. Saliente-se que a pesquisa foi efectuada sobre o elemento *dc:subject*, e não sobre o elemento *dc:title*. Todos os artigos devolvidos deverão estar classificados como artigos sobre o tema ou assunto "Web Semântica". Ou seja, sabendo que os parâmetros que chegaram ao servidor de agentes foram: Palavras-chave: "Web Semântica"; Tipo de LOs: "Artigos ou posters"; Metadados: "Todos os metadados"; Elemento: "Assunto"; para além de outros parâmetros definidos que já haviam sido usados do lado do servidor de agentes para construir o percurso: Tipo de sistemas: "Todos os sistemas registados" e Tipo de pesquisa: "Pesquisa semântica baseada nos documentos RDF/XML"; então a string de consulta criada pelo agente de informação para submissão ao motor de inferência SPARQL é:

```
$querystring = '
PREFIX dc <http://purl.org/dc/elements/1.1/>
PREFIX rdf <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
SELECT ?title ?subject
WHERE { ?x dc:subject ?subject . ?x dc:title ?title
FILTER regex(?subject, '.$pesquisa.', "i")}';
```

Como podemos verificar na Figura 126, o quarto artigo "Rumo a uma Web mais inteligente" não contém no título a expressão de pesquisa submetida, mas como está classificado como um artigo sobre "Web Semântica" também faz parte dessa lista. De fora da lista de resultados ficaram artigos que contêm as palavras "Web" e "Semântica" mas que não são artigos acerca da "Web Semântica".

No.	?title	?subject
1.	Literal: Ensino a Distancia e Web Semantica	Literal: Web Semantica
2.	Literal: Um Sistema de e-Learning para a Web Semantica baseado na Tecnologia de Agentes	Literal: Web Semantica
3.	Literal: POSTER: Um Sistema de Pesquisa de Conteudos de Aprendizagem para a Web Semantica	Literal: Web Semantica
4.	Literal: Rumo a uma Web mais inteligente	Literal: Web Semantica
5.	Literal: Web Semantica e Cerebro Global juntos por uma boa causa	Literal: Web Semantica

Base URI:			Size: 5
Prefix:	Namespace:		
-	-		
No.	Subject	Predicate	Object
1.	Resource: http://www.vgportal.ipb.pt/vg/rdf/vg_bib.rdf#Goncalves05	Resource: http://purl.org/dc/elements/1.1/subject	Literal: Web Semantica
2.	Resource: http://www.vgportal.ipb.pt/vg/rdf/vg_bib.rdf#VGEMC2005	Resource: http://purl.org/dc/elements/1.1/subject	Literal: Web Semantica
3.	Resource: http://www.vgportal.ipb.pt/vg/rdf/vg_bib.rdf#VGEMC0206	Resource: http://purl.org/dc/elements/1.1/subject	Literal: Web Semantica
4.	Resource: http://www.vgportal.ipb.pt/vg/rdf/vg_bib.rdf#VGEMC04	Resource: http://purl.org/dc/elements/1.1/subject	Literal: Web Semantica
5.	Resource: http://www.vgportal.ipb.pt/vg/rdf/vg_bib.rdf#VGEMCCSR05	Resource: http://purl.org/dc/elements/1.1/subject	Literal: Web Semantica

Figura 126 – Resultado de uma pesquisa semântica

Este exemplo, por si só, concretiza os objectivos iniciais que nos propusemos. Não obstante, quanto maior for a expressividade do documento XML/OWL, mais preciso será o resultado devolvido. Por conseguinte, este exemplo constitui apenas um pequeno passo rumo à tão almejada Web Semântica no contexto educativo.

7.5- Avaliação do protótipo do Sistema WSE

O sistema WSE tem como objectivo principal permitir a pesquisa e recuperação de objectos de aprendizagem a partir de servidores remotos, nomeadamente aqueles onde se encontram alojados sistemas de e-Learning. Para tal, os objectos de informação devem ser anotados prévia e semanticamente para permitir realizar inferências providas de contexto e significado.

A aplicação das tecnologias para a Web Semântica aos sistemas remotos integrados na infra-estrutura distribuída do sistema WSE pode ser resumida através do esquema da Figura 127 baseado na arquitectura para a Web Semântica e na visão do *Semantic Web Community Portal* (SWCP, 2006).

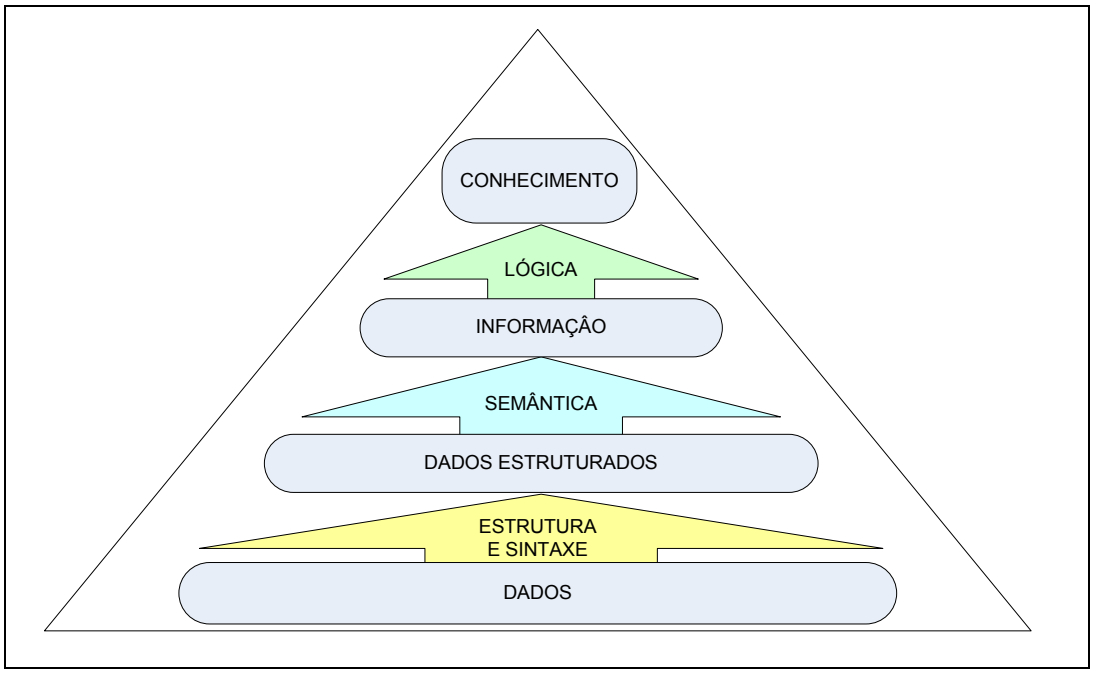


Figura 127 – Aplicação das tecnologias para a Web Semântica

Os dados correspondem aos objectos de aprendizagem, documentos ou páginas Web, a partir dos quais qualquer pessoa consegue obter informação para gerar conhecimento. No entanto, para que os agentes de software possam tirar proveito desses mesmos dados foi necessário aplicar as tecnologias para a Web Semântica.

A XML(S) forneceu a sintaxe e a estrutura que permitiu estruturar os dados. A RDF(S) permitiu expressar os metadados e forneceu a semântica a esses dados estruturados, disponibilizando informação com significado aos agentes de software. A OWL permitiu aumentar a expressividade semântica da informação. Finalmente, a combinação da informação baseada nos metadados, nas ontologias e nas regras de inferência permite que os agentes de software giram conhecimento para os utilizadores que o requisitem.

A avaliação do sistema WSE, incluindo os módulos dos sistemas de e-Learning, foi feita em várias fases, de acordo com o momento ou ciclo de desenvolvimento em que o protótipo se encontrava. Para facilitar a apresentação de resultados vamos dividir a actividade de exploração e utilização em dois momentos decisivos (primeiro protótipo computacional e protótipo final) e em duas partes distintas, mas complementares (as funcionalidades disponibilizadas pelo servidor de agentes do sistema WSE e as funcionalidades disponibilizadas pelos módulos e bloco de pesquisa dos sistemas de e-Learning Moodle). Note-se que, para a anotação semântica dos restantes sistemas de e-Learning (*http://www.eseb-learning.ipb.pt*) e repositórios de objectos de aprendizagem (*http://www.vgportal.ipb.pt* e *http://www.catraios.pt/cbticeb1/*) foi usado um módulo externo disponível através do URL *http://www.vgportal.ipb.pt/metagen/*. No entanto, a utilização deste módulo foi limitada e serviu apenas para garantir que existiam recursos descritos com metadados em todos os sistemas remotos a visitar pelos agentes de pesquisa.

A fase de avaliação do primeiro protótipo decorreu no segundo semestre do ano lectivo 2005/2006 e baseou-se essencialmente na observação directa ou acompanhamento *on-line* de um grupo de utilizadores constituído por:

- Cinco professores do Ensino Básico e Secundário (EBS) leccionando 5 disciplinas do Curso Tecnológico de Informática;
- Cinco professores do Ensino Superior (ES) leccionando 4 disciplinas do Curso de Especialização Tecnológica e 2 acções de formação contínua de professores;
- Cerca de 110 alunos ou formandos.

Criado o primeiro protótipo do sistema WSE e dos componentes para os sistemas de e-Learning, foi indispensável possibilitar o acesso e a utilização dos mesmos pelos potenciais utilizadores. Para tal foram disponibilizados os seguintes URL:

- VG-Learning para o EBS: *http://www.vgportal.ipb.pt/vglearning.htm*
- Easy-Learning para o ES: *https://www.easy-learning.ipb.pt*
- Sistema WSE: *http://www.vgportal.ipb.pt/wse*

Durante a observação da utilização dos primeiros protótipos, foi necessário discutir com os utilizadores eventuais alterações ou novas funcionalidades a implementar. Tal como referido anteriormente, os módulos de sumários das aulas e dos programas das disciplinas foram um exemplo resultante dessa interacção no âmbito dos sistemas de e-Learning. Outros exemplos foram alguns dos resultados expostos nos anexos I e J e aspectos ligados à usabilidade, à acessibilidade, à segurança e à optimização dos diversos componentes do protótipo.

No âmbito da pesquisa de objectos de aprendizagem foi sugerida a execução de determinadas tarefas de pesquisa e, observando o modo como o utilizador interagia com o protótipo, foram detectadas as dificuldades que sentia, o que lhe agradava ou desagradava e o seu grau de satisfação com os resultados devolvidos. A insatisfação nos resultados derivada da baixa expressividade semântica inicial foi sendo iterativa e incrementalmente reduzida até se alcançar um protótipo final que na generalidade foi bem aceite.

No que diz respeito ao grau de satisfação com os resultados devolvidos, apresentamos seguidamente e de forma sucinta um exemplo de actividade de pesquisa.

Consideremos a actividade de pesquisa: "Quais os objectos de aprendizagem do tipo *lição* acerca de *usabilidade* publicados pelo formador *Vitor Gonçalves*, enquanto professor da *ESE-IPB*?". Vamos apresentar as correspondentes tarefas de alteração e optimização do sistema com vista a clarificar o incremento no grau de satisfação.

Embora existam vários documentos onde aparece o termo "usabilidade", apenas quatro estão descritos com o assunto ou palavra-chave "usabilidade" e destes, só dois são lições. Mas apenas um recurso corresponde à pesquisa: a lição designada "Abordagem Ergonómica de Sistemas" com o uri ".../ergo1.html".

A pesquisa foi submetida a três níveis: nível sintáctico, nível estrutural e nível semântico.

Nível sintáctico:

Os alvos da pesquisa eram os ficheiros de metadados gerados automaticamente em formato RDF/XML. Para realizar a consulta a nível sintáctico recorremos à linguagem de consulta *XQuery*. Não obstante, a codificação desta consulta tornou-se extremamente complexa, uma vez que o modelo RDF é um grafo e não uma árvore hierárquica. Por conseguinte, embora fosse possível através da sintaxe XML recuperar os LOs que continham o assunto "usabilidade" ou todas as lições, ao tentar filtrar com "Vitor Gonçalves" tornou-se complexo garantir que todos os LOs devolvidos correspondessem aos recursos leccionados por esse professor ou que o resultado incluísse todos os LOs do modelo RDF. Para além deste inconveniente, foi impossível lidar com a relação entre a instituição e o docente.

Nível estrutural:

Uma vez que um documento RDF representa um conjunto de triplas, cada tripla representando uma declaração na forma de *Subject-Predicate-Object* (Davies et al., 2003), uma linguagem para inferir documentos RDF, tal como a linguagem RDQL ou SPARQL, permite-nos consultar os LOs que têm um determinado assunto e uma determinada subclasse. Ou seja, se "Abordagem Ergonómica de Sistemas" é do tipo "licao" e se "licao" é subclasse de "LO", então o uri ".../ergo1.html" também é do tipo LO.

```
SELECT ?x
FROM ficheiro rdf
WHERE (rdf::type ?xLO)
```

Essa semântica pode estar representada em RDFS mas se não houver uma tripla explícita a inferência falhará. Uma vez que a tripla (…/ergo1.html, type, licao) não é explícita, não conseguimos obter o LO em questão. Uma alternativa válida pode ser:

```
SELECT ?x
FROM ficheiro rdf
WHERE (rdf::type ?x ?classe),
      rdfs::subClassOf ?classe LO)
OR    ?classe ~ LO
```

No entanto, isto apenas resolve este problema específico. Não soluciona o problema em geral. Portanto, para além da inferência das declarações RDF relativas ao nível estrutural é necessário inferir o conhecimento inerente ao RDFS (nível semântico).

Nível semântico:

Tomando como referência a ontologia do sistema WSE (ou a ontologia do sistema de e-Learning), é possível inferir semanticamente as subclasses de "LO", obtendo a classe "licao" e todas as suas instâncias. Da mesma forma é possível inferir as subclasses de

"Metadados", obtendo a classe DCM e LOM e as suas instâncias filtradas com base na propriedade "subject" ou "keywords", respectivamente. Igualmente, para a instância "Vitor Gonçalves" da subclasse "Formador" da classe "Participante" e para a instância "ESE-IPB" da classe "Entidade", tal como podemos ver parcialmente nas seguintes linhas SPARQL:

```
SELECT Y, $Y, K, $K
FROM licao{X} . publicado_por{Y : $Y} . Professor{Z} . integrado_em{K, $K}
```

Genericamente, este exemplo aplica-se tanto ao sistema WSE como aos sistemas de e-Learning. A única diferença é a dimensão da base de objectos de aprendizagem sobre a qual é efectuada a pesquisa, uma vez que no sistema de e-Learning estamos limitados aos objectos de aprendizagem desse servidor.

A ontologia OWL apresentada resumidamente no anexo H permite que inferências ainda mais complexas possam ser configuradas através dos agentes de informação desenvolvidos para o sistema WSE ou do bloco de pesquisa semântica desenvolvido para os sistemas de e-Learning Moodle.

A fase de avaliação do protótipo final decorreu durante o primeiro semestre de 2006/2007 e baseou-se essencialmente na observação directa, no acompanhamento *on-line* e na realização de inquéritos aos utilizadores actuais do sistema easy-Learning da ESE-IPB. Em Fevereiro de 2007, este sistema de b-Learning contava já com 100 disciplinas, 487 formandos inscritos e 22 formadores leccionando disciplinas diversas dos cursos de licenciatura (plano antigo e plano de acordo com Bolonha), disciplinas de um curso de especialização tecnológica e acções de formação contínua de professores.

Os resultados de avaliação que a seguir se apresentam dizem respeito à avaliação do sistema easy-Learning e resultaram essencialmente dos inquéritos apresentados no anexo K.

Dos 22 formadores, 12 preencheram e participaram activamente na avaliação e validação do Sistema easy-Learning, mais concretamente dos módulos e bloco de pesquisa. Enquanto que dos 487 formandos, apenas 53 participaram nesta avaliação.

Dos 12 formadores apenas 4 afirmaram ter utilizado outra plataforma de e-Learning antes de usar o easy-Learning, 2 como formadores (*Luvit* e *Atutor*) e 2 como formandos (plataformas proprietárias). Enquanto que dos 53 formandos apenas 2 haviam usado antes outras plataformas de e-Learning (plataforma IPB e plataforma *Cisco Systems*).

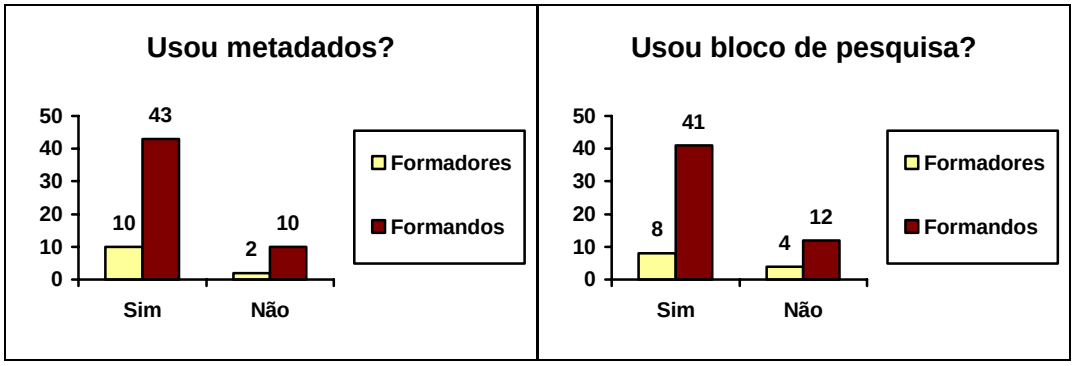
Dos 12 formadores, 50% afirmaram ter alguma experiência ao nível da catalogação e indexação de recursos; todavia nenhum deles conhecia qualquer norma ou esquema de

metadados. Com o easy-Learning constatámos que 10 formadores usaram os módulos de metadados para mais do que 5 objectos de aprendizagem, perfazendo um total de 498 registos de metadados em 1608 recursos e actividades (LOs). Questionados sobre a modalidade de geração de metadados, a preferência dos formadores recaiu sobre módulos com geração (semi)automática de metadados, permitindo edição por parte do autor.

A baixa participação na avaliação do sistema justifica-se, no nosso entender, com três factores principais: o sistema de b-Learning é experimental no âmbito de um projecto de investigação, não tendo sido enquadrado no plano estratégico da instituição; o desempenho do servidor degrada-se significativamente com mais de 10 utilizadores simultâneos; e a grande maioria dos formadores e formandos teve só agora o seu primeiro contacto com um sistema de e-Learning, encontrando-se ainda numa fase de aprendizagem da plataforma e de percepção dos novos desafios com os quais são confrontados.

Numa escala de 1 a 5 (1 = nada importante; 2 = pouco importante, 3 = importante; 4 = muito importante; 5 = fundamental), apresentamos os resultados mais relevantes:

1) Já utilizou os módulos de metadados para descrever (ou obter informação) sobre os objectos de aprendizagem de uma disciplina? E o bloco de pesquisa (Procurar LO-metadados)?



2) Quanto à importância dos módulos de metadados e do bloco de pesquisa para a satisfação dos utilizadores com o Sistema Moodle easy-Learning, destaca-se a seguinte avaliação dos factores:

Factores	Formador	Formando
A existência de um bloco de pesquisa para procurar conteúdos: Procurar LO-metadados	4.8	4.0
A existência de metadados para descrever e complementar os conteúdos	4.5	3.9
O módulo “Sumários de Aulas”	4.7	4.2
O módulo “Programa da Disciplina”	4.8	4.3

3) Quanto à importância dos módulos de metadados no Sistema Moodle easy-Learning, destaca-se o seguinte:

Factores	Formador	Formando
A eficácia dos módulos ao permitir a pesquisa de objectos de aprendizagem	4.5	4.0
A eficácia dos módulos na descrição dos objectos de aprendizagem	4.5	4.0
A visibilidade dos metadados de um objecto de aprendizagem mesmo que não tenha autorização para ver esse objecto de aprendizagem	4.0	3.6
A visibilidade dos metadados juntamente com os objectos de aprendizagem de uma disciplina (na primeira página da disciplina).	3.8	3.9

4) No que diz respeito à importância do bloco de pesquisa, destaca-se que:

Factores	Formador	Formando
Pesquisa de objectos de aprendizagem nas minhas disciplinas	4.5	4.3
Pesquisa de objectos de aprendizagem em todas as disciplinas do easy-Learning	4.2	3.8
Resultados devolvidos (os objectos devolvidos normalmente correspondem à pesquisa solicitada)	4.2	4.0

5) Relativamente à utilidade dos metadados para os intervenientes do processo educativo:

Factores	Formador	Formando
Úteis para perceber a que se referem os objectos de aprendizagem devolvidos pelo bloco de pesquisa	3.9	4.2
Úteis para suportar a tomada de decisão de potenciais alunos (obter informação sobre as temáticas leccionadas nas disciplinas para eventualmente adicionar essa disciplina ao seu plano de estudos)	3.8	4.0
Úteis para saber que conteúdos são leccionados em disciplinas nas quais não estou inscrito	4.0	4.0
Úteis para cruzar informação sobre os temas, conteúdos e conceitos leccionados nas várias disciplinas de um curso (úteis para definir estratégias de interdisciplinaridade)	4.4	4.1
Úteis para complementar ou esclarecer dúvidas sobre os objectos de aprendizagem	4.5	4.3

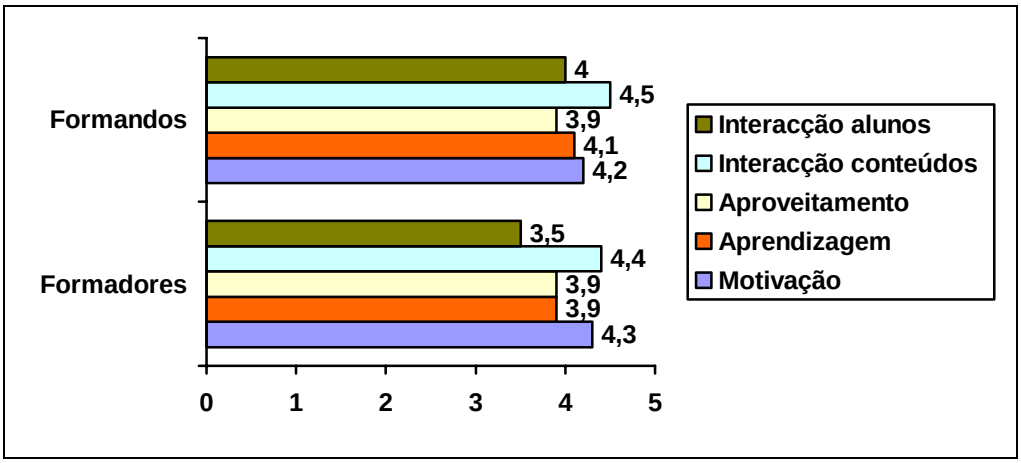
6) Quando comparado com o ensino presencial, o Sistema Moodle easy-Learning influenciou positivamente os alunos no que diz respeito ao seu nível de:

a) Interacção com os conteúdos e com outros alunos;

b) Motivação;

c) Aprendizagem;

d) Aproveitamento;



Globalmente, podemos concluir que o sistema de e-Learning correspondeu satisfatoriamente às expectativas, em boa parte fruto da avaliação do primeiro protótipo e do constante apoio aos formadores. Para além de munir o sistema de e-Learning de um mecanismo de pesquisa para recuperar qualquer objecto de aprendizagem, disponibiliza informação sobre os recursos e actividades (metadados) e permite realizar inferências sobre esses metadados e a ontologia do sistema.

No que diz respeito ao sistema WSE convém referir que não foi possível disponibilizar a versão do protótipo final à comunidade da ESE-IPB, uma vez que a plataforma Voyager requer uma licença relativamente dispendiosa. Contudo, foram efectuados testes para validar o protótipo final com uma licença de avaliação para fins educativos disponibilizada pela *Recursion Software*.

Quando comparado com as pesquisas realizadas remotamente, este tipo de pesquisa baseada em agentes móveis pode ser 100 a 10000 vezes mais rápida, dependendo do tráfego existente na rede e dos equipamentos remotos, de acordo com indicações da *Recursion Software* (Recursion, 2003). Para avaliar esta situação decidimos medir o tempo gasto numa pesquisa de todos os recursos de "Vitor Gonçalves" em 2 servidores integrados no sistema WSE (30 registos anotados com esse nome no elemento *creator* ou *entity*), efectuando 4 tipos de pesquisa distintos:

a) Pesquisa individual usando o bloco de pesquisa SQL em cada uma das BDs locais dos sistemas de e-Learning através de SQL;

b) Pesquisa individual em cada servidor através do *SPARQL Server Joseki*;

c) Pesquisa baseada nos metadados armazenados nas BD remotas através do sistema WSE;

d) Pesquisa baseada nos metadados e ontologia armazenados nos discos remotos através do sistema WSE.

Os valores médios obtidos nas 3 tentativas para cada uma das situações foram:

a) 2982 milisegundos;

b) 1885 milisegundos;

c) 3092 milisegundos;

d) 2105 milisegundos.

Conclui-se, portanto, que à medida que o volume de registos e o número de servidores registados no sistema aumentam, o servidor WSE obtém resultados ainda melhores, já que o tamanho do agente de pesquisa não ultrapassa os 4 *Kbytes*. Para além disso, note-se que o processamento da pesquisa através do servidor WSE ocorre quase em simultâneo em vários servidores, uma vez que após entregar o pedido ao agente de informação, o agente de pesquisa segue o seu caminho para o próximo servidor remoto, não necessitando de esperar pelos resultados. Aliás, convém referir que os 3 e 2 segundos gastos pelo servidor de agentes correspondem ao tempo total gasto, mas os primeiros resultados foram entregues ao cliente em menos de 1 segundo.

A medição destes tempos foi efectuada automaticamente pelo código das linguagens de programação usadas. Por exemplo, no caso do php foi usada uma classe disponível na documentação on-line do website php.net:

```
class stopwatch
{
    private $round     = 3;
    function __construct ( )
    {
        $this->start = microtime();
    }
    function now()
    {
        $start     = $this->math($this->start);
        $now     = $this->math();
        return round($now - $start, $this->round);
    }
    function math ($time = FALSE)
    {
        if ( !$time ) $time = microtime();
        $temp = explode(' ', $time);
        return $temp[0] + $temp[1];
    }
}
```

Cada um dos ficheiros responsáveis pela pesquisa de objectos de aprendizagem inclui no topo e no fundo do ficheiro o seguinte código php:

```
$stopwatch = new stopwatch();
//código do ficheiro responsável pela pesquisa
echo $stopwatch->now();
```

De referir que estes testes decorreram durante o horário normal de funcionamento e no contexto da intranet da ESE-IPB, à qual pertenciam os dois servidores de e-Learning, o servidor do Sistema WSE e o computador cliente, cuja largura de banda é de 100 Mbps.

7.6- Resumo

Este capítulo apresentou as características de implementação, instalação e avaliação mais relevantes do protótipo do sistema proposto, de acordo com as prioridades de implementação que foi necessário definir.

Primeiramente, foram descritos os principais componentes do protótipo e respectivas funcionalidades e as características de implementação e instalação mais importantes. Posteriormente, e uma vez que os agentes de informação residem nos servidores remotos, tornou-se necessário preparar os sistemas externos ou remotos com vista à sua integração na infra-estrutura distribuída. Neste sentido, foram descritas as actividades de programação e instalação dos sistemas de e-Learning e outros sistemas remotos que disponibilizam objectos de aprendizagem devidamente descritos através de metadados e ontologias.

As actividades de implementação e instalação recorreram a diversas tecnologias: umas mais estáveis, como são os casos da linguagem Java, do acesso a bases de dados, da computação de objectos distribuídos e dos sistemas de agentes móveis; e outras ainda em fase de alguma instabilidade, como são os casos das tecnologias para os metadados, para as ontologias e para a inferência.

Finalmente, foram apresentados alguns dos principais resultados referentes à avaliação do protótipo e dos módulos e do bloco de pesquisa semântica desenvolvidos para um sistema de e-Learning particular, o Moodle.

8- Considerações Finais

Cientes de que o futuro da Web passa pela Web Semântica e de que a Educação pode ser um dos campos que mais proveito poderá tirar desta evolução, neste capítulo apresentamos uma análise global do trabalho realizado e dos resultados decorrentes da avaliação e validação dos protótipos, para além de efectuarmos uma comparação com alguns projectos que surgiram recentemente e de perspectivar o trabalho futuro.

Este projecto de doutoramento enquadrou-se no âmbito dos Sistemas Distribuídos para a Web, da Tecnologia Educativa e da Gestão da Informação e do Conhecimento aplicados à Web, mais concretamente, no contexto do e-Learning e da Web Semântica. Pretendeu-se realçar o papel que esta nova visão da Web pode originar nos Websites educativos, e mais concretamente nos sistemas de e-Learning, tendo em vista a melhoria dos mecanismos de pesquisa de recursos de informação em geral e de objectos de aprendizagem em particular, na Internet.

A primeira geração da Internet preocupou-se com a implementação da infra-estrutura tecnológica desta rede de computadores e dos serviços respectivos. No âmbito desta dissertação interessou-nos essencialmente a evolução do serviço Web. A segunda incidiu na construção de aplicações Web. Para além do conteúdo estático, a Web passou também a disponibilizar páginas dinâmicas geradas a partir de base de dados, incluindo os ambientes virtuais de aprendizagem. Actualmente, a terceira geração avança rumo a uma Web mais inteligente, apostando na migração de uma Web de documentos interligados através de links para uma Web de Dados relacionando os conceitos para fornecer tanto aos humanos como às máquinas a semântica para realizar inferências mais precisas sobre os dados. Referimo-nos à iniciativa Web Semântica. Entretanto, outras transições vão sendo percebidas. Actualmente, destaca-se a que tem vindo a ser apelidada de Web 2.0 – termo associado à mais recente transição percebida na Web em 2004: passamos de uma colecção de Websites para plataformas de aplicações Web (blogs, wikis e redes sociais como o *MySpace* ou aplicações de “inteligência colectiva”).

O crescimento da Web e a interiorização do conceito de educação ao longo da vida têm vindo a impulsionar novas formas de auto-aprendizagem, nomeadamente: a aprendizagem através de conteúdos organizados e disponibilizados por sistemas de e-

Learning ou outros ambientes de aprendizagem Web e a aprendizagem através de conteúdos Web dispersos localizados por motores de busca. O crescimento da Web proporciona a professores, alunos e cidadãos em geral um vasto conjunto de oportunidades de aprendizagem ou de geração de novo conhecimento que nem sempre é possível concretizar dadas as dificuldades de localização das fontes de informação úteis, uma vez que a representação do saber e a forma dos motores de busca a procurarem assenta na sintaxe, deixando a análise semântica para o utilizador.

8.1- Retrospectiva do Trabalho Realizado

Tanto os consórcios e grupos de investigação em torno da visão da Web Semântica, como os relacionados com o e-Learning, têm vindo a criar condições no sentido de construir os alicerces de um inovador espaço de conhecimento e de aprendizagem. Os últimos seis anos foram frutíferos em normas, especificações, recomendações e propostas de tecnologias e linguagens para a representação e estruturação da informação e do conhecimento, para a descoberta e recuperação de informação e para o suporte à (auto-)aprendizagem. A tecnologia de agentes de software tem vindo também a evoluir consideravelmente e, entre muitas outras áreas de aplicação, é já usada na criação de ambientes distribuídos com vista a recuperar informação de bases de dados heterogéneas. O estado da arte e os fundamentos das três áreas de investigação basilares desta tese foram descritos nos capítulos II, III, IV e V.

No capítulo II apresentámos as tecnologias e especificações mais adequadas para desenvolver, partilhar e reutilizar conteúdos de e-Learning: o modelo SCORM e a filosofia dos objectos de aprendizagem. Actualmente, reconhece-se que o uso de objectos de aprendizagem é muitas vezes conotado com o acto de folhear um livro, levando os alunos a ter a sensação de que estão perante conteúdos estáticos e que, consequentemente, provocam a sua desmotivação. Neste sentido, o *Learning Design* assume-se como a forma de minimizar estas limitações já que coloca ênfase na personalização dos conteúdos e promove uma participação mais activa do aluno no processo de aprendizagem, através da modelação de unidades de aprendizagem.

De forma abstracta e omitindo alguns pormenores técnicos para facilitar a compreensão das funções e características fundamentais das tecnologias para a Web Semântica, podemos sintetizar a arquitectura para a Web Semântica em quatro camadas principais: camada estrutural, camada sintáctica, camada semântica e camada lógica. Assumindo que a tecnologia *Resource Description Framework* implementa a camada central da arquitectura para a Web Semântica, os capítulos III e IV abordaram as tecnologias das camadas mais baixas (estrutural e sintáctica) e as tecnologias das camadas mais altas (semântica e lógica) da arquitectura para a Web Semântica, respectivamente.

Genericamente, concluímos que as duas alternativas de auto-aprendizagem referidas (auto-aprendizagem através de sistemas de e-Learning e auto-aprendizagem através de motores de busca) beneficiam de uma visão segundo objectos de aprendizagem, devidamente descritos através de metadados, com significados semanticamente explícitos e baseados em ontologias não só porque permitem procuras mais precisas desses objectos, mas também porque favorecerem a sua reutilização e interoperabilidade.

Os agentes devem ser capazes de analisar o ambiente em que se inserem e reagir ou responder às alterações ocorridas neste, segundo uma base de conhecimento, isto é, seguindo um conjunto de regras previamente definidas ou entretanto apreendidas, com o intuito de comunicar com o utilizador ou interagir com outros agentes. As ontologias e as regras de inferência podem constituir a base de conhecimento indispensável à actuação dos agentes de software, tal como explicado anteriormente no capítulo V. Por conseguinte, a visão da Web Semântica traduzirá os conteúdos Web numa estrutura com significado e definirá a forma de lidar com essa estrutura, criando um ambiente propício para os agentes de software. Ora, se os agentes de pesquisa possuírem mobilidade suficiente para migrarem de um ambiente de aprendizagem para outro, então poderemos utilizá-los na construção de um mecanismo de busca baseado em metadados e ontologias, com vista a facilitar o acesso, descrição, localização e reutilização dos conteúdos educativos, para além de reduzir o tráfego de dados na rede e aumentar o desempenho, passando o processamento das pesquisas ou consultas de informação para o lado dos servidores remotos.

Os capítulos VI e VII deste documento ilustraram as principais actividades do projecto de investigação:

- A identificação e especificação dos requisitos e a correspondente modelação da "Arquitectura do sistema WSE" (modelo para um sistema de recuperação de objectos

de aprendizagem para a Web Semântica, denominado WSE - Web Semântica Educativa), tendo por base tanto a comunidade educativa do Ensino Básico e Secundário como a comunidade educativa da ESE-IPB;

- Especificação e desenvolvimento de um protótipo experimental e a correspondente descrição das funcionalidades principais do mesmo, para além da preparação de supostos sistemas externos remotos fornecedores de objectos de aprendizagem ou de cursos de formação, nomeadamente sistemas de e-Learning Moodle, sistemas de gestão de conteúdos e página do professor para a Web Semântica.

Concretamente, pretendíamos avaliar se a utilização de metadados para descrever objectos de aprendizagem e o recurso a ontologias para orientar a localização e recuperação de objectos de aprendizagem permitia pesquisas mais rápidas, precisas e em conformidade com as necessidades do utilizador.

Assim, a arquitectura proposta resultou num protótipo acessível através da Web para a recuperação de objectos de aprendizagem armazenados em sistemas de e-Learning, repositórios de objectos de aprendizagem e noutros servidores Web de conteúdos educativos descritos semanticamente. O ambiente do protótipo apresenta as seguintes características:

- Distribuído: os sistemas externos podem estar localizados em áreas geográficas dispersas;
- Aberto: o sistema WSE aceita a integração de novos componentes, tais como outros servidores de agentes ou outros sistemas externos;
- Modular: as diversas funcionalidades que compõem o sistema estão organizadas em módulos que comunicam entre si, o que permite uma fácil expansão do sistema;
- Transparente: a pesquisa de objectos de aprendizagem remotos decorre de forma transparente para o utilizador e independente da linguagem de programação e dos sistemas operativos dos sistemas externos remotos.

Ao contrário dos módulos implementados no âmbito do sistema de e-Learning Moodle, o protótipo experimental do sistema WSE não foi devidamente validado, por falta de licença da plataforma Voyager. No entanto, dos testes de validação e avaliação que foi possível realizar, destacam-se os seguintes aspectos positivos relativos à validade e ao desempenho deste sistema na pesquisa de objectos de aprendizagem:

1) O facto da pesquisa de LOs ser orientada com base em duas camadas semânticas, uma baseada na anotação semântica associada aos sistemas remotos, armazenada na BD do sistema, e que permite definir o percurso para o agente de pesquisa, e a outra baseada na pesquisa semântica sobre os metadados armazenados nas bases de dados locais ou na inferência semântica sobre documentos RDF(S)/OWL armazenados num directório local de cada servidor remoto. A primeira permite que nem todos os sistemas façam parte do roteiro do agente de pesquisa, uma vez que o utilizador pode filtrar a pesquisa pelo tipo de sistemas, pelo tipo de LOs, pelo nível de aprendizagem das disciplinas, pelo tipo de metadados a pesquisar, pelo tipo de inferência a realizar (nos metadados armazenados na BD local e/ou na pasta de metadados local) pelo agente de informação, bem como por outros parâmetros. A segunda permite definir, por exemplo, qual ou quais os elementos de metadados sobre os quais irá incidir a pesquisa, quando o agente de pesquisa chegar a cada um dos destinos do seu percurso. Desta forma, conseguimos não só acelerar o processo de pesquisa, mas também obter resultados mais precisos e semanticamente correctos.

2) A utilização de agentes de pesquisa móveis, já que o processamento da pesquisa passa para o lado dos servidores remotos, sendo efectuada nas BDs locais e nos documentos RDF/XML pelos agentes de informação dos servidores remotos. Uma vez que não é da responsabilidade do agente móvel a devolução dos resultados, o agente ganha em mobilidade visto que não tem que transportar informação, a rede ganha em largura de banda dado que o processamento ocorre do lado dos servidores remotos e o utilizador ganha em rapidez na obtenção de resultados porque pode ir visualizando a informação à medida que esta vai sendo enviada pelos agentes de informação, não tendo que aguardar pelo regresso do seu agente de pesquisa.

3) Desenvolvimento de cursos de formação de elevada granularidade: pouco a pouco os professores têm vindo a mentalizar-se de que cursos de formação compostos por documentos de tamanho grande (pdf ou ppt), disponibilizados através do sistema de e-Learning Moodle, não são solução para o processo de aprendizagem, nem para a recuperação e reutilização de termos ou conceitos mais específicos. Por um lado, os protótipos do bloco de pesquisa e dos módulos de metadados, de sumários e de programas contribuíram indirectamente para incrementar a granularidade das disciplinas ou cursos de formação, beneficiando assim a recuperação de LOs sobre um determinado conceito específico. Por outro, estes módulos e o bloco permitiram que alunos e professores

tivessem acesso a informações sobre os LOs para suportar a tomada de decisão relativa à escolha das disciplinas por um aluno ou inerente a processos de interdisciplinaridade que um professor decida implementar, para além de impulsionar o auto-estudo, desde que o acesso aos LOs não esteja bloqueado. De salientar também o facto de que a integração da adição de metadados automática com a criação do próprio recurso (mais concretamente, com a criação do programa da disciplina e a criação de sumários das aulas) impulsionou gradualmente a actividade de refinamento dos metadados gerados por parte dos formadores, o que se traduziu numa melhoria significativa dos resultados de pesquisa e da inferência baseada na ontologia.

Por conseguinte, a arquitectura proposta é nitidamente um contributo não só para tornar a Web mais visível, mas também para permitir localizar e recuperar LOs (ou apenas informação sobre esses LOs) de forma semântica, ou seja, garantindo que a pesquisa de conteúdos educativos seja baseada em palavras-chave providas de significado e de contexto.

8.2- Comparação com outros Trabalhos ou Projectos

Ao longo deste documento foram apresentados alguns projectos, nomeadamente no âmbito do estado da arte de cada uma das tecnologias ou temáticas. No contexto específico da Educação têm vindo a surgir cada vez mais trabalhos e projectos baseados no alinhamento das tecnologias para a Web Semântica com as tecnologias para o e-Learning. Porém, muitos deles fazem uso apenas das tecnologias relativas às camadas mais baixas da arquitectura da WS, limitam-se a situações específicas ao nível de uma Intranet ou não têm propriamente aplicabilidade no contexto educativo e, raramente, o nível semântico é rico em expressividade. Com esta afirmação não pretendemos dizer que esses trabalhos ou projectos não cumprem os objectivos que orientaram o seu desenvolvimento, apenas tencionamos reduzir o número de iniciativas comparáveis com a arquitectura aqui proposta. Outros projectos e trabalhos têm surgido no espaço europeu, impulsionados por entidades como o JISC (*Joint Information Systems Committee*) (JISC, 2006) ou financiados pelo programa IST (*Information Society Technology*) (IST, 2006). Contudo, muitos deles encontram-se em fase embrionária de planeamento ou desenvolvimento.

Os projectos educativos que destacámos anteriormente, por fazerem uso das tecnologias para a Web Semântica para disponibilizar objectos de aprendizagem e/ou permitir a sua localização e recuperação, foram: ARIADNE, GEM, SMETE, EDUTELLA e ELENA.

Os projectos ARIADNE, GEM e SMETE são projectos que têm algumas semelhanças com o sistema WSE, já que têm como finalidades principais a indexação, a partilha e a reutilização de LOs.

Para tal, o ARIADNE disponibiliza o *ARIADNE Knowledge Pool System* que tem como principal objectivo partilhar e reutilizar LOs através da sua anotação automática com metadados (funcionalidade disponibilizada pela recente API SAmgI), para além do ARIADNE *Web-Based Learning Environment* (WebLE) que permite o desenvolvimento dos LOs através de ferramentas de autor.

O GEM é como uma porta de acesso a milhares de recursos educativos previamente anotados com metadados. Recentemente, o GEM passou também a usar RDF/XML para descrever os recursos educativos, para além dos *profiles* GEM e dos esquemas LOM e DCM expressos em XML.

O SMETE disponibiliza um conjunto de funcionalidades para o armazenamento, catalogação, partilha e pesquisa de recursos educativos. As tecnologias para a catalogação e indexação de LOs e correspondente anotação baseiam-se nos metadados *IMS-compliant meta-data*. Com um conjunto de classes Java e a linguagem WSDL (*Web Services Description Language*) foi desenvolvida a *SMETE Search API*, baseada no protocolo SOAP, que tem como principal função pesquisar LOs através dos seus metadados.

Comparando com a infra-estrutura distribuída da WSE, a diferença mais notória é que todos os projectos referidos se baseiam na arquitectura Cliente-Servidor, pelo que a menor exigência em termos de largura de banda e a possibilidade de descobrir e recuperar recursos em vários servidores remotos simultaneamente constituem claramente vantagens a favor do Servidor WSE.

O EDUTELLA é o projecto que mais se assemelha com a proposta que aqui se apresenta, uma vez que nos últimos anos tem vindo a usar as tecnologias para a WS na sua infra-estrutura P2P, disponibilizando as seguintes funcionalidades:

- Serviço de consulta: Consulta e recuperação de metadados expressos em RDF através da linguagem QEL (*Query Exchange Language*).

- Serviço de anotação: Anotação dos materiais educativos armazenados em qualquer lugar na rede Edutella.
- Serviço de mapeamento: Tradução entre diferentes vocabulários de metadados para permitir a interoperabilidade entre diferentes nós da rede.
- Serviço de Replicação: Fornece persistência/disponibilidade dos dados e balanceamento dos mesmos enquanto mantém a sua integridade e a consistência.

Paralelamente ao planeamento e desenvolvimento deste projecto de investigação, surgiu uma iniciativa europeia no âmbito do programa IST: o projecto ELENA, cuja finalidade foi criar *Smart Spaces for Learning*. Dos vários estudos de caso desenvolvidos no âmbito deste projecto destaca-se o SQI (*Simple Query Interface*), um esforço aberto e colaborativo, impulsionado pelo *CEN/ISSS Learning Technologies Workshop* e em cooperação com a rede *PROLEARN Network of Excellence*, para conseguir a interoperabilidade entre repositórios de LOs. Como resultados destacamos a arquitectura para a personalização de serviços na Web Semântica Educativa, nomeadamente através do desenvolvimento de um conjunto de esquemas RDFS (*learner performance*, *portfolio*, *preferences*, *goals*, *history*, *learner assessment*, *competencies* e *privacy*). Este caso de estudo baseou-se no uso do SQI para ligar repositórios heterogéneos, tais como Amazon.com, ambientes de formação e de aprendizagem *on-line*, entre outros.

Nestes últimos meses surgiram mais alguns projectos que pretendem aplicar as tecnologias para a Web Semântica no contexto educativo ou contextos próximos, dos quais destacamos dois projectos que, embora em fase embrionária, apresentam maiores similaridades com a arquitectura do sistema WSE: LUISA (*Learning content management system Using Innovative Semantic web services Architecture*) (LUISA, 2006) e MOSAICA (*Semantically Enhanced Multifaceted Collaborative Access to Cultural Heritage*) (MOSAICA, 2006).

A missão do projecto LUISA é explorar as vantagens de uma arquitectura baseada em *Semantic Web Services* para tornar o processo de pesquisa mais rico e flexível e permitir a especificação das necessidades de aprendizagem no contexto dos LMSs e dos repositórios de LOs. Esta arquitectura pretende abrir novas possibilidades na automatização da descoberta, da selecção e da composição de LOs, dentro das arquitecturas distribuídas devidamente integradas através de ontologias.

Embora não seja um projecto de índole educativa, mas sim cultural, decidimos incluir também nesta secção o projecto MOSAICA face às suas similaridades com alguns componentes do sistema WSE. O MOSAICA é um projecto também em fase embrionária que permite o acesso colaborativo a recursos culturais anotados semanticamente pelos diversos membros do sistema. Pelo que se consegue perceber da parca informação descritiva no Website do projecto, para atingir os objectivos definidos a arquitectura do sistema resultará da integração dos módulos de anotação semântica baseados em metadados e ontologias criados colaborativamente com os módulos de pesquisa e navegação semântica nos recursos devolvidos. Portanto, a arquitectura deste sistema poderá ser muito parecida à arquitectura distribuída que apresentamos, a menos que os recursos e sua anotação residam no sistema, em vez de estarem dispersos pelos servidores remotos registados no sistema.

Por conseguinte, a abordagem que seguimos para aplicar a Web Semântica ao contexto educativo tem vindo a ser também usada paralelamente pelas equipas de planeamento e de desenvolvimento dos mais recentes projectos.

Para além dos projectos que usam as tecnologias para a Web Semântica aplicadas no contexto educativo, não podemos deixar de referir também três projectos nacionais: *Metamorphosis* (um ambiente de processamento de *topic maps*, propiciando extracção automática, validação e geração de navegadores conceptuais sobre um universo de discurso), *Adrian* (gerador de interfaces que, recebendo uma ontologia especificada com *topic maps*, produz um conjunto de páginas Web que permitem a navegação na rede conceptual especificada e aceder aos recursos de informação apontados pelas ocorrências no TM) e Servidor de Agentes (modelo baseado em agentes móveis para procura de informação em sistemas distribuídos). Os dois primeiros relacionam-se com a aplicação de mapas de tópicos e resultaram em dissertações submetidas à Universidade do Minho, o outro relaciona-se com a aplicação das tecnologias de agentes para a recuperação de informação multimédia a partir de BD heterogéneas e resultou numa dissertação submetida à Universidade de Trás-os-Montes e Alto Douro.

Embora fosse necessário rever algumas questões técnicas, parece-nos que *Metamorphosis* (Librelloto, 2005) ou *Adrian* (Adrian, 2004) poderiam ser integrados com o sistema WSE no âmbito dos utilitários para a edição de mapas de tópicos e respectiva navegação semântica.

O modelo do servidor de agentes é muito parecido com o modelo do servidor de agentes móveis do sistema WSE. Aliás, as bases dessa componente assentaram na descrição apresentada na tese de dissertação correspondente. No entanto, esse componente apenas permitia a pesquisa de informação multimédia através do tipo de informação e palavras-chave, ou seja, não permitia a pesquisa semântica de todos os tipos de LOs baseada em metadados e ontologias. Portanto, esta é uma das inovações do presente projecto de doutoramento.

8.3- Perspectiva do Trabalho Futuro

O objectivo da arquitectura proposta e do respectivo protótipo era não só contribuir para a adopção de um modelo de estruturação, representação e procura de conteúdos educacionais, mas também abrir caminho para outros trabalhos que tenham como objectivo contribuir para a criação de uma infra-estrutura que integre as tecnologias subjacentes ao e-Learning e à Web Semântica.

Tal como referimos anteriormente, os utilitários para edição e navegação em ontologias e mapas de tópicos no âmbito do servidor de agentes móveis ainda estão numa fase de implementação inicial. A edição de ontologias OWL *on-line* deverá também incluir a possibilidade de definir regras de inferência SWRL. Portanto, um dos primeiros trabalhos a realizar será concluir o desenvolvimento desses módulos e proceder à sua validação e avaliação.

Seria também desejável que os LOs devolvidos como resultado de uma pesquisa fossem enquadrados em pelo menos um dos mapas de tópicos existentes no sistema WSE, ou seja, fossem incluídos como links dos conceitos ou tópicos descritos em XTM, agrupando-os num curso personalizado navegável semanticamente. Contudo, a integração automática dos LOs nos mapas de tópicos pode traduzir-se em cursos de formação pedagogicamente incoerentes. Este requisito poderá constituir por si só um desafio para outros trabalhos de investigação.

Intencionalmente, o mecanismo de pesquisa possui duas formas de pesquisa alternativas e mesmo complementares: pesquisa baseada nos metadados armazenados na BD e pesquisa baseada nos metadados armazenados nos documentos RDF/XML que orientam a pesquisa semântica através da ontologia do sistema. Esta situação deveu-se essencialmente a dois factores. Por um lado, nem sempre os docentes adicionam metadados aos recursos e actividades do Moodle, pelo que a pesquisa continuaria a funcionar sobre, pelo menos, 2 campos (título e resumo). Por outro, necessitávamos de uma forma intermédia para comparar a pesquisa desprovida de semântica com a pesquisa semântica. Não obstante, podemos ajustar esta funcionalidade recorrendo à *framework* Jena para permitir realizar inferências semânticas sobre as BD locais dos sistemas externos.

No que diz respeito aos módulos e ao bloco desenvolvidos para o sistema de e-Learning Moodle, a optimização dos mesmos com o recurso à tecnologia AJAX (*Asynchronous JavaScript and XML*) permitiria muni-los de maior interactividade. Para além disso, seria também necessário num futuro muito próximo munir a extracção de palavras-chave de um algoritmo mais eficiente e eficaz.

Uma alternativa para colmatar a falta de metadados nos sistemas remotos poderia passar pela melhoria do módulo de anotação semântica, passando a realizar a anotação semântica de todos os LOs do lado do servidor de agentes, tal como já acontece com a anotação semântica de cursos. Desta forma, poderíamos filtrar ainda mais o percurso do agente e, consequentemente, melhorar o processo de pesquisa em termos de precisão. No entanto, esta solução retiraria parte da rapidez que caracteriza a actual arquitectura, uma vez que passaria a existir um motor de inferência centralizado no servidor de agentes móveis, em detrimento da descentralização que o modelo actual oferece.

Finalmente, o sistema WSE poderia ser aplicado como um sistema de e-Learning para o Ensino Secundário, uma vez que a avaliação do sistema permitiu concluir que o servidor de agentes se poderia comportar como um sistema de e-Learning Moodle integrando diversos sistemas de e-Learning Moodle distribuídos por diversas escolas secundárias.

8.4- Conclusão

O aparecimento da Internet e dos seus serviços estimulou uma mudança radical na sociedade em geral e na educação em particular. A sociedade industrial baseada na produção deu lugar a uma sociedade baseada na informação e no conhecimento e o paradigma de uma educação para toda a vida deu lugar a um modelo de educação ao longo da vida.

Os computadores deixaram de ser dispositivos isolados e, actualmente, constituem os pontos de entrada na rede de troca de informação e das transacções comerciais (Fernández-López, 2002). Do ponto de vista tecnológico, verifica-se que actualmente as capacidades de computação distribuída, em particular das redes de computadores usando os serviços e tecnologias Internet, acrescidas das tecnologias de integração de informação dos mais diversos formatos (tecnologias multimédia), atingiram um estado de desenvolvimento que oferece um enorme potencial de utilização dos meios informáticos como suporte ao ensino e aprendizagem.

Estes factores conduziram gradualmente ao aparecimento de ferramentas de e-Learning e ambientes de aprendizagem hipermédia não só como meio alternativo aos tradicionais processos de ensino, formação e aprendizagem, mas também como ambientes de desenvolvimento de relações sociais e de suporte à realização de actividades colaborativas.

À medida que a Web cresce, a descoberta e a recuperação de materiais educacionais úteis torna-se cada vez mais problemática. Com vista a resolver este problema, é necessário criar uma estrutura de metadados para e-Learning baseada na Web Semântica que satisfaça as necessidades dos professores e alunos aquando da busca de materiais educacionais úteis na Web. Neste sentido, podemos afirmar que a convergência entre as tecnologias de e-Learning (especificações de metadados acreditadas para a tecnologia de aprendizagem) e as tecnologias da Web Semântica (RDF(S)/OWL enquanto recomendações para o desenvolvimento de aplicações legíveis pelos agentes de software) estimulará o desenvolvimento futuro do e-Learning baseado na Web Semântica.

Um dos principais desafios é transformar os dados e aplicações em elementos úteis, legíveis e compreensíveis para os agentes inteligentes, facultando-lhes a comunicação e a cooperação dinâmicas. Ou seja, por um lado, é necessário facilitar a pesquisa e a interpretação através de normas que definam de forma universal o significado do conteúdo das páginas Web, personalizando os conteúdos. Por outro, é urgente desenvolver aplicações que consigam

simultaneamente partilhar informação de diversas fontes e inferir informação ou conhecimento adicional. Tal como acontece na geração de páginas Web baseadas em bases de dados, pretende-se obter um comportamento semelhante em toda a Web, mas incluindo conhecimento implícito.

Os últimos anos têm sido caracterizados pela aplicação das tecnologias para a Web Semântica às Intranets. O sistema WSE proposto nesta tese aplica as tecnologias para a Web Semântica ao nível de uma Extranet, ou seja, entre sistemas de e-Learning. A gradual aplicação destas tecnologias a toda a Web permitirá, num futuro próximo, a sua evolução para uma Web provida de semântica ou significado não só para os humanos, mas também para as máquinas.

Actualmente, poucos são aqueles que, fora das comunidades científicas, têm consciência de que o futuro passa pela Web Semântica. Com a Web Semântica, o modo de lidar com a informação, de obter dados e de os relacionar na rede mudará. Contudo, muito há ainda a fazer. Por um lado, enquanto não houver um verdadeiro interesse comercial a evolução para uma WS que aparentemente todos desejamos decorrerá muito lentamente. Por outro, quando os menos atentos passarem a trabalhar com um computador capaz de entender o que necessitam, capaz de encontrar na Web aquela informação que tanto desejam e capaz de aprender com eles tornando-se num verdadeiro parceiro do processo de aprendizagem, então sim reconhecerão todo o trabalho que está a ser levado a cabo.

As “máquinas de ensinar” de Skinner passaram de moda, mas a actualidade de uma frase que lhe foi atribuída mantém-se intacta relativamente ao impacto desta tecnologia no final do século XX: *The real problem is not whether machines think but whether men do*.

Sabendo que nem tudo que luz é ouro e que a educação não é um “produto” fácil de embalar, armazenar e distribuir, estamos convictos de que a Web Semântica contribuirá significativamente para o sucesso do e-Learning.

Acreditamos num futuro muito mais próximo do que alguns auguram, em que passaremos mais tempo a ler, ver e ouvir recursos educativos e muito menos tempo a filtrar essa informação.

Referências Bibliográficas e Webliográficas

ADL (2004). Sharable Content Object Reference Model (SCORM). *Advanced Distributed Learning* Acesso em: 04 de Junho de 2004, Disponível em: http://www.adlnet.org

Almeida, & Bax. (2003). Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção. *Ciência da Informação, Instituto Brasileiro de Informação em Ciência e Tecnologia, Ministério da Ciência e Tecnologia*. Acesso em: 01 de Novembro de 2004, Disponível em: http://www.ibict.br/cienciadainformacao/viewarticle.php?id=36&layout=html

Almeida, M., Moura, M., Cardoso, A. M., & Cendón, B. (2005). Uma iniciativa insterinstitucional para construção de ontologia sobre ciência da informação: Visão geral do projecto P.O.I.S., *Encontros BIBLI* (pp. 53-72). Florianopolis: Universidade Federal de Santa Catarina.

Alves, A. F. (2003). Uma Nova Metáfora para a Aprendizagem. In A. F. Alves & V. Gonçalves (Eds.), *Sebenta de Educação Multimédia - Mestrado em Tecnologia Multimédia*. Bragança: Escola Superior de Educação do Instituto Politécnico de Bragança.

Amaral, L. (1994). *Planeamento de Sistemas de Informação, Dissertação de Doutoramento.* Universidade do Minho.

Amaral, M. T. (2005). Rumo à sociedade da aprendizagem. *Parceiros na Aprendizagem - Microsoft Educação Brasil*. Acesso em: 25 de Janeiro de 2006, Disponível em: http://www.microsoft.com/brasil/educacao/parceiro/entrev.mspx

Angele, J., Boley, H., Bruijn, J. d., Fensel, D., Hitzler, P., Kifer, M., et al. (2005). Web Rule Language (WRL). *W3C Member Submission*. Acesso em: 16 de Novembro de 2006, Disponível em: http://www.w3.org/Submission/2005/SUBM-WRL-20050909/

Antoniou, G., & Harmelen, F. (2004). A Semantic Web Primer. [S.l.]: The MIT Press.

ARIADNE (2004). Transformation of ARIADNE XML instances into instances of LOM. *ARIADNE Foundation for the European Knowledge Pool*. Acesso em: 15 de Julho de 2005, Disponível em: http://www.ariadne-eu.org/index.html

Articulation (2006). Articulation Service. *Version 0.2*. Acesso em: 26 de Janeiro de 2006, Disponível em: http://codip.grci.com/Tools/ArtiServicePage.html

ATutor. (2004). ATutor: Learning Content Management System. Acesso em: 30 de Setembro de 2004, Disponível em: http://www.atutor.ca/atutor/download.php

Ausubel, D. P. (1968). *Educational Psychology: A Cognitive View*. New York: Holt, Rinehart and Winston.

Barbosa, I. C. (2003). *Prospecção de Dados na Manutenção Semi-automática de Ontologias, Dissertação de mestrado.* Universidade Federal de Minas Gerais, Minas Gerais.

Barta, R. (2005). AsTMa= 2.0 Language Definition. *Bond University*. Acesso em: 12 de Dezembro de 2005, Disponível em: http://astma.it.bond.edu.au/astma=-spec-2.0r1.0.dbk

Beckett, D. (2004). RDF/XML Syntax Specification (Revised). *W3C Recommendation*. Acesso em: 10 de Agosto de 2005, Disponível em: http://www.w3.org/TR/2004/REC-rdf-syntax-grammar-20040210/

Beckett, D., Miller, E., & Brickley, D. (2002). Expressing Simple Dublin Core in RDF/XML. *Dublin Core Metadata Initiative*. Acesso em: 30 de Outubro de 2004, Disponível em: http://dublincore.org/documents/2002/07/31/dcmes-xml/

Bergman, M. K. (2001). The Deep Web: Surfacing Hidden Value. *Journal of Electronic Publishing from the University of Michigan*. Acesso em: 10 de Novembro de 2003, Disponível em: http://www.press.umich.edu/jep/07-01/bergman.html

Bergman, M. K. (2001). The Deep Web: Surfacing Hidden Value. *BrighPlanet*. Acesso em: 01 de Outubro de 2004, Disponível em: http://brightplanet.com/technology/deepweb.asp

Berners-Lee, T. (2000). Axioms of Web Architecture - Rules and Facts: Inference engines vs Web. *World Wide Web Consortium*. Acesso em: 20 de Outubro de 2004, Disponível em: http://www.w3.org/DesignIssues/Rules.html

Berners-Lee, T. (2002). The Semantic Web. 2002. Acesso em: 30 de Agosto de 2005, Disponível em: http://www.w3.org/2002/Talks/04-sweb-sloan/Overview.html

Berners-Lee, T. (2005). Semantic Web Concepts. *WWW Consortium, MIT Computer Science & Artificial Inteligence Laboratory*. Acesso em: 30 de Outubro de 2006, Disponível em: http://www.w3.org/2005/Talks/0517-boit-tbl/

Berners-Lee, T., Fielding, R., & Masinter, L. (1998). RFC 2396 - Uniform Resource Identifiers (URI): Generic Syntax. *IETF*. Acesso em: 10 de Maio de 2004, Disponível em: http://www.isi.edu/in-notes/rfc2396.txt

Berners-Lee, T., Hendler, J., & Lassila, O. (2001). The semantic web – a new form of the Web content that is meaningful to computer will unleash a revolution of new possibilities. *Scientific American*. Acesso em: 01 de Setembro de 2003, Disponível em: http://www.sciam.com/print_version.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21

Biezunski, M., Bryan, M., & Newcomb, S. (2002). M. Biezunski, M. Bryan, S. Newcomb. *English - ISO/IEC 13250 Topic Maps - Information Technology - Document Description and Processing Languages - Second Edition*. Acesso em: 28 de Julho de 2006, Disponível em: http://www1.y12.doe.gov/capabilities/sgml/sc34/document/0322_files/iso13250-2nd-ed-v2.pdf

Boehm, B. (1988). A Spiral Model of Software Development and Enhancement. *IEEE Computer, 21*, 61-72.

Boehm, B. (2000). Spiral Development: Experience, Principles, and Refinements. In W. Hansen (Ed.), *Spiral Development Workshop - Special Report CMU/SEI-2000-SR-008*. Pittsburgh: Software Engineering Institute - Carnegie Mellon University.

Boley, H., Dean, M., Grosof, B., Kifer, M., Tabet, S., & Wagner, G. (2005a). RuleML Position Statement, *W3C Workshop on Rule Languages for Interoperability Position Paper*. Washington, D.C., USA: The RuleML Initiative - WWW Consortium.

Boley, H., Dean, M., Grosof, B., Sintek, M., Spencer, B., Tabet, S., et al. (2005b). FOL RuleML: The First-Order Logic Web Language. *W3C Member Submission*. Acesso em: 15 de Novembro de 2006, Disponível em: http://www.w3.org/Submission/2005/SUBM-FOL-RuleML-20050411/

Bollen, J., & Heylighen, F. (2005). Learning Webs. In Heylighen, Joslyn & Turchin (Eds.), *Principia Cybernetica Web*. Brussels: Principia Cybernetica.

Booch, G., Jacobson, I., & Rumbaugh, J. (1999). *Unified Modeling Language User Guide*. [S.l.]: Addison-Wesley Longman.

Bouquet, P., Serafini, L., & Zanobini, S. (2003). Semantic Coordination: A New Approach and an Application, *Proceedings of the International Semantic Web Conference* (pp. 130-143). Florida.

Bourda, Y., & Bich-Liên, D. (2003). The Semantic Web for Learning Resources. *IEEE - Computer Society*. Acesso em: 23 de Novembro de 2004, Disponível em: http://csdl.computer.org/comp/proceedings/icalt/2003/1967/00/19670322.pdf

Brandon, H. (2002). LMS and LCMS Demystified. *Brandon Hall Research*. Acesso em: Janeiro 2004, Disponível em: http://www.brandon-hall.com/free_resources/lms_and_lcms.shtml

Brase, J., & Nejdl, W. (2004). Ontologies and Metadata for eLearning. In S. Staab & R. Studer (Eds.), *Handbook on Ontologies* (pp. 555-574). Berlin: Springer-Verlag.

Bray, T., Hollander, D., & Layman, A. (1999). Namespaces in XML. *World Wide Web Consortium*. Acesso em: 16 de Janeiro de 2006, Disponível em: http://www.w3.org/TR/1999/REC-xml-names-19990114/

Bray, T., Paoli, J., C.M., S.-M., & Maler, E. (2000). Extensible Markup Language (XML) 1.0, Second Edition. *World Wide Web Consortium*. Acesso em: 17 de Dezembro de 2003, Disponível em: http://www.w3.org/TR/2000/REC-xml-20001006

Breitman, K. (2005). *Web Semântica: A Internet do Futuro*. Brasil: LTC.

Brennan, M., Susan, F., & Cushing, A. (2001). The Learning Content Management System - A New eLearning Market Segment Emerges, *IDC White Paper*: International Data Corporation.

Brickley, D., & Guha, R. V. (1999). Resource Description Framework (RDF) schema specification. *W3C Proposed Recommendation*. Acesso em: 26 de Novembro de 2004, Disponível em: http://www.w3.org/TR/1999/PR-rdf-schema-19990303/

Brickley, D., & Guha, R. V. (2004). RDF Vocabulary Description Language 1.0: RDF Schema. *W3C Recommendation*. Acesso em: 25 de Agosto de 2005, Disponível em: http://www.w3.org/TR/2004/REC-rdf-schema-20040210/

Brown, A., Fuchs, M., Robie, J., & Wadler, P. (2001). XML Schema: formal description. *W3C Working Draft*. Acesso em: 30 de Setembro de 2006, Disponível em: http://www.w3.org/TR/2001/WD-xmlschema-formal-20010925/

Bruner, J. (2000). *Cultura da Educação*. Lisboa: Edições 70.

Buzan, T. (1996). Saber Pensar. Portugal: Editorial Presença.

Caglayan, A., & Harrison, C. (1997). *AGENT Sourcebook: A Complete Guide to Desktop, Internet, and Intranet Agents*. [S.l.]: John Wiley & Sons, Inc.

Calvanese, D., Giacomo, G. D., Horrocks, I., Lutz, C., Motik, B., Parsia, B., et al. (2006). OWL 1.1 Web Ontology Language - Tractable Fragments. *W3C Member Submission*. Acesso em: 29 de Dezembro de 2006, Disponível em: http://www.w3.org/Submission/owl11-tractable/

Campos, M., & Lima, P. (2006). GRID Semântico e Ontologias, *Pós-graduação em Informática*: IM/NCE-UFRJ.

Cañas, A. J., Carvajal, R., Carff, R., & Hill, G. (2004). CmapTools, Web Pages & Website. *Technical Report IHMC CmapTools 2004-01*. Acesso em: 06 de Maio de 2005, Disponível em: http://cmap.ihmc.us/Publications/WhitePapers.php

CanCore. (2005). Documents and Presentations. *CanCore*. Acesso em: 11 de Dezembro de 2005, Disponível em: http://www.cancore.ca/en/index.html

Cardinaels, K., Meire, M., & Duval, E. (2005). Automating Metadata Generation: the Simple Indexing Interface, *International World Wide Web Conference Committee (WWW 2005)*. Chiba, Japan: ACM.

Carmo, H. (1999). A Formação da Universidade Aberta, Balanço e Perspectivas, Encontro Nacional de Ensino a Distância. In A. Santos (Ed.), *Ensino a Distância & Tecnologias de Informação - e-learning*. [Lisboa]: FCA - Editora de Informática.

Carriço, J., & Carriço, A. (1998). *JAVA. Computing Guides* (2.ª ed.). Lisboa: CTI Edições.

Carvalho, C., & Cardoso, E. L. (2004). O E-learning e o Ensino Superior em Portugal. *Revista do Sindicato Nacional do Ensino Superior*. Acesso em: 18 de Março de 2005, Disponível em: http://www.snesup.pt/ensinosuperior/revista_10/es10_07.htm

Carvalho, J. Á., & Amaral, L. (1993). Matriz de Actividades: Um enquadramento Conceptual para as Actividades de Planeamento e Desenvolvimento de Sistemas de Informação. *Sistemas de Informação - Revista da Associação Portuguesa de Sistemas de Informação*.

Castañon, G. A. (2005). Construtivismo e ciências humanas, *Ciências & Cognição* (Vol. 5).

Chute, A. G., Thompson, M. M., & Hancock, B. W. (1999). *The McGraw-Hill Handbook of Distance Learning*. [S.l.]: McGraw-Hill.

Clark, R., & Mayer, R. (2002). *E-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning*. San Francisco, CA: Jossey-Bass/Pfeiffer.

Correia, C. (1999). Formação Multimédia On-line, Encontro Nacional de Ensino a Distância. In A. Santos (Ed.), *Ensino a Distância & Tecnologias de Informação - e-learning*. [Lisboa]: FCA - Editora de Informática.

Coser, A. (1999). *Utilização de Agentes Inteligentes no Trabalho Colaborativo Via Internet, Dissertação de Mestrado.* Universidade Federal de Santa Catarina, Florianópolis - SC.

Cowan, J., & Tobin, R. (2001). XML Information Set. *World Wide Web Consortium*. Acesso em: 10 de Agosto de 2006, Disponível em: http://www.w3.org/TR/2001/REC-xml-infoset-20011024

Crosetti, B. d. B. (2000). *Possibilidades educativas de las Webtools*. Palma: Universitat de les Illes Balears.

Cunha, L. (2002). Web Semântica: Estudo Preliminar. *Embrapa*. Acesso em: 06 de Março de 2004, Disponível em: http://www.cnptia.embrapa.br/publica/2002/doc18.pdf

DAML (2002). Why use DAML? *The DARPA Agent Markup Language*. Acesso em: 26 de Julho de 2004, Disponível em: http://www.daml.org

Davies, J., Fensel, D., & Harmelen, F. v. (2003). *Towards The Semantic Web – Ontology-Driven Knowledge Management*. Chichester, England: John Wiley & Sons, Ltd.

DCMI (2002). Dublin Core Metadata. *Dublin Core Metadata Initiative*. Acesso em: 17 de Março de 2004, Disponível em: http://www.dublincore.org

DCMI (2006). Tools and Software. *Dublin Core Metadata Initiative*. Acesso em: 01 de Abril de 2006, Disponível em: http://www.dublincore.org/tools/

DCMLTSCTaskforce (2005). Joint DCMI/IEEE LTSC Taskforce - Introduction. *Dublin Core Metadata Initiative*. Acesso em: 03 de Agosto de 2006, Disponível em: http://www.dublincore.org/educationwiki/DCMIIEEELTSCTaskforce

Denny, M. (2004). Ontology Building: A Survey of Editing Tools Revisited *O'Reilly's XML.com*. Acesso em: 26 de Outubro de 2006, Disponível em: http://www.xml.com/pub/a/2004/07/14/onto.html

DGES. (2005). Descritores de Dublin. *Direcção Geral do Ensino Superior - Ministério da Ciência, Tecnologia e do Ensino Superior*. Acesso em: 20 de Março de 2006, Disponível em: http://www.dges.mctes.pt/Bolonha

DOAN, A., MADHAVAN, J., DHAMANKAR, R., DOMINGOS, P., & HALEVY, A. (2003). Learning to match ontologies on the Semantic Web. *The VLDB Journal - The International Journal on Very Large Data Bases, 12*(4), 303-319.

Dori, D. (2003). Object-Process Methodology as a basis for the Visual Semantic Web, *14th International Workshop on Database and Expert Systems Applications (DEXA'03)*.

Driscoll, M. (1998). *Web - Based Training: using technology to design adult learning experiences*. [S.l.]: Jossey-Bass Pfeiffer.

Drucker, P. (2000). Need to Know: Integrating e-Learning with High Velocity Value Chains. *A Delphi Group White Paper*. Acesso em: 17 de Fevereiro de 2004, Disponível em: http://www.delphigroup.com/research/whitepapers.htm

Duval, E., Hodgins, W., Sutton, S., & Weibel, S. L. (2002). Metadata Principles and Practicalities, *D-Lib Magazine* (Vol. 8).

Edutella. (2006). Edutella Project home: RDF-based Metadata Infrastructure for P2P Applications. Acesso em: 19 de Agosto de 2006, Disponível em: http://edutella.jxta.org/

ELENA (2006). About ELENA: Smart Spaces for LearningTM. *Information Society Technologies.* Acesso em: 17 de Fevereiro de 2006, Disponível em: http://www.elena-project.org

Fallside, D. C. (2001). XML Schema Part 0: Primer. *W3C Recommendation*. Acesso em: 09 de Julho de 2004, Disponível em: http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/

Fellbaum, C. (1999). *WORDNET: an electronic lexical database and some of its applications*. Cambridge, MA: MIT Press.

Fensel, D. (2001). *Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce*. [S.l.]: Springer Verlag.

Fensel, D., Wahlster, W., Lieberman, H., & Hendler, J. (2002). *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. [S.l.]: MIT Press.

Ferber, J. (1999). Multi-Agent System - An Introduction to Distributed Artificial Intelligence. [S.l.]: Addison-Wesley Publishers.

Fernández-López, M. (2002). Deliverable 1.4: A survey on methodologies for developing, maintaining, evaluating and reengineering ontologies. *IST Project IST-2000-29243 OntoWeb: Ontology-based Information Exchange for Knowledge Management and Electronic Commerce*. Acesso em: 28 de Setembro de 2006, Disponível em: http://www.aifb.uni-karlsruhe.de/WBS/ysu/publications/OntoWeb_Del_1-4.pdf

Ferreira, E., & Kofugi, S. (2006). Geração Automática de Metadados. In M. Cunha & Á. Rocha (Eds.), *Actas da 1.ª Conferência Ibérica de Sistemas e Tecnologias de Informação - Sistemas e Tecnologias de Informação no Espaço Ibérico* (Vol. 2). Ofir: Escola Superior de Tecnologia - Instituto Politécnico do Cávado e do Ave.

Ferreira, J. M. M., & Santiago, M. A. S. (1999). Ensino e Aprendizagem na Era da Internet. In J. A. Alves, P. Campos & P. Q. Brito (Eds.), *O futuro da Internet*. [S.l.]: Edições Centro Atlântico.

Freitas, J. C. (1999). De onde vimos e para onde vamos: o futuro da Internet na Escola. In J. A. Alves, P. Campos & P. Q. Brito (Eds.), *O futuro da Internet*. [S.l.]: Edições Centro Atlântico.

Frigo, L. B., Pozzebon, E., & Bittencourt, G. (2004). O Papel dos Agentes Inteligentes nos Sistemas Tutores Inteligentes. *World Congress on Engineering and Technology Education*. Acesso em: 09 de Novembro de 2004, Disponível em: http://www.inf.ufsc.br/~l3c/artigos/frigo04a.pdf

Gaines, B. R., & Shaw, M. L. G. (1995). Concept Maps as Hypermedia Components, *Knowledge-based hypermedia, International journal of human-computer studies* (Vol. 43). London: Elsevier.

García, R. I. C. (2002). Búsqueda de información en Internet. In *Educar en Red - Internet como recurso para la educación* (pp. 83-100). [S.l.]: Ediciones ALJIBE, SL.

Garrison, D. R. (1989). *Understanding Distance Education - a framework for the future*. London and New York: Routledge.

Garshol, L. M. (2002). LTM – The Linear Topic Map Notation. *Ontopia*. Acesso em: 13 de Dezembro de 2005, Disponível em: http://www.ontopia.net/topicmaps/ltm.html

Garshol, L. M. (2003). Living with topic maps and RDF. *Ontopia*. Acesso em: 08 de Junho de 2006, Disponível em: http://www.ontopia.net/topicmaps/materials/tmrdf.html

Garshol, L. M. (2004). Metadata? Thesauri? Taxonomies? Topic Maps! Making sense of it all. *Ontopia*. Acesso em: 28 de Fevereiro de 2006, Disponível em: http://www.ontopia.net/topicmaps/materials/tm-vs-thesauri.html

Garshol, L. M., & Barta, R. (2005). Topic Map Query Language (TMQL). *ISO/IEC Topic Maps*. Acesso em: 18 de Agosto de 2006, Disponível em: http://www.isotopicmaps.org/tmql/

Giese, L. F. (1998). *GIESE, Estrutura de Agentes para o Processo de Compra e Venda utilizando Tomada de Decisão Difusa, Dissertação de Mestrado.* Universidade Federal de Santa Catarina, Florianópolis.

Goldfarb, C. F., & Prescod., P. (2001). *XML Handbook* (4.ª ed.). USA: Prentice Hall.

Gomes, M. J. (2005). Desafios do e-learning: do conceito às práticas. In S. A. Leandro & D. S. Bento (Eds.), *Actas do VIII Congresso Galaico-Português de Psico-Pedagogia* (pp. 66-76). Braga: Centro de Investigação em Educação - Universidade do Minho.

Gomes, M. J. (2006). E-Learning e Educação On-Line: Contributos para os princípios de Bolonha, *Actas do VII Colóquio sobre Questões Curriculares - Globalização e (des) igualdades: os desafios curriculares (III Colóquio Luso-Brasileiro)*. Braga: Centro de Investigação em Educação - Universidade do Minho.

Gonçalves, V. (2002). *Desenvolvimento de Sistemas de Informação para a Web, Dissertação de Mestrado em Tecnologia Multimédia.* Faculdade de Engenharia da Universidade do Porto.

Gonçalves, V., & Rodrigues, C. (2006). Um Sistema de b-Learning no ensino Secundário Português. In L. Alonso, L. González, B. Manjón & M. Nistal (Eds.), *Proceedings of the 8th International Symposium on Computers in Education (SIIE'06)* (Vol. 2). Leon: Universidad de Leon.

Goñi, J., Fernandes, C., & Lucena, C. (2002). E-learning e a Web Semântica. *VI Congresso da RIBIE, Pontifícia Universidade Católica do Rio de Janeiro*. Acesso em: 01 de Abril de 2004, Disponível em: http://lsm.dei.uc.pt/ribie/docfiles/txt2003729185022paper-212.pdf

Grant, J., & Beckett, D. (2004). RDF Test Cases. *W3C Recommendation*. Acesso em: 12 de Setembro de 2005, Disponível

Greenagel, F. L. (2002). The Illusion of e-Learning: Why We Are Missing Out on the Promise of Technology. *League for Innovation in the Community College*. Acesso em: 18 de Agosto de 2005, Disponível em: http://www.league.org/publication/whitepapers/0802.html

Greenberg, J., Spurgin, K., & Crystal, A. (2005). Final Report for the AMeGA (Automatic Metadata Generation Applications) Project. *Library of Congress*. Acesso em: 12 de Novembro de 2006, Disponível em: http://www.loc.gov/catdir/bibcontrol/lc_amega_final_report.pdf

Gruber, T. R. (1993). A translation approach to portable ontologies, *Knowledge Acquisition* (Vol. 5(2), pp. 199-220).

Gutiérrez, R. (2006). Steve Ballmer - Presidente de Microsoft: Nuestra idea es ser líderes en búsquedas y publicidad 'online', *El pais*.

Haarslev, V., & Moller, R. (2003). Racer: A Core Inference Engine for the Semantic Web, *Proceedings of the 2nd International Workshop on Evaluation of Ontology-Based Tools (EON 2003)*. ISWC2003, Sanibel Island, Florida, USA.

Haav, H. M., & Lubi, T. L. (2001). A survey of concept-based information retrieval tools on the web, *Proceedings of the East-European Conference Adbis*.

Han, H. C., Giles, L., Manavoglu, E., Zha, H., Zhang, Z., & Fox, E. A. (2003). Automatic document metadata extraction using support vector machines, *Proceedings of the Third ACM/IEEE-CS Joint Conference on Digital Libraries* (pp. 37-48). New York: ACM Press.

Hannafin, M. J., Land, S., & Oliver, K. (1999). Open Learning Environments: Foundations, Methods and Models. In C. M. Reigeluth (Ed.), *Instructional-design theories and models: A New Paradigm of Instructional Theory* (Vol. 2). Mahwah, NJ: Lawrence Erlbaum Associates.

Harmelen, F. (2004). The Semantic Web: What, Why, How, and When. *IEEE Intelligent Systems*.

Harold, E. R. (1999). *XML Bible* (1.ª ed.). Chicago: IDG Books.

Hayes, P. (2004). RDF Semantics. *W3C Recommendation.* Acesso em: 10 de Setembro de 2005, Disponível em: http://www.w3.org/TR/2004/REC-rdf-mt-20040210/

Hendler, J. (2001). Agents and the Semantic Web. *IEEE Intelligent Systems Journal*. Acesso em: 09 de Maio de 2004, Disponível em: http://www.cs.umd.edu/users/hendler/AgentWeb.html

Heylighen, F. (2000). The Social Superorganism and its Global Brain. In Heylighen, Joslyn & Turchin (Eds.), *Principia Cybernetica Web*. Brussels: Principia Cybernetica.

Heylighen, F. (2002). Conceptions of a Global Brain: an historical review. In Heylighen, Joslyn & Turchin (Eds.), *Principia Cybernetica Web*. Brussels: Principia Cybernetica.

Heylighen, F., Joslyn, C., & Turchin, V. (2001). Overview of Principia Cybernetica. In F. Heylighen, C. Joslyn & V. Turchin (Eds.), *Principia Cybernetica Web*. Brussels: Principia Cybernetica.

Hillmann, D. (2005). Using Dublin Core. *Dublin Core Metadata Initiative*. Acesso em: 30 de Março de 2006, Disponível em: http://dublincore.org/documents/usageguide/

Hjelm, J. (2001). *Creating the Semantic Web with RDF*. [S.l.]: John Wiley.

Hodges, W. (1997). A Shorter Model Theory. Cambridge: University Press.

Hodgson, R. (2006). Semantic Web Capability Cases - making the business case for Semantic Technology, *Proceedings of the 15th International World Wide Web Conference*. Edinburgh International Conference Centre, Scotland.

Hofmann, J. (2002). Blended Learning Case Study. In A. Rossett (Ed.), *The ASTD E-Learning Handbook: Best Practices, Strategies and Cases Studies for an emerging field*. New York: McGraw-Hill.

Hoog, R. D. (1998). Methodologies for Building Knowledge Based Systems: Achievements and Prospects. In J. Liebowitz (Editor) Handbook of Expert Systems. In J. Liebowitz (Ed.), *Handbook of Expert Systems*. [S.l.]: CRC.

Horrocks, I., Patel-Schneider, P. F., Boley, H., Tabet, S., Grosof, B., & Dean, M. (2004). SWRL: A Semantic Web Rule Language Combining OWL and RuleML. *W3C Member Submission.* Acesso em: 14 de Novembro de 2006, Disponível em: http://www.w3.org/Submission/2004/SUBM-SWRL-20040521/

Horrocks, I., Patel-Schneider, P. F., & Harmelen, F. V. (2003). From SHIQ and RDF to OWL: The Making of a Web Ontology Language. *Journal of Web Semantics, 1(1)*, 7-26.

IEEE-LTSC (2002). IEEE 1484.12.1-2002 - Draft Standard for Learning Object Metadata. *Learning Technology Standards Committee of the IEEE*. Acesso em: 11 de Dezembro de 2004, Disponível em: http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf

IMS (2004). IMS Learning Resource Metadata Specification. *Instructional Management Systems Project - IMS Global Learning Consortium.* Acesso em: 02 de Junho de 2004, Disponível em: http://www.imsglobal.org/specifications.html

IMS (2006). Learning Resource Metadata Specification - Version 1.3 - Final Specification – HTML. *Instructional Management Systems Project - IMS Global Learning Consortium*. Acesso em: 02 de Outubro de 2006, Disponível em: http://www.imsglobal.org/metadata/index.html

Infante, I. e. O., Pedro. (2007, Janeiro). O poder está nas suas mãos. *Revista Exame Informática, Edimpresa, 139*.

IST (2006). Information Society Technology. *Community Research & Development Information Service*. Acesso em: 28 de Dezembro de 2006, Disponível em: http://cordis.europa.eu/ist/

Jang, M., & Sohn, J.-c. (2004). Bossam: an extended rule engine for the web, *Proceedings of RuleML 2004* (Vol. 3323): LNCS.

Jasper, R., & Uschold, M. (1999). A framework for understanding and classifying ontology applications, *IJCAI-99 - ONTOLOGY WORKSHOP*. Stockholm.

JAVA (2004). Java SE at a Glance. *Sun Microsystems*. Acesso em: 05 de Novembro de 2004, Disponível em: http://java.sun.com/javase/

Jenkins, C., Jackson, M., Burden, P., & Wallis, J. (1999). Automatic RDF Metadata generation for Resource Discovery. In E. Tang (Ed.), *Proceedings of the WWW8 - Eighth International World Wide Web Conference* (Vol. 31, pp. 1305-1320). Toronto, Canada: Foretec Seminars.

JISC (2006). JISC: Supporting education and research. *Joint Information Systems Committee*. Acesso em: 28 de Dezembro de 2006, Disponível em: http://www.jisc.ac.uk/

Johnston, P., & Powell, A. (2006). Update on Work of Joint DCMI/IEEE-LTCS Task Force, *CETIS Metadata & Digital Repositories SIG Meeting*. UKOLN, Bath.

Jonassen, D. (1999). Designing Constructivist Learning Environments. In C. M. Reigeluth (Ed.), *Instructional-design theories and models: A New Paradigm of Instructional Theory*. University of Georgia: Lawrence Erlbaum Associates.

Jurado, F., Redondo, M., & Ortega, M. (2006). Modelo de referencia para especificación de tareas colaborativas com IMS-LD. In L. Alonso, L. González, B. Manjón & M. Nistal (Eds.), *Proceedings of the 8th International Symposium on Computers in Education (SIIE'06)* (Vol. 2). Leon: Universidad de Leon.

Keller, J. M. (1999). Applying the ARCS Model of Motivational Design in Distance Learning. *Florida State University*. Acesso em: 02 de Junho de 2005, Disponível em: http://mailer.fsu.edu/~jkeller/john/index.htm

Kemp, J. E., Morrison, G. R., & E Ross, S. M. (1998). *Designing Effective Instruction*. New Jersey: Prentice-Hall.

Kemp, J. E., Morrison, G. R., & Ross, S. M. (1998). *Designing Effective Instruction* (2.ª ed.). Upper Saddle River, NJ: Prentice Hall.

Khan, B. H. (2001). A framework for Web-based learning. In B. H. Khan (Ed.), *Web-based training*. Englewood Cliffs, NJ: Educational Technology Publications.

Khan, B. H. (2004). The people-process-product continuum in e-learning: The e-learning P3 model. *Educational Technology Publications*, *44*(5), 33-44.

Khan, B. H. (2005). Managing e-learning: Design, delivery, implementation, and evaluation. In B. H. Khan (Ed.). Hershey, PA: Information Science Publishing.

Khan, B. H. (2006). *Flexible Learning in an Information Society*. [S.l.]: Information Science Publishing.

Klyne, G., & Carroll, J. (2004). Resource Description Framework (RDF): Concepts and Abstract Syntax. *W3C Recommendation*. Acesso em: 05 de Agosto de 2005, Disponível em: http://www.w3.org/TR/2004/REC-rdf-primer-20040210/

Kokkelink, S., & Schwänzl, R. (2002). Expressing Qualified Dublin Core in RDF/XML. *Dublin Core Metadata Initiative*. Acesso em: 17 de Setembro de 2005 Disponível em: http://dublincore.org/documents/2002/05/15/dcq-rdf-xml/

Koper, R. (2001). Modeling units of study from a pedagogical perspective: the pedagogical meta-model behind EML. *First Draft. Open University of the Netherlands: Educational Technology Expertise Centre* Acesso em: 12 de Setembro de 2004, Disponível em: http://eml.ou.nl/introduction/docs/ped-metamodel.pdf

Kruse, K. (2003). The State of e-Learning: Looking at History with the Technology Hype Cycle. *e-LearningGuru*. Acesso em: 22 de Maio de 2005, Disponível em: http://www.e-learningguru.com/articles/hype1_1.htm

Kruse, K. (2004). Evaluating e-Learning: Introduction to the Kirkpatrick Model. *e-LearningGuru.* Acesso em: 18 de Maio de 2005, Disponível em: http://www.e-learningguru.com/articles/art2_8.htm

Kruse, K. e. K., Jason. (2000). *Technology-Based Training: the Art and Science of Design, Development, and Delivery*. California: Jossey-Bass/Pfeiffer.

Lange, D., & Oshima, M. (1998). *Programming and Deploying Java Mobile Agents with Aglets.* Boston: Addison-Wesley Longman Publishing Co., Inc.

Lassila, O., & Swick, R. (1999). Resource Description Framework (RDF) Model and Syntax Specification. *World Wide Web Consortium.* Acesso em: 12 de Janeiro de 2004, Disponível em: http://www.w3.org/TR/1999/REC-rdf-syntax-19990222/

Lévy, P. (2003). *Semantic Web: A tool for Education and Human Development.* Artigo apresentado em: University of Ottawa, Ottawa.

Librelotto, G. R. (2005). *Topic Maps: da Sintaxe à Semântica, Dissertação de doutoramento.* Universidade de Braga, Braga.

Lima, J. R., & Capitão, Z. C. (2003). E-Learning e E-Conteúdos, *Sociedade da Informação*. [S.l.]: Edições Centro Atlântico.

LOMWG (2002). Standard for Learning Object Metadata. *IEEE-LTSC Committee.* Acesso em: 26 de Abril de 2004, Disponível em: http://ltsc.ieee.org/wg12/index.html

Lourenso, R. (2005). *Segmentação De Objetos Complexos Em Um Sistema De Banco De Dados Objeto Relacional Baseado Em Grids, Dissertação de Doutoramento.* Escola Politécnica da Universidade de São Paulo, São Paulo.

LUISA (2006). Learning Content Management System Using Innovative Semantic Web Services Architecture. *European Comission FP6 funded Specific Targeted Research Project*. Acesso em: 20 de Novembro de 2006, Disponível em: http://www.luisa-project.eu

Maedche, A., & Staab, S. (2001). Learning Ontologies for the Semantic Web. *Proceedings of the Semantic Web 2001*. Acesso em: 03 de Julho de 2004, Disponível em: http://www.aifb.uni-karlsruhe.de/WBS/sst/Research/Publications/semweb2001.pdf

Manola, F., & Miller, E. (2004). RDF Primer. *W3C Recommendation*. Acesso em: 05 de Agosto de 2005, Disponível em: http://www.w3.org/TR/2004/REC-rdf-primer-20040210/

Marsh, J. (2001). XML Base. *World Wide Web Consortium*. Acesso em: 18 de Março de 2004, Disponível em: http://www.w3.org/TR/2001/REC-xmlbase-20010627/

Martínez, J. M. (2003). MPEG-7 Overview (version 9). *ISO/IEC JTC1/SC29/WG11*. Acesso em: 04 de Setembro de 2004, Disponível em: http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm

Mayer, R. (1999). Design Instruction for Constructivist Learning. In C. M. Reigeluth (Ed.), *Instructional-design theories and models: A New Paradigm of Instructional Theory*. University of California: Lawrence Erlbaum Associates.

McGreal, R., Gram, T., & Marks, T. (2000). A Survey of New Media Development and Delivery Sotware for Internet-Base Learning. In B. d. B. Crosetti (Ed.), *Possibilidades educativas de las Webtools*. Palma: Universitat de les Illes Balears.

McGuinness, D. L., & Harmelen, F. v. (2004). OWL Web Ontology Language - Overview. *W3C Recommendation*. Acesso em: 15 de Julho de 2005, Disponível em: http://www.w3.org/TR/2004/REC-owl-features-20040210/

McLuhan, M. (1964-1969). *La compresión de los medios como las extensiones del hombre*. [S.l.]: Editorial Diana.

Merril, D. M. (2000). Does Your Instruction Rate 5 Stars?, *International Workshop on Advanced Learning Technologies*. New Zealand: IEEE Learning Technology Task Force.

Michael, C., Obrst, L. J., & Smith, K. T. (2003). *The Semantic Web - A guide to the future of XML, Web Services and Knowledge Management*. Indianapolis, Canada: Wiley Publishing, Inc.

Miller, E. (1998). An Introduction to the Resource Description Framework. *D-Lib Magazine*. Acesso em: 15 de Julho de 2004, Disponível em: http://www.dlib.org/dlib/may98/miller/05miller.html

Miller, E. (2004). Weaving Meaning: An Overview of The Semantic Web. *Presentation at the University of Michigan, Ann Arbor, Michigan USA*. Acesso em: 02 Julho de 2006, Disponível em: http://www.w3.org/2004/Talks/0120-semweb-umich/

Miller, E. (2006, 26 de Abril de 2006). *W3C Semantic Web Activity Lead.* Artigo apresentado em: Semantiske dager 2006 (Norwegian Semantic Days 2006), Stavanger, Norway.

Mizoguchi, R., Vanwelkenhuysen, J., & Ikeda, M. (1995). Task ontology for reuse of problem solving knowledge, *Proceedings Of Ecai'94 Towards Very Large Knowledge Bases* (pp. 46-59). Amsterdam: IOS Press.

Moodle. (2004). Moodle course management system. *Moodle Pty Ltd*. Acesso em: 01 de Julho de 2004, Disponível em: http://moodle.org

Moore, G., Bogachev, D., & Nishikawa, M. (2005). Topic Maps Constraint Language. *ISO/IEC Topic Maps*. Acesso em: 29 de Agosto de 2006, Disponível em: http://www.isotopicmaps.org/tmql/

Moore, M. (1991). Theory of Distance Education, *Distance Education Symposium: Selected Papers, Part 3*. Pennsylvania: Pennsylvania State University.

Moore, M. G., & Thompson, M. M. (1997). The Effects of Distance Learning. University Park, PA: American Center for the Study of Distance Education - Pennsylvania State University.

Mortimer, L. (2002). Learning Objects of Desire: Promise and Practicality. *Learning Circuits*. Acesso em: 07 de Agosto de 2004, Disponível em: http://www.learningcircuits.org/2002/apr2002/mortimer.html

MOSAICA (2006). Semantically Enhanced Multifaceted Collaborative Access to Cultural Heritage. *European Commission - IST Programme* Acesso em: 16 de Dezembro de 2006, Disponível em: http://www.mosaica-project.eu/

Moura, A. M. (2006). Linguagens de Regras para a Web. *Tópicos Especiais em Sistemas de Informação I - Web Semântica*. Acesso em: 13 de Dezembro de 2006, Disponível em: http://dataware.nce.ufrj.br:8080/dataware/fisico/cursos/c2006/TESIWS/

MSI (1997). Livro Verde para a Sociedade da Informação, *Missão para a Sociedade da Informação*. Lisboa: Edição da MSI - Ministério da Ciência e Tecnologia.

MySQL (2005a). MySQL The world's most popular open source database *MySQL AB*. Acesso em: 21 de Janeiro 2005, Disponível em: http://www.mysql.com/

MySQL (2005b). MySQL Connector/J. Acesso em: 14 de Abril de 2005, Disponível em: http://www.mysql.com/products/connector/j/

MySQL (2006). MySQL 5.0 Reference Manual. *MySQL AB*. Acesso em: 27 de Julho de 2006, Disponível em: http://www.mysql.com

Naeve, A., Nilsson, M., Palmér, M., & Paulsson, F. (2005). Contributions to a Public e-Learning Platform - Infrastructure, Architecture, Frameworks and Tools. *International Journal of Learning Technology, 1*(3), 352-381.

Najjar, J., Duval, E., Ternier, S., & Neven, F. (2003). Towards Interoperable Learning Object Repositories: The Ariadne Experience, *Proceedings of the IADIS International Conference on WWW/Internet 2003* (Vol. 1, pp. 219-226). Portugal: International Association for Development of the Information Society.

Nejdl, W., Wolf, B., Qu, C., Decker, S., Sintek, M., Naeve, A., et al. (2002). EDUTELLA: A P2P Networking Infrastructure Based on RDF. Acesso em: 07 de Dezembro de 2004, Disponível em: http://edutella.jxta.org/reports/edutella-whitepaper.pdf

Nejdl, W., Wolf, B., Staab, S., & Tane, J. (2001). EDUTELLA: Searching and Annotating Resources within an RDF based P2P Network. Acesso em: 06 de Dezembro de 2004, Disponível em: http://edutella.jxta.org/reports/edutella_p2p.pdf

Newcomb, S. R., Biezunski, M., & Bryan, M. (2003). The HyTime Topic Maps (HyTM) Syntax. *Information Technology - Document Description and Processing Languages.* Acesso em: 12 de Dezembro de 2005, Disponível em: http://www.jtc1sc34.org/repository/0391.htm

Nilsson, M., Palmér, M., & Brase, J. (2003). The LOM RDF Binding - Principles and Implementation. *Proceedings of the Third Annual ARIADNE conference* Acesso em: 12 de Fevereiro de 2006, Disponível em: http://kmr.nada.kth.se/papers/SemanticWeb/LOMRDFBinding-ARIADNE.pdf

Novak, J. D. (1977). *A Theory of education*. New York: Ithaca, Cornell. University Press

Novak, J. D. (2000). *Aprender, criar e utilizar o conhecimento. Mapas conceptuais como ferramentas de facilitação nas escolas e empresas.* Lisboa: Plátano Universitária.

Noy, N., & Mcguinness, D. (2001). Ontology Development 101 – A guide to creating your first ontology. *KSL Technical Report, Standford University*. Acesso em: 12 Fevereiro de 2006, Disponível em: http://www.ksl.stanford.edu/people/dlm/papers/ontologytutorial-noy-mcguinness-abstract.html

Noy, N. F., & Musen, M. A. (2001). Anchor-PROMPT: Using Non-Local Context for Semantic Matching, *Workshop on Ontologies and Information Sharing at the International Joint Conference on Artificial Intelligence*. Seattle.

Nwana, H. (1996). Software Agents: An Overview, *Knowledge Engineering Review*. [S.l.]: Cambridge University Press.

OCDE (2006). Reviews of National Policies for Education - Tertiary Education in Portugal - Examiners' Report, *Directorate for Education - Education Committee - Organisation de Coopération et de Développement Economiques (Organisation for Economic Co-operation and Development)*. Centro Cultural de Belém, Lisbon, Portugal.

Oliveira, P. (2003). *Um Modelo baseado em Agentes Móveis para Procura de Informação em Sistemas de Informação Distribuídos, Dissertação de Doutoramento.* Universidade de Trás-os-Montes e Alto Douro, Vila Real.

Ontopia (2005a). The Ontopia Knowledge Suite. *Ontopia.* Acesso em: 05 de Agosto de 2005, Disponível em: http://www.ontopia.net/solutions/products.html

Ontopia (2005b). The Ontopia Omnigator. *Ontopia.* Acesso em: 06 de Agosto de 2005, Disponível em: http://www.ontopia.net/omnigator/

OntoWeb (2002). Summary of Proposal. *Ontology-based information exchange for knowledge management and electronic commerce*. Acesso em: 19 de Fevereiro de 2004, Disponível em: http://ontoweb.ontoware.org/About/SummaryOfProposal

OTK (2004). On-To-Knowledge-Project : Content-driven Knowledge-Management through Evolving Ontologies. *Information Society Technologies*. Acesso em: 20 de Fevereiro de 2004, Disponível em: http://www.ontoknowledge.org

Palloff, R. M., & Pratt, K. (2002). *Construindo Comunidades de Aprendizagem no Ciberespaço: Estratégias eficientes para salas de aula on-line*. Porto Alegre: Artmed.

Park, J., & Hunting, S. (2003). *XML Topic Maps: Creating and Using Topic Maps for the Web*. Boston: Addison Wesley.

Payne, M., & Peacock, D. (2004). Collection Object to Learning Object. In D. Bearman & J. Trant (Eds.), *Museums and the Web*. USA: Archives and Museum Informatics.

Paz, F., Teixeira, P., Librelotto, G. R., Lopes, S. C., & Henriques, P. R. (2005). XML Topic Maps e Mapas de Conceitos, *Actas XATA2005 – XML: Aplicações e Tecnologias Associadas*. Braga.

Pepper, S. (2000). The TAO of Topic Maps - finding the way in the age of infoglut. *Ontopia.* Acesso em: 27 de Abril de 2005, Disponível em: http://www.ontopia.net/topicmaps/materials/tao.html

Pepper, S., & G., M. (2001). XML Topic Maps (XTM) 1.0. *TopicMaps.Org Specification.* Acesso em: 12 de Maio de 2005, Disponível em: http://www.topicmaps.org/xtm/1.0/

Philips, J. J., & Philips, P. P. (2002). Technology's return on investment. *Advances for Developing Human Resources, 4*(4), 512-532.

PHP (2005). PHP: Hypertext Preprocessor. *The PHP Group*. Acesso em: 24 de Janeiro de 2005, Disponível em: http://www.php.net/

Pimentel, M. G. C., Teixeira, C. A. C., & Santanche, A. (2000). *XML: Explorando suas Aplicações na Web*. Artigo apresentado em: Congresso da Sociedade Brasileira de Computação - Actas da XIX Jornada de Actualização em Informática, Curitiba.

Polikoff, I., & Allemang, D. (2003). TopQuadrant Technology Briefing - Semantic Integration: Strategies and Tools. *TopQuadrant, Inc.* Acesso em: 14 de Março de 2005, Disponível em: http://www.topquadrant.com/

Pollock, J. T., & Hodgson, R. (2004). *Adaptive Information: Improving Business Through Semantic Interoperability, Grid Computing, and Enterprise Integration. Wiley Series in Systems Engineering and Management*. New Jersey: John Wiley & Sons.

Polsani, P. R. (2003). Use and Abuse of Reusable Learning Objects. *Journal of Digital Information*. Acesso em: 25 de Novembro de 2005, Disponível em: http://jodi.ecs.soton.ac.uk/Articles/v03/i04/Polsani/

POSC (2006). Programa Operacional Sociedade do Conhecimento. *Unidade de Missão, Inovação e Conhecimento (UMIC) - Agência para a Sociedade do Conhecimento.* Acesso em: 20 de Abril de 2006, Disponível em: http://www.posc.mctes.pt

Preto, G., Tanaka, K., & Campos, M. (1999). *MetaSig: Metadados para Aplicações de Sistemas de Informação Geográficos*. [S.l.]: Instituto Militar de Engenharia - Departamento de Engenharia de Sistemas.

Prud'hommeaux, E., & Seaborne, A. (2004). SPARQL Query Language for RDF. *W3C Working Draft.* Acesso em: 23 de Setembro de 2006, Disponível em: http://www.w3.org/TR/2004/WD-rdf-sparql-query-20041012/

Prud'hommeaux, E., & Seaborne, A. (2006). SPARQL Query Language for RDF. *W3C Working Draft.* Acesso em: 26 de Dezembro de 2006, Disponível em: http://www.w3.org/TR/2006/WD-rdf-sparql-query-20061004/

Rabelo, R. J. (2002). Sistemas Multi-agentes. Acesso em: 07 de Setembro de 2004, Disponível em: http://www.das.ufsc.br/~rabelo

Ramalho, F. d. S. Web Semântica. *I Workshop de Dissertações da COPIN, Universidade Federal de Campina Grande / UFCG.* Acesso em: 28 de Maio de 2004, Disponível em: http://www.dsc.ufcg.edu.br/~copin/eventos/IWDCOPIN/IWDCOPIN.htm

Ramalho, J. C., Henriques, P. R., & Librelotto, G. R. (2004). ADRIAN: sistema de suporte à produção de conteúdos. *Universidade do Minho*. Acesso em: Disponível em: https://repositorium.sdum.uminho.pt/bitstream/1822/733/1/RHL04-eLes.pdf

Rath, H. H. (2003). White Paper: The Topic Maps Handbook. *Empolis - Arvato Knowledge Management*. Acesso em: 25 de Outubro de 2005, Disponível em: http://www.itee.uq.edu.au/~infs4206/Metamodeling/_Resources/empolistopicmapwhitepaper_eng.pdf

Recursion (2003). Voyager ORB Developer's Guide. *Recursion Software, Inc.* Acesso em: 30 de Julho de 2005, Disponível em: http://www.recursionsw.com/Products/voyager.html

Rezende, B. V., & Bax, M. P. (2002). Geração automática de metadados para documentos técnicos científicos, *XXV Congresso Brasileiro de Ciências da Comunicação*. Salvador/BA: INTERCOM – Sociedade Brasileira de Estudos Interdisciplinares da Comunicação.

Romiszowski, A. (2003). O futuro de E-Learning como inovação educacional: Fatores influenciando o sucesso ou fracasso de projetos, *Revista Brasileira de Aprendizagem Aberta e a Distância*.

Rosenberg, M. J. (2001). *e-Learning: Strategies for Delivering Knowledge in the Digital Age*. [S.l.]: McGraw-Hill.

RuleML. (2006). Rule Markup Language. *Rule Markup Initiative*. Acesso em: 30 de Outubro de 2006, Disponível em: http://www.ruleml.org

Santos, A. (2000). *Ensino a Distância & Tecnologias de Informação - e-learning*. [Lisboa]: FCA - Editora de Informática.

Santos, P. (2006, Dezembro). Pesquisas Web mais eficazes: Como encontrar. *Soluções, Exame Informática, Edimpresa Editora, 138*.

Santos, P. (2007, Janeiro). Pesquisas Web mais eficazes: Como ser encontrado. *Soluções, Exame Informática, Edimpresa Editora, 138*.

Schank, R. (1993). Learning Via Multimedia Computers, *Communications of the ACM* (Vol. 36): Technology in Education.

ScholOnto (2004). Scholarly Ontologies Project. *Distributed Information Management Programme* Acesso em: 22 de Agosto de 2004, Disponível em: http://kmi.open.ac.uk/projects/scholonto/

Schreiber, G., Wielinga, B., Akkermans, H., Velde, W. V. d., & Anjewierden, A. (1994). CML: the CommonKADS conceptual modeling language. In L. Steels, G. Schreiber & W. V. d. Velde (Eds.), *A Future for Knowledge Acquisition, Proceedings 8th European Knowledge Acquisition Workshop (EKAW94) - Lecture Notes in Artificial Intelligence 867*. Hoegaarden: Springer-Verlag.

Schwarz, S. (2005). RDFhomepage Project. Acesso em: 08 de Maio de 2005, Disponível em: http://rdfhomepage.opendfki.de/

Seaborne, A. (2004). RDQL - A Query Language for RDF. *W3C Member Submission*. Acesso em: 04 de Outubro de 2006, Disponível em: http://www.w3.org/Submission/2004/SUBM-RDQL-20040109/

Shen, Z., Shi, Y., & Xu, G. A Learning Resource Metadata Management System Based on LOM Specification. *Computer Science Department Tsinghua University*. Acesso em: 18 de Dezembro de 2004, Disponível em: http://www.cs.ucsb.edu/~szn/LRMMS.pdf

Sherman, C., & Price, G. (2001). The Invisible Web: Uncovering Information Sources Search Engines Can´t See. Medford, New Jersey: CyberAge Books.

Shi, R., Mary, K., & Zubair, M. (2003). *Automatic Metadata Discovery from non-cooperative Digital Librariesitle*. USA: Dep. of Computer Science, Old Dominion University.

Silva, A. (1999). *Agentes de Software na Internet – A próxima geração de aplicações*. [Lisboa]: Edições Centro Atlântico.

Silva, G. H. (2004). Usando o Jena para consultas na Web Semântica. *Projecto BIBO – Buscas Inteligentes Baseadas em Ontologias, Universidade de São Paulo*. Acesso em: 04 de Setembro de 2005, Disponível em: http://www.linux.ime.usp.br/~ghsilva/websemantica/

SingCore (2001). SingCore Application profile, *eLearning Standards Seminar - Practical Considerations for eLearning Standards Implementation* NUS: Information Technology Standards Committee.

Singh, M. P. (2002, Setembro/Outubro). Deep Web Structure. *IEEE Internet Computing*.

Sirin, E., Parsia, B., Grau, B. C., Kalyanpur, A., & Katz, Y. (2006). Pellet: A practical OWL-DL reasoner. *Journal of Web Semantics*.

Smith, J. R. (2006). Multimedia Annotations on the Semantic Web, Standards. *IEEE multimédia*. Acesso em: 23 de Dezembro de 2006, Disponível em: http://www.cs.vu.nl/~guus/papers/Stamou06a.pdf

Smith, P. L., & Ragan, T. J. (1999). *Instructional Design* (2.ª ed.). The University of Oklahoma, New York: John Wiley & Sons, Inc.

Spivack, N. (2004). Minding the Planet: From Semantic Web to Global Mind. *Radar Networks*. Acesso em: 20 de Fevereiro de 2005, Disponível em: http://novaspivack.typepad.com/nova_spivacks_weblog/2004/06/minding_the_pla.html

Staab, S. (2000). MPEG-7 - Semantic Annotation of Motion Pictures: Intelligent Systems on the World Wide Web - 11 MPEG-7. *Institute for Applied Computer Science and Formal Description Methods (AIFB) Karlsruhe University*. Acesso em: 01 de Outubro de 2004, Disponível em: http://www.aifb.uni-karlsruhe.de/WBS/sst/Teaching/

Staab, S., & Studer, R. (2004). Handbook on Ontologies, *International Handbooks on Information Systems*. Berlin: Springer-Verlag.

Stamou, G. (2006). SWBPD MM Task Force Description - Multimedia Annotation on the Semantic Web (MM). *W3C Semantic Web Activity*. Acesso em: 20 de Dezembro de 2006, Disponível em: http://www.w3.org/2005/Incubator/mmsem/

Stojanovic, L., Staab, S., & Studer, R. (2001). eLearning based on the Semantic Web. *Institute of Applied Informatics and Formal Description Methods*. Acesso em: 02 de Outubro de 2004, Disponível em: http://www.aifb.uni-karlsruhe.de/WBS/Publ/2001/WebNet_lstsstrst_2001.pdf

Stork, H.-G. (2002). The Semantic Web – A Web of knowledge and services, *Workshop on DB/IS Research for Semantic Web and Enterprises*. Georgia: Large Scale Distributed Information Systems.

Strickland, A. W. (2006). ADDIE. *Idaho State University College of Education Science, Math & Technology Education*. Acesso em: 05 de Setembro de 2006, Disponível em: http://ed.isu.edu/addie/index.html

Sullivan, D. (2002a). How Search Engines Work. *Search Engine Watch*. Acesso em: 16 de Novembro de 2004, Disponível em: http://searchenginewatch.com/webmasters/print.php/34751_2168031

Sullivan, D. (2002b). How To Use HTML Meta Tags. *Search Engine Watch*. Acesso em: 16 de Novembro de 2004, Disponível em: http://searchenginewatch.com/webmasters/print.php/34751_2167931

SUMO (2005). Suggested Upper Merged Ontology (SUMO). Acesso em: 10 de Maio de 2005, Disponível em: http://www.ontologyportal.org/

Swartout, W., & Tate, A. (1999). Ontologies. *IEEE Inteligent Systems and their applications, 14*(1).

SWCP. (2006). Data - Information - Knowledge. *The Semantic Web Community Portal*. Acesso em: 26 de Dezembro de 2006, Disponível em: http://www.semanticweb.org/about.html

Takasu, A. (2003). Bibliographic attribute extraction from erroneous references based on a statistical model, *Proceedings of the Third ACM/IEEE-CS Joint Conference on Digital Libraries* (pp. 49-60). New York: ACM Press.

TEK (2006). 16º Congresso APDC – Pesquisa continua em crescimento com forte potencial de monetização. *Casa dos Bits – TEKSapo*. Acesso em: 15 de Novembro de 2006, Disponível em: http://tek.sapo.pt/4M0/706398.html

Thompson, J. (2004, Maio). Features: Semantic Web – Strainght Talking. *PC Pro*.

TM4J (2005). Topic Maps for Java. *TM4J*. Acesso em: 03 de Outubro de 2005, Disponível em: http://tm4j.org/

Torres, J., Dodero, J., Diaz, P., & Ignacio, A. (2006). Diseño Instruccional en Internet y los Lenguages de Modelado Educativo. In L. Alonso, L. González, B. Manjón & M. Nistal (Eds.), *Proceedings of the 8th International Symposium on Computers in Education (SIIE'06)* (Vol. 2). Leon: Universidad de Leon.

Tsarkov, D., & Horrocks, I. (2004). Fact++. *University of Manchester*. Acesso em: 30 de Setembro de 2006, Disponível em: http://owl.man.ac.uk/factplusplus/

UMIC (2006b). TIC nas Escolas. *Edição UMIC - Ministério da Ciência, Tecnologia e Ensino Superior (MCTES)*. Acesso em: 01 de Novembro de 2006, Disponível em: http://www.umic.pt/index.php?option=com_content&task=view&id=13&Itemid=86

UMIC (2003). Uma Nova Dimensão de Oportunidades – Plano de Acção para a Sociedade da Informação – Portugal em Acção. *Edição UMIC - Presidência do Conselho de Ministros* Acesso em: 15 de Novembro de 2005, Disponível em: http://www.umic.pcm.gov.pt/UMIC/CentrodeRecursos/Publicacoes/si_publicacao.htm

UMIC (2006a). Unidade de Missão, Inovação e Conhecimento - Agência para a Sociedade do Conhecimento. *Ministério da Ciência e do Ensino Superior (MCTES)*. Acesso em: 30 de Maio de 2006, Disponível em: http://www.umic.pt

UNESCO ([1973]). *Guidelines for the establishment and development of monolingual thesauri*. Paris: The Organization United Nations Educational, Scientific and Cultural Organization.

Unicode (2000). The Unicode Standard, Version 3. *The Unicode Consortium, Addison-Wesley*. Acesso em: 13 de Janeiro de 2006, Disponível em: http://www.unicode.org/unicode/standard/versions/

Uschold, M., & Gruninger, M. (1996). Ontologies: principles, methods an applications. *Knowledge Engineering Review, 11*(2).

Van-Heijist, G., Schreiber, A. T., & Wielinga, B. J. (1997). Using explicit ontologies in KBS development. *International Journal of Human-Computer Studies, 46*(2-3), 183-192.

W3C (2001). Semantic Web Activity. *World Wide Web Consortium*. Acesso em: 10 de Outubro de 2003, Disponível em: http://www.w3.org/2001/sw/

W3C (2002). User Agent Accessibility Guidelines 1.0. *World Wide Web Consortium*. Acesso em: 10 de Novembro de 2004, Disponível em: http://www.w3.org/TR/WAI-USERAGENT/cover.html

W3C (2004). World Wide Web Consortium Issues RDF and OWL Recommendations. *World Wide Web Consortium*. Acesso em: 24 de Agosto de 2004, Disponível em: http://www.w3.org/2004/01/sws-pressrelease.html.en

W3C (2006). Semantic Web - Groups. *World Wide Web Consortium*. Acesso em: 10 de Dezembro de 2006, Disponível em: http://www.w3.org/2001/sw/#groups

Weiser, M. (1991). The Computer for the Twenty-First Century, *Scientific American* (pp. 94-104).

Wiley, D. (2002). Learning Objects Need Instructional Design Theory. In A. Rossett (Ed.), *The ASTD E-Learning Handbook: Best Practices, Strategies and Cases Studies for an emerging field*. New York: McGraw-Hill.

Willis, J. (1995). A recursive, reflective instructional design model based on constructivist-interpretivist theory, *Journal of Educational Technology* (pp. 5-23). Englewood Cliffs, NJ: Educational Technology Publications.

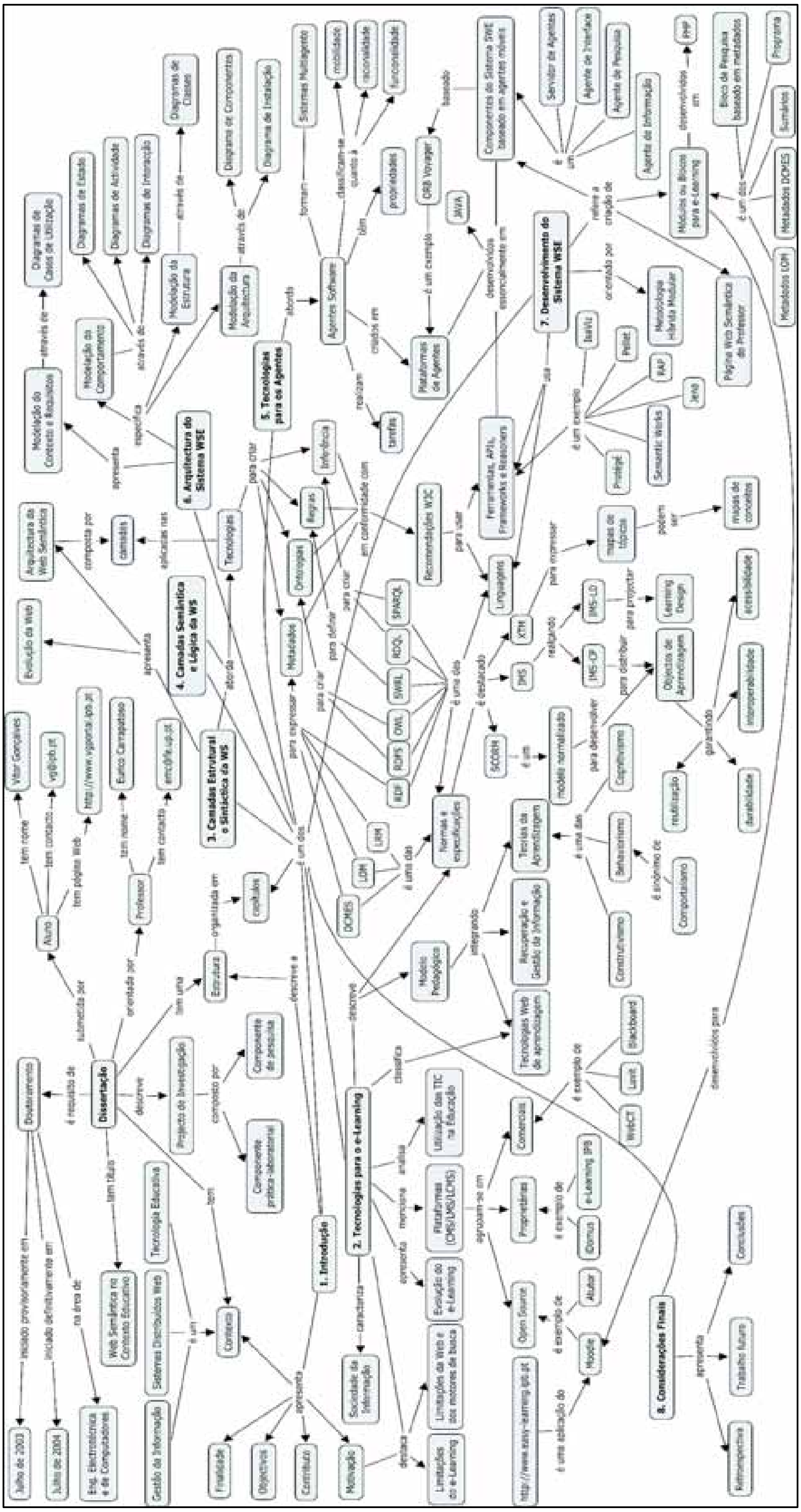
WISE (2002). Review Of Knowledge Management Tools. *Web-enabled Information Services for Engineering, The WISE Consortium*. Acesso em: 23 de Janeiro de 2004, Disponível em: http://www.eurisco.org/Wise/

Yee, R. (2001). The sea change of the Web: What is the Second-Generation, Semantic Web?, *IST-Interactive University Berkeley Computing & Communications* (Vol. 11).

Zenger, J., & Uehlein, C. (2001). Why Blended will Win. *Training & Development, 55*(8), 54-60.

Anexos

Anexo A – Mapa de Conceitos da Tese



Anexo B – Cérebro Global

Introdução:

A informação, outrora escassa, é hoje tão abundante que pode conduzir à desinformação. Esta avalanche de informação é cada vez mais preocupante (Garcia, 2002). Tal como um Website, cada ser humano só por si é uma fonte de conhecimento para outro. Não obstante, acreditar que dispor de toda a informação que queiramos é sinónimo de mais conhecimento seria uma quimera. Passar de uma Sociedade da Informação para uma Sociedade do Conhecimento que permita seleccionar e relacionar a informação imprescindível com vista à geração de conhecimento e de inteligência é talvez o desafio actual mais importante. Para tal, exige-se uma Web que "pense" connosco.

A Web actual é como que uma grande biblioteca desordenada porque não existem bibliotecários a estruturá-la e organizá-la ou, quando existem, nem sempre seguem as mesmas normas de estruturação e organização dos conteúdos. A Web Semântica será como um espaço repleto de recursos devidamente indexados e catalogados em que a estrutura de qualquer conteúdo terá significado tanto para os humanos, como para as máquinas.

Diversos autores têm vindo a defender que existem muitas semelhanças entre a Internet e o cérebro humano. Neste sentido podemos afirmar que a emergente Web Semântica contribuirá significativamente para aumentar essa semelhança e constitui o primeiro passo rumo a uma Web mais inteligente. Parece uma ideia para um filme? Talvez não, é uma visão optimista sobre o futuro da Internet e do impacto que a Web Semântica poderá representar.

Se relembrarmos a mitologia, o Deus Apolo resolveu construir um templo que servisse de oráculo para os homens. Falamos do Oráculo de Delfos que respondia a todo o tipo de perguntas dos humanos, como se de um extraordinário motor de busca inteligente se tratasse, proporcionando a resposta correcta à pergunta formulada (Garcia, 2002). Analogamente, tem vindo a ser idealizado um novo oráculo no ciberespaço o qual, em vez de nos disponibilizar as centenas de possíveis respostas à questão submetida, tal como acontece nos actuais motores de busca, nos apresentará as respostas o mais possível filtradas e precisas, separando o trigo do joio.

No início deste novo milénio, podemos afirmar que existiam duas visões para converter a actual Internet numa Rede mais inteligente: uma via de investigação aposta num sistema que permita que a procura e recuperação de informação sejam mais intuitivas e precisas através da aplicação da semântica, de ontologias e de agentes inteligentes (Tim

Berners-Lee, Hans-Georg Stork e James Hendler, entre outros) e a outra baseia-se na possibilidade dos servidores poderem ser capazes de aprender o percurso que fazem os cibernautas com vista a que a rede possa funcionar como um Cérebro Global (Francis Heylighen, Johan Bollen, Valentin Turchin e Cliff Joslyn, entre outros). Finalmente, com vista à simbiose desta duas visões no âmbito da estruturação e partilha do conhecimento, os Sistemas de Conhecimento Distribuído assumem particular importância. Embora diferentes, estas visões podem assim ser complementares.

Mas, o desenvolvimento da Internet já não será fruto apenas de decisões tecnológicas das Ciências da Computação e da Inteligência Artificial. As Ciências Sociais, a Psicologia e a Sociologia, entre outras, contribuirão significativamente para a concepção de novos modelos para a Web.

Não obstante, é necessário atenuar as atitudes pessimistas em relação às consequências que estas visões possam implicar. É imprescindível assumir que não estamos a assistir ao desenvolvimento de um cérebro que vai controlar as nossas vidas. A ideia base é implementar um conjunto de assistentes pessoais cuja finalidade será recomendar e aconselhar o utilizador de acordo com o seu perfil, mas para tal a informação Web deverá ser reorganizada e estruturada com vista a facilitar a sua manipulação pelas máquinas e seus agentes inteligentes. Ou seja, há que preparar e ensinar a Web para realizar tarefas simples, monótonas e repetitivas que requereriam de nós humanos uma inteligência primária.

Cérebro Global:

Em primeira instância, a ideia do "Cérebro Global" ou "Mente Global", incide na comparação da sociedade com um organismo. Embora não seja um termo ou ideia recente, ganhou novo fôlego com o aparecimento da Internet e, em particular, da Web. Francis Heylighen e Johan Bollen têm à sua disposição a Web, um sistema de inteligência distribuída de carácter global, o que lhes confere uma grande vantagem sobre os seus antecessores, no sentido de desenvolver uma filosofia sistémica evolutiva da mente global. O projecto "Principia Cibernética" (*http://pespmc1.vub.ac.be/default.html*) materializa esta linha de investigação.

Podemos afirmar que um dos desafios actuais é converter a Web numa mente global, isto é, num sistema nervoso central composto de milhares de milhões de neurónios que se abastecem da inteligência distribuída da Rede. Por conseguinte, a inteligência colectiva de todos os membros da Rede Internet constituirá o Cérebro Global.

Uma via para o conseguir poderá passar pela visão de Tim Berners-Lee denominada "Web Semântica", onde a informação contida na Web seria compreensível não só pelos humanos, mas também pelas máquinas que a estruturariam, reorganizariam por forma a disponibilizá-la de acordo com as necessidades (explícitas ou não) dos utilizadores (Berners-Lee, 2001).

Uma Web com estas características terá de evoluir ela própria, fruto da compreensão das suas próprias ligações e da actividade dos utilizadores. Neste sentido, a aplicação da ideia do Cérebro Global à Web implicará a reconstrução das ligações de acordo com as necessidades do utilizador. O servidor Web (*http://pespmc1.vub.ac.be/SERVER.html*) do projecto Principia Cibernética, desenvolvido por Heylighen e Bollen, tenta materializar este modelo de cérebro global ao permitir a actualização e recomposição constante das suas ligações de acordo com as procuras do utilizador, eliminando as que vão perdendo actualidade pelo escasso ou nulo grau de consultas que recebem.

O servidor Web adjudica um agente inteligente a cada visitante cujo algoritmo permite detectar qual o percurso que este segue dentro da Web e memorizar o histórico do seu comportamento. Quando voltar a aceder ao servidor, o agente inteligente mostrará as páginas que supostamente interessam ao utilizador e ajustará a estrutura de ligações à esfera dos seus interesses. Se os agentes comunicarem entre si, poderão intercambiar informações com base no grau de utilização ou satisfação dos utilizadores sobre determinadas ligações.

A inteligência do cérebro global da Web não corresponderá apenas ao somatório das inteligências individuais da Rede, mas sim a uma inteligência colectiva em plena evolução.

Podemos afirmar que a Web é como uma criança que foi adquirindo informação, mas que continua ávida de conhecimento. Para que os agentes Web possam facilmente tomar decisões inteligentes, a Web precisa, primeiramente, de se ajustar aos interesses do utilizador para, posteriormente, favorecer a interpretação da informação disponível e a consequente geração de conhecimento.

O cérebro global parte do princípio de que a sociedade pode ser vista como um sistema baseado na inteligência colectiva de todos os seus membros, ou seja, uma estrutura que une cada nó (computador ligado à rede) como se fossem os neurónios de um cérebro.

Esta ideia baseia-se no conceito de "inteligência partilhada", segundo o qual a inteligência do grupo é maior que a soma das inteligências dos seus membros. A Internet viabiliza este conceito com uma dimensão muito maior do que qualquer outra infra-estrutura

de comunicação anterior. Podemos também afirmar que as redes P2P (*Peer-to-Peer*) se assumem como uma das tecnologias que favorecem a concretização desta ideia.

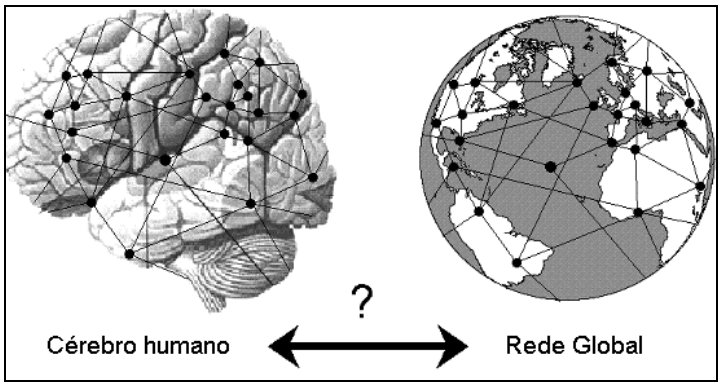


Figura 128 – Analogia entre Cérebro Humano e Rede Internet

No âmbito do Projecto Principia Cibernética, o cérebro global está organizado como uma rede de associações, tal como se se tratasse de neurónios interligados que se desenvolvem mediante aprendizagem (Bollen e Heylighen, 2005). Sempre que se encontram as mesmas ligações repetidamente, o cérebro global aprende. Se pensar é a activação sucessiva de neurónios quando nos concentramos em determinados assuntos, e se essa activação provoca ligações com outros neurónios que, por sua vez, activam outros conceitos, então na Web também se podem estabelecer novas ligações entre páginas consoante o percurso efectuado pelo utilizador.

Esta linha de investigação visa aplicar o modelo cerebral à Web para demonstrar que esta pode ser uma rede associativa. Ou seja, através de agentes inteligentes, o utilizador pode ir deixando pistas para que em futuras pesquisas obtenha da rede documentos mais próximos dos seus interesses.

Se atribuirmos um algoritmo a cada utilizador de modo a sondar o caminho que percorre dentro de um Website e memorizar o histórico do seu comportamento, da próxima vez o *browser* só lhe mostrará as páginas que lhe interessam, ajustando as ligações com vista a guiá-lo no contexto dos seus interesses. Consequentemente, torna-se possível detectar a redundância ou escassez de informação e conhecimento nas áreas de interesse de cada utilizador.

Quando um utilizador activa uma hiperligação, confere um grau de relevância a essa ligação. Para saber qual o grau de relevância pode pedir-se ao utilizador que pontue essa página (sondagem) ou observar o que faz com essa hiperligação (adiciona-a aos favoritos, guarda-a no disco rígido ou imprime?) e quanto tempo despende com ela (*cookies*). Através

de *cookies* inteligentes é recolhida informação sobre o utilizador com vista a traçar o seu percurso na rede e reconhecê-lo posteriormente. Esta última opção, uma vez que não requer o esforço do utilizador, é talvez mais útil, tanto mais que se baseia no tempo que passamos numa determinada página. Assim, quanto mais tempo o utilizador passe em frente a uma página, mais interesse demonstra nela, logo maior será a sua relevância.

Por conseguinte, os agentes inteligentes perceberão quais são os nossos interesses e, pouco a pouco, podem oferecer-nos alternativas. Francis Heylighen chega a afirmar que os agentes inteligentes encontram o que o utilizador pode necessitar, embora ainda não o tenha visto ou tentado procurar. Assim, a Web antecipa as nossas necessidades, devolvendo os resultados mais relevantes, mais usados e mais precisos, uma vez que correspondem às ligações mais activadas.

Obviamente que têm vindo a levantar-se algumas vozes contra este tipo de abordagem, uma vez que a privacidade do utilizador pode ser posta em causa. Mas, mesmo que houvesse uma suposta quebra de privacidade, os resultados não seriam suficientemente interessantes para valer a pena correr esse risco, desde que de forma controlada? Sendo uma ideia interessante, é necessário minimizar os problemas que dela possam decorrer, sem a limitar à identificação do percurso Web efectuado pelo utilizador. Ou seja, o percurso do utilizador é apenas mais um dos parâmetros que permitirão aos agentes inteligentes filtrar a informação de forma a responder mais fácil e precisamente aos requisitos do utilizador.

O funcionamento do cérebro global aplicado à estrutura da Internet pode traduzir-se na criação de novas hiperligações que se activam quando se detecta que o utilizador altera a sua intenção de pesquisa, de modo a que as ligações se actualizem ou eliminem automaticamente consoante o seu uso. Assim, as ligações da Internet na perspectiva do cérebro global assemelhar-se-ão à estrutura de um cérebro humano, porque se criam novas ligações em função do uso tal como os neurónios. Segundo Francis Heylighen, qualquer questão que surja por parte do utilizador deixará de ser uma tarefa árdua porque a rede organizar-se-á e adaptar-se-á ao que o utilizador espera dela.

O Cérebro Global pretende disponibilizar uma Web que “pensa e aprende”. Uma Web que genericamente sabe de que informação necessitamos, mesmo que nós próprios não saibamos o que procuramos ou que palavras-chave necessitamos para o encontrar.

Observando o funcionamento do cérebro humano, sabemos que a recuperação de dados é flexível e eficiente, que a nossa massa encefálica é capaz de recolher da nossa memória a

informação ou a resposta que necessitamos. Os nossos neurónios ligam-se entre si mediante sinapses (*synapses*) e a sua união reforça-se pelo uso. Este exemplo de aprendizagem baseia-se na chamada regra de Hebb: Se dois neurónios se activam sucessivamente, o seu poder de interligação aumenta.

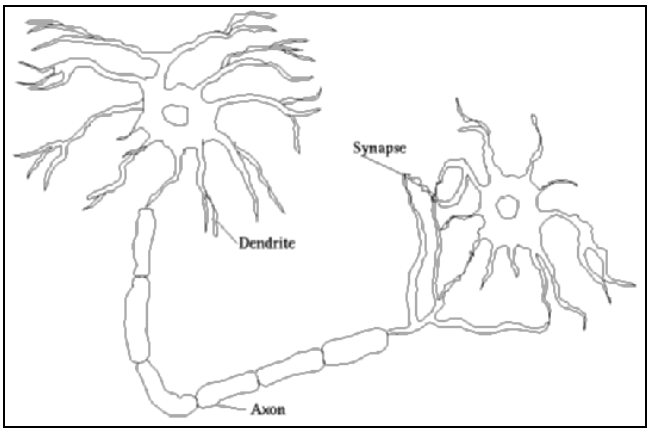


Figura 129 – Regra de Hebb

Da mesma forma, se duas páginas Web são consultadas pelo mesmo utilizador num breve intervalo de tempo, a hiperligação existente entre elas assume um peso mais importante ou dá origem a uma nova ligação. Pelo contrário, ligações que não são consultadas assiduamente, podem desaparecer. Inconscientemente, estamos a seleccionar de entre os milhares de milhões de páginas Web aquelas que mais nos interessam.

Em suma, os agentes inteligentes recordarão as páginas Web mais usadas e, através da activação sucessiva de ligações, encontrarão outras páginas Web associadas ou relacionadas que nos recomendarão caso sejam do nosso interesse. Um exemplo básico disto pode ser já visualizado no Website *Amazon*: ao procurar um livro, obtemos também recomendações relacionadas com esse livro feitas por outros utilizadores, para além de críticas ou outras páginas relacionadas.

Em resumo, a Web transformar-se-ia num sistema inteligente que aprende com os seus utilizadores, reage aos seus desejos, reorganiza-se, interpreta, deduz e associa, enfim, pensa pelos seus utilizadores. Esta Web constitui assim um Cérebro Global, uma extensão do nosso próprio cérebro.

Mas como ensinar uma rede de hipertexto?

O facto de uma rede de hipertexto apresentar a informação em linguagem humana, limita o processo de recuperação de informação. Há que recorrer a um conjunto de regras de aprendizagem que, automaticamente (sem intervenção humana), reorganizem as redes de

hipertexto e racionalizem a sua estrutura, filtrando a informação. Estas regras de aprendizagem são três (Bollen e Heylighen, 2005):

a) Frequência: se um utilizador vai do Website A ao Website B a ligação entre eles reforça-se.

b) Transição: Se um utilizador vai do Website A ao Website B e deste ao C, cria-se uma ligação do Website A para o C.

c) Simetria: ao mesmo tempo que se reforça a ligação entre um Website A e um Website B, também se reforça a ligação no sentido inverso.

Estas três regras permitem estruturar a navegação do utilizador e determinar qual é o seu percurso, de modo a que o agente inteligente seja capaz de "adivinhar" quais os interesses do utilizador para que lhe possa oferecer novas ligações.

O Cérebro Global visa ordenar de outra forma os documentos da Web, de uma forma que seja mais útil ao utilizador e mais personalizada, baseando-se nas suas necessidades.

Embora inicialmente se pensasse que as redes de hipertexto facilitariam a obtenção de dados, actualmente perde-se muito tempo navegando à procura de informação e muitas das vezes o utilizador acaba por se desorientar. A ideia do Cérebro Global é criar um sistema estruturado a partir das ligações, do seu uso e da sua fiabilidade, compreendendo todo o tipo de documentos (texto HTML, texto do e-mail, pdf, multimédia, etc.) e em qualquer idioma. Uma das limitações desta linha de investigação é a utilização de páginas dinâmicas. Mas, num futuro próximo as bases de dados também indicarão aos agentes qual a informação que armazenam e auto-organizar-se-ão a partir do comportamento dos utilizadores. Alguns critérios poderão passar pela marcação personalizada dos conteúdos mais vistos e do tempo dedicado a cada um. Logo, os utilizadores estarão a dar pistas ao agente acerca dos seus gostos e preferências. Por conseguinte, cada acção do utilizador implica a reestruturação e reorganização da Rede, criando novas ligações. A Rede altera-se constantemente, tornando-a mais eficiente e inteligente. Desta forma, cada um de nós terá uma Internet personalizada.

Reconhecer o perfil do utilizador e ter informação sobre o que sabe e o que poderá vir a precisar saber é útil para a actuação dos agentes inteligentes e dos sistemas de conhecimento distribuído (DKS - *Distributed Knowledge Systems*). As arquitecturas de sistemas de conhecimento distribuído são desenvolvidas com base em normas e tecnologias Web para permitir a integração de (e acesso uniforme a) componentes de sistemas inteligentes, possibilitando que os agentes interajam com os recursos de informação existentes na rede.

Os DKS têm vindo a obter resultados tangíveis gerindo e gerando conhecimento na rede, tanto à escala individual como à organizacional. Por exemplo, os DKS permitem que o computador A possa recuperar o conhecimento do computador B, ou seja, se usar agentes inteligentes, o computador A não recupera apenas informação, mas sim conhecimento gerado a partir da associação das informações existentes no computador B. Atendendo a que a acção de pensar e de compreender implica a activação de neurónios e a associação entre conceitos, estes sistemas de conceito distribuído devem ser capazes de entender tais conceitos. Ou seja, devem ser capazes de entender o conteúdo das páginas Web para poderem proporcionar informação precisa ao utilizador. É aqui que entra a Web Semântica e os agentes inteligentes. A Web Semântica possibilita que os servidores compreendam o conteúdo que armazenam e distribuem na rede. Os agentes inteligentes são a sua ferramenta, devolvendo informação relevante aos utilizadores que dela necessitam e assistindo-os na tomada de decisão. Por conseguinte, uma das principais vantagens é que a informação que o utilizador recebe da Web inteligente é sempre filtrada, relevante e adequada às suas necessidades, evitando assim perder tempo a ver páginas Web que nada têm a ver com a temática que pretende, hiperligações que não levam a lado nenhum ou páginas que entretanto caducaram.

A Web Semântica assume-se como uma evolução da Internet rumo a um sistema automático de reconhecimento de dados que proporciona ao utilizador apenas a informação relevante.

Considerações Finais:

Expressões como “inteligência emergente” (Steven Johnson), “colectivos inteligentes” (Howard Rheingold), “cérebro global” (Francis Heylighen), “sociedade da mente” (Marvin Minsk), “inteligência conectiva” (Derrick de Kerckhove), “redes inteligentes” (Albert Barabasi), “inteligência colectiva” (Pierre Lévy) são cada vez mais referidas por teóricos reconhecidos. Todos eles apontam para uma mesma situação: estamos em rede, interligados com um número cada vez maior de pontos e a uma frequência crescente.

A Web não pode apenas armazenar a inteligência distribuída dos seus utilizadores, deverá conseguir aprender com o que os utilizadores fazem e comportar-se de acordo com essa aprendizagem.

Nesta perspectiva, a visão da Web Inteligente percebe-se em duas visões diferentes, mas complementares. Uma refere-se à analogia da Internet com o cérebro humano, a outra à

capacidade dos servidores Web compreenderem o significado de todos os conteúdos que armazenam e distribuem pela Rede, independentemente da sua forma (texto, imagem, gráficos, áudio ou vídeo). Para que se possa compreendê-los, será necessário estruturá-los e organizá-los semanticamente. Portanto, as metalinguagens semânticas (XML, RDF(S), OWL) assumem particular importância neste contexto. As metalinguagens semânticas permitem expressar, distribuir e partilhar formalmente o conhecimento necessário à interpretação e utilização inteligente da informação, independentemente dos receptores da informação serem humanos ou máquinas. Mas, para que os servidores possam negociar entre eles os conteúdos Web, tornar-se-á necessário desenvolver sistemas de agentes inteligentes. Referimo-nos pois ao projecto da Web Semântica.

A Web Semântica possibilita que os servidores compreendam o conteúdo que armazenam e disponibilizam na rede. Os agentes inteligentes tratam da recuperação da informação relevante aos utilizadores que dela necessitam e assistem-nos na tomada de decisão.

De referir que, no âmbito da estruturação do conhecimento, os Sistemas de Conhecimento Distribuído em articulação com a Web Semântica constituem uma peça essencial para o desenvolvimento da arquitectura para uma Web Inteligente.

Mas, se for possível perceber o comportamento do utilizador, a Web pode personalizar-se de forma a colocar em primeiro plano aquilo que supostamente o utilizador necessita. Assim, todos estes esforços apontarão para o funcionamento da Rede segundo um modelo “Cérebro Global”. Ou seja, a Web Semântica viabiliza a ideia do Cérebro Global.

O Cérebro Global não é mais do que uma metáfora para caracterizar uma rede inteligente formada por pessoas que detêm conhecimento e tecnologias de comunicação que permitem a interligação de todos (Heylighen, 2002).

Se os problemas se resolvem pensando, isto é, activando neurónios, então os resultados na Web Inteligente (segundo a perspectiva do Cérebro Global) obtêm-se activando conceitos relacionados entre si.

Através de um agente inteligente, é possível analisar o percurso de um utilizador através da Rede, e em função desse percurso, deduzir o que lhe interessa. A ideia é observar as ligações que um utilizador efectua entre as páginas Web. Consequentemente, a Web estará a aprender à medida que o utilizador navega, transformando-se numa rede associativa. O

algoritmo responsável por essa aprendizagem reforça as ligações que mais se utilizam em detrimento das que menos se utilizam.

Em suma, partindo da análise do comportamento global dos utilizadores e do comportamento de cada um, o Cérebro Global constitui uma verdadeira rede associativa.

Mas a ideia do Cérebro Global não se deverá esgotar aqui. Esta é apenas uma forma dos agentes inteligentes se aperceberem do perfil dos utilizadores (os seus gostos e preferências). O cérebro global deverá ser fruto da inteligência de todos os utilizadores da rede, beneficiando não só cada um individualmente, mas também os grupos, tais como empresas ou as escolas.

A actualidade é caracterizada pela incorporação espacial de computadores embutidos em todos os lugares, tal como perspectivava Weiser (1991) há mais de duas décadas atrás. Enfim, é a era da computação ubíqua em que vários computadores são operados por uma única pessoa.

A Internet é o cérebro e a Web é a sua mente (Spivack, 2004). A ideia do cérebro global assume que cada um de nós faz parte dele. A novidade está em adicionar as máquinas, ou melhor, os agentes de software, a esta realidade, dotando-os de capacidades de raciocínio próximas das dos humanos.

Por conseguinte, cada um de nós, inconscientemente ou não, faz parte desta mente global. A mente global está emergindo em torno de nós, e através de nós, é a nossa criação, mas pode ganhar alguma independência, desde que os agentes de software interpretem metadados e consigam dialogar e colaborar na resolução de tarefas sem intervenção humana directa (Spivack, 2004).

No entanto, tal como questiona Pierre Lèvy (1999): Seremos capazes de nos mobilizar a favor de um projecto de sociedade centrado na inteligência colectiva que favoreça a partilha livre de conhecimentos?

Anexo C – Aplicação das Tecnologias para a Web Semântica

Consideremos a seguinte tabela de dados (CDs ou DVDs de música) que poderia ser visualizada através de qualquer browser ou navegador na Web actual:

Título do Álbum	Artista	País	Editora	Preço	Ano
Evanescence Fallen	Evanescence	USA	Sony Music Entertainment	18.80	2004
Live Summer 2003	Robbie Williams	UK	Chrysalis Records Ltd	16.90	2003

Esta tabela de álbuns musicais poderia derivar do seguinte código HTML:

```
<html>
<head>
<title>Lista de Música</title>
</head>
<body>
<table border="1" cellpadding="0" cellspacing="0">
  <tr bgcolor="#CCCCCC">
    <td width="135" valign="top"><p><strong>Título do Álbum </strong></p></td>
    <td width="112" valign="top"><p><strong>Artista </strong></p></td>
    <td width="72" valign="top"><p><strong>País </strong></p></td>
    <td width="184" valign="top"><p><strong>Editora </strong></p></td>
    <td width="72" valign="top"><p><strong>Preço </strong></p></td>
    <td width="56" valign="top"><p><strong>Ano </strong></p></td>
  </tr>
  <tr>
    <td width="135" valign="top"><p>Evanescence Fallen </p></td>
    <td width="112" valign="top"><p>Evanescence </p></td>
    <td width="72" valign="top"><p>USA </p></td>
    <td width="184" valign="top"><p>Sony Music Entertainment </p></td>
    <td width="72" valign="top"><p>18.80 </p></td>
    <td width="56" valign="top"><p>2004 </p></td>
  </tr>
  <tr>
    <td width="135" valign="top"><p>Live Summer 2003 </p></td>
    <td width="112" valign="top"><p>Robbie Williams </p></td>
    <td width="72" valign="top"><p>UK </p></td>
    <td width="184" valign="top"><p>Chrysalis Records Ltd </p></td>
    <td width="72" valign="top"><p>16.90 </p></td>
    <td width="56" valign="top"><p>2003 </p></td>
  </tr>
</table>
</body>
</html>
```

Figura 130 – Álbuns de música: código HTML

A HTML não fornece a descrição do que significam esses dados, apenas se preocupa com a apresentação da informação. Para atribuir semântica a estes dados vamos aplicar as tecnologias para a Web Semântica a este documento.

A XML permite descrever a estrutura desse documento, separando os rótulos de dados dos rótulos de apresentação. Para além de permitir etiquetar as páginas Web, a XML fornece um formato de intercâmbio de dados.

```
<?xml version="1.0" encoding="ISO-8859-1" standalone="no"?>
<?xml-stylesheet type="text/xsl" href="exemplo.xsl"?>
<!DOCTYPE BIBLIOTECA SYSTEM "exemplo.dtd">
<MUSICA>
   <CD>
      <TITLE>Evanescence Fallen</TITLE>
      <ARTIST>Evanescence</ARTIST>
      <COUNTRY>USA</COUNTRY>
      <COMPANY>Sony Music Entertainment</COMPANY>
      <PRICE>18.80</PRICE>
      <YEAR>2004</YEAR>
   </CD>
   <CD>
      <TITLE>Live Summer 2003</TITLE>
      <ARTIST>Robbie Williams</ARTIST>
      <COUNTRY>UK</COUNTRY>
      <COMPANY>Chrysalis Records Ltd</COMPANY>
      <PRICE>16.90</PRICE>
      <YEAR>2003</YEAR>
   </CD>
.
.
.
</MUSICA>
```

Figura 131 – Álbuns de música: código XML

A estrutura do documento em XML resulta da criação de uma DTD ou de um XML *Schema*. A apresentação do documento ao utilizador resulta da utilização de CSS ou XSL.

Mas, e o significado ou a semântica? Para descrever a informação de forma não ambígua e com significado recorremos à tecnologia RDF:

```
<?xml version="1.0"?>
<rdf:RDF
xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:cd="http://www.exemplo.pt/cd">
<rdf:Description
 rdf:about="http://www.exemplo.pt/cd/Evanescence Fallen">
  <cd:artist>Evanescence</cd:artist>
  <cd:country>USA</cd:country>
  <cd:company>Sony Music Entertainment</cd:company>
  <cd:price>18.80</cd:price>
  <cd:year>2004</cd:year>
</rdf:Description>
<rdf:Description
 rdf:about="http://www.exemplo.pt/cd/Live Summer 2003">
  <cd:artist>Robbie Williams</cd:artist>
  <cd:country>UK</cd:country>
  <cd:company>Chrysalis Records Ltd</cd:company>
  <cd:price>16.90</cd:price>
  <cd:year>2003</cd:year>
</rdf:Description>
.
.
.
</rdf:RDF>
```

Figura 132 – Álbuns de música: código RDF

A RDF descreve os recursos DVDs inerentes à tabela apresentada inicialmente, permitindo que um agente de software perceba que o valor de uma propriedade que representa o artista é uma referência a uma pessoa e não a uma localidade ou veículo. Não obstante, necessitamos de um esquema que valide as afirmações expressas em RDF. Em suma, necessitamos do esquema RDFS ou de outra proposta ontológica semanticamente mais expressiva.

Com o recurso a uma ferramenta de criação e edição de ontologias como a Protégé podemos especificar uma ontologia simples em RDF(S). A Figura 133 ilustra o ambiente da ferramenta Protégé.

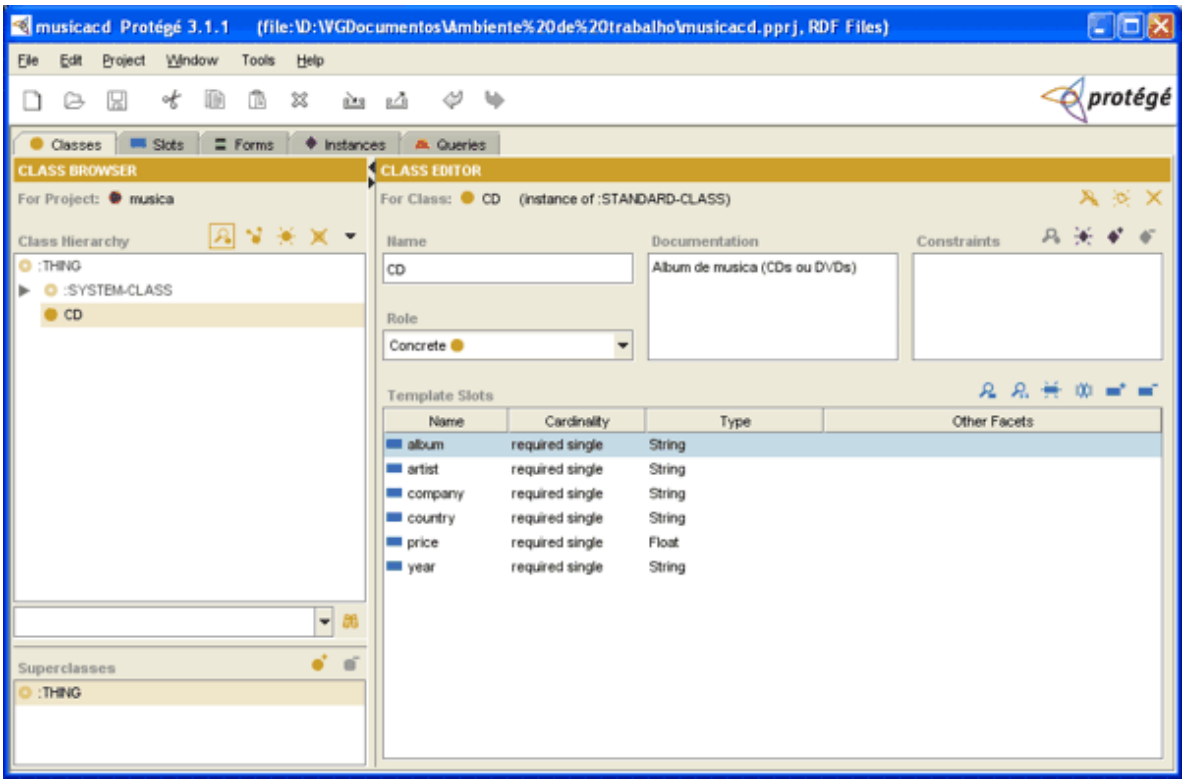


Figura 133 – Álbuns de música: projecto no Protégé

O resultado do projecto de uma ontologia simples em RDFS para o exemplo em questão pode resumir-se a um ficheiro RDFS para especificar as classes e um ficheiro RDF para as instâncias. A Figura 134 corresponde ao código do esquema RDFS e a Figura 135 ao código RDF.

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE rdf:RDF [
        <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
        <!ENTITY a 'http://protege.stanford.edu/system#'>
        <!ENTITY cd 'http://www.vgportal.ipb.pt/cd#'>
        <!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'>
]>
<rdf:RDF xmlns:rdf="&rdf;"
        xmlns:a="&a;"
        xmlns:rdfs="&rdfs;"
        xmlns:cd="&cd;">
<rdfs:Class rdf:about="&cd;CD"
        rdfs:comment="Album de musica (CDs ou DVDs)"
        rdfs:label="CD">
       <rdfs:subClassOf rdf:resource="&rdfs;Resource"/>
</rdfs:Class>
<rdf:Property rdf:about="&cd;album"
        a:maxCardinality="1"
        a:minCardinality="1"
        rdfs:comment="Titulo do album (CD ou DVD de musica)"
        rdfs:label="album">
       <rdfs:domain rdf:resource="&cd;CD"/>
       <rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property rdf:about="&cd;artist"
        a:maxCardinality="1"
        a:minCardinality="1"
        rdfs:comment="Interprete ou banda musical"
        rdfs:label="artist">
       <rdfs:domain rdf:resource="&cd;CD"/>
       <rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property rdf:about="&cd;company"
        a:maxCardinality="1"
        a:minCardinality="1"
        rdfs:comment="Editora discografica"
        rdfs:label="company">
       <rdfs:domain rdf:resource="&cd;CD"/>
       <rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property rdf:about="&cd;country"
        a:maxCardinality="1"
        a:minCardinality="1"
        rdfs:comment="Pais do artista, interprete ou banda musical"
        rdfs:label="country">
       <rdfs:domain rdf:resource="&cd;CD"/>
       <rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property rdf:about="&cd;price"
        a:maxCardinality="1"
        a:minCardinality="1"
        a:range="float"
        rdfs:comment="Preco do album musical"
        rdfs:label="price">
       <rdfs:domain rdf:resource="&cd;CD"/>
       <rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>
<rdf:Property rdf:about="&cd;year"
        a:maxCardinality="1"
        a:minCardinality="1"
        rdfs:comment="Ano de lancamento"
        rdfs:label="year">
       <rdfs:domain rdf:resource="&cd;CD"/>
       <rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>
</rdf:RDF>
```

Figura 134 – Álbuns de música: ficheiro RDFS

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE rdf:RDF [
        <!ENTITY rdf 'http://www.w3.org/1999/02/22-rdf-syntax-ns#'>
        <!ENTITY rdfs 'http://www.w3.org/2000/01/rdf-schema#'>
        <!ENTITY cd 'http://www.vgportal.ipb.pt/cd#'>
]>
<rdf:RDF xmlns:rdf="&rdf;"
        xmlns:cd="&cd;"
        xmlns:rdfs="&rdfs;">
<cd:CD rdf:about="&cd;cd01"
        cd:album="Evanescence Fallen"
        cd:artist="Evanescence"
        cd:company="Sony Music Entertainment"
        cd:country="USA"
        cd:price="18.80"
        cd:year="2004"
        rdfs:label="Evanescence Fallen"/>
<cd:CD rdf:about="&cd;cd02"
        cd:album="Live Summer 2003"
        cd:artist="Robbie Williams"
        cd:company="Chrysalis Records Ltd"
        cd:country="UK"
        cd:price="16.90"
        cd:year="2003"
        rdfs:label="Live Summer 2003"/>
</rdf:RDF>
```

Figura 135 – Álbuns de música: ficheiro RDF

Esta solução poderia evoluir para uma ontologia mais expressiva em OWL. Para além de especificarmos 3 classes, aproveitamos também para alterar as propriedades “artist” e “company”, criando as relações “interpretado_por” e “editado_por”, tal como se pode verificar na Figura 136.

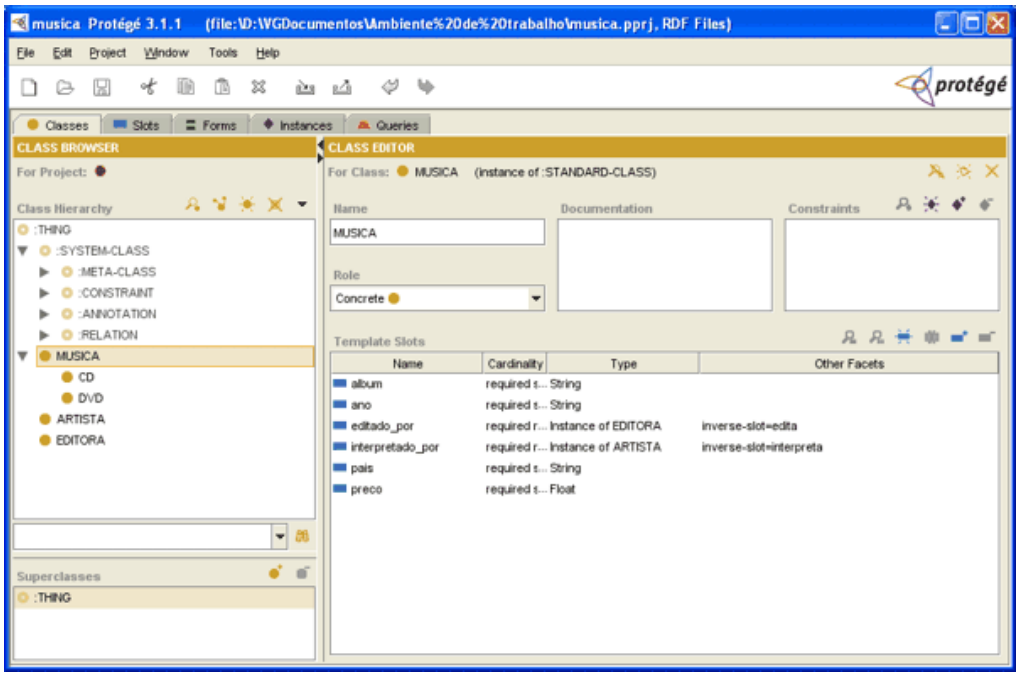


Figura 136 – Álbuns de música: optimização do projecto no Protégé

A exportação para OWL do projecto ilustrado na Figura 136 é apresentada a seguir:

```
<?xml version="1.0"?>
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://www.owl-ontologies.com/unnamed.owl#"
  xml:base="http://www.owl-ontologies.com/unnamed.owl">
  <owl:Ontology rdf:about=""/>
  <owl:Class rdf:ID="DVD">
    <rdfs:subClassOf>
      <owl:Class rdf:ID="MUSICA"/>
    </rdfs:subClassOf>
  </owl:Class>
  <owl:Class rdf:ID="EDITORA"/>
  <owl:Class rdf:ID="CD">
    <rdfs:subClassOf rdf:resource="#MUSICA"/>
  </owl:Class>
  <owl:Class rdf:ID="ARTISTA"/>
  <owl:ObjectProperty rdf:ID="edita">
    <owl:inverseOf>
      <owl:ObjectProperty rdf:ID="editado_por"/>
    </owl:inverseOf>
    <rdfs:domain>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Thing"/>
          <owl:Class rdf:about="#EDITORA"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:domain>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#editado_por">
    <rdfs:domain>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Thing"/>
          <owl:Class rdf:about="#MUSICA"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:domain>
    <owl:inverseOf rdf:resource="#edita"/>
    <rdfs:comment  rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Um  album
pode ser editado por 1 ou 2 editoras</rdfs:comment>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:ID="interpretado_por">
    <owl:inverseOf>
      <owl:ObjectProperty rdf:ID="interpreta"/>
    </owl:inverseOf>
    <rdfs:domain>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Thing"/>
          <owl:Class rdf:about="#MUSICA"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:domain>
    <rdfs:comment  rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Um  album
pode ser interpretado por pelo menos 1 interprete ou banda. No maximo pode ser
interpretado por 5.</rdfs:comment>
  </owl:ObjectProperty>
  <owl:ObjectProperty rdf:about="#interpreta">
    <owl:inverseOf rdf:resource="#interpretado_por"/>
    <rdfs:domain>
```

```
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Thing"/>
          <owl:Class rdf:about="#ARTISTA"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:domain>
  </owl:ObjectProperty>
  <owl:DatatypeProperty rdf:ID="album">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Titulo do album (CD ou DVD)</rdfs:comment>
    <rdfs:domain>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Thing"/>
          <owl:Class rdf:about="#MUSICA"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:domain>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="editora">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Editora discografica</rdfs:comment>
    <rdfs:domain>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Thing"/>
          <owl:Class rdf:about="#EDITORA"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:domain>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="pais">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Thing"/>
          <owl:Class rdf:about="#MUSICA"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:domain>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="artista">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Interprete ou banda</rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Thing"/>
          <owl:Class rdf:about="#ARTISTA"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:domain>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="ano">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Ano de lancamento do album</rdfs:comment>
    <rdfs:domain>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Thing"/>
          <owl:Class rdf:about="#MUSICA"/>
        </owl:unionOf>
```

```
      </owl:Class>
    </rdfs:domain>
  </owl:DatatypeProperty>
  <owl:DatatypeProperty rdf:ID="preco">
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#float"/>
    <rdfs:domain>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Thing"/>
          <owl:Class rdf:about="#MUSICA"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:domain>
  </owl:DatatypeProperty>
  <ARTISTA rdf:ID="Evanescence">
    <artista rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Evanescence</artista>
    <interpreta>
      <CD rdf:ID="Evanescence_Fallen">
        <album rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Evanescence Fallen</album>
        <ano rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >2004</ano>
        <preco rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
        >18.8</preco>
        <pais rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >USA</pais>
        <editado_por>
          <EDITORA rdf:ID="Sony">
            <edita rdf:resource="#Evanescence_Fallen"/>
            <editora rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
            >Sony Music Entertainment</editora>
          </EDITORA>
        </editado_por>
        <interpretado_por rdf:resource="#Evanescence"/>
      </CD>
    </interpreta>
  </ARTISTA>
  <DVD rdf:ID="Live_Summer_2003">
    <ano rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >2003</ano>
    <preco rdf:datatype="http://www.w3.org/2001/XMLSchema#float"
    >16.9</preco>
    <pais rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >UK</pais>
    <album rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Live Summer 2003</album>
    <interpretado_por>
      <ARTISTA rdf:ID="Robbie_Williams">
        <artista rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Robbie Williams</artista>
        <interpreta rdf:resource="#Live_Summer_2003"/>
      </ARTISTA>
    </interpretado_por>
    <editado_por>
      <EDITORA rdf:ID="Chrysalis">
        <editora rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
        >Chrysalis Records Ltd</editora>
        <edita rdf:resource="#Live_Summer_2003"/>
      </EDITORA>
    </editado_por>
  </DVD>
</rdf:RDF>
```

De referir ainda que, usando a ferramenta Protégé, poderíamos acrescentar novas instâncias das subclasses CD ou DVD da ontologia OWL, ou seja, novos registos de albúns de música, tal como se pode verificar na Figura 137. Esses registos podem ser consultados usando a funcionalidade "Queries", tal como se pode verificar na Figura 138.

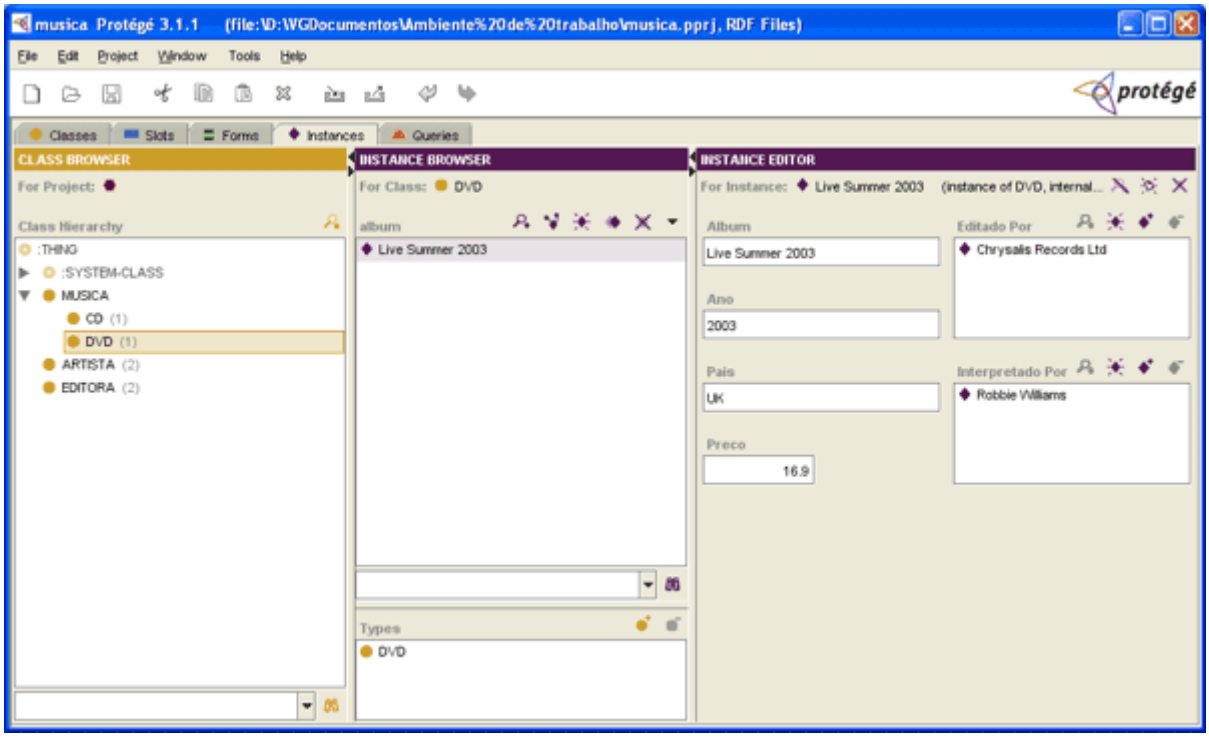


Figura 137 – Álbuns de música: adicionar instâncias no Protégé

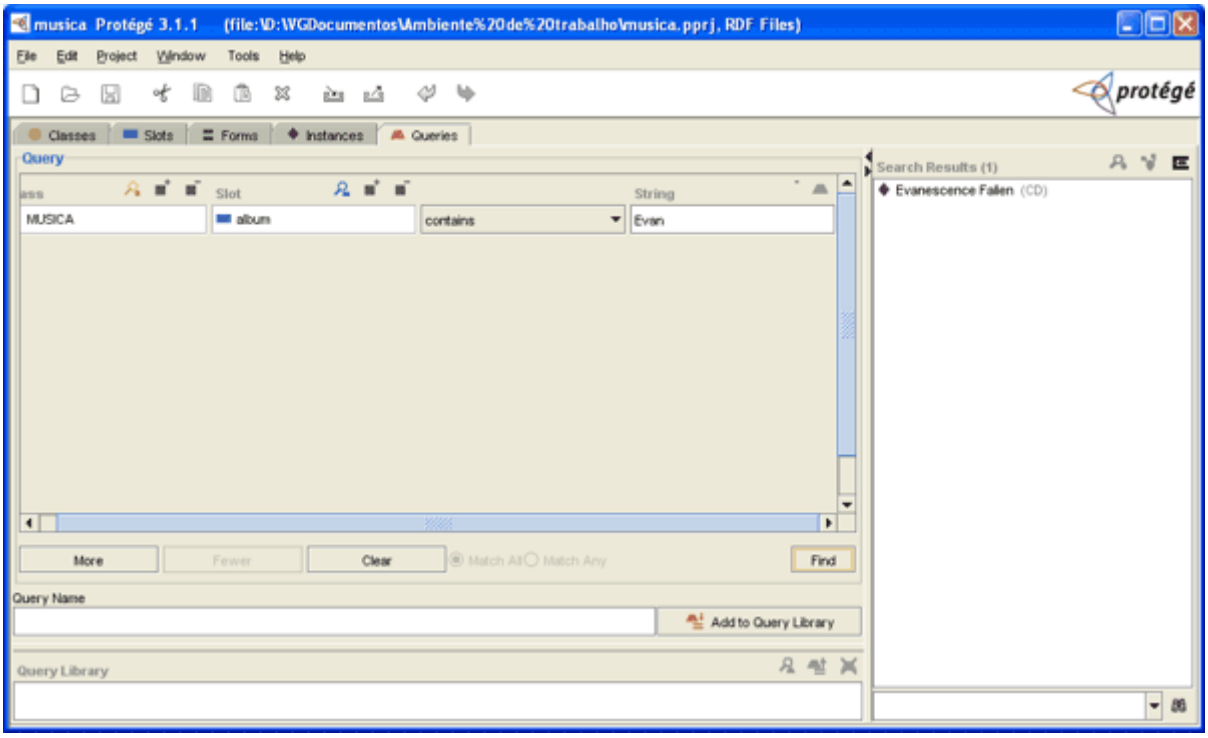


Figura 138 – Álbuns de música: realizar inferência no Protégé

Anexo D – Linguagens para as Ontologias

As linguagens ou formalismos para representação de ontologias podem ser agrupadas em linguagens baseadas na lógica de primeira ordem ou em *frames* (KIF, Ontolingua, Loom, etc.) e em linguagens baseadas em XML (SHOE, XOL, OML, RDFS, OIL, DAML+OIL, OWL, etc). Para além das linguagens OWL, RDF(S) e DAML+OIL referidas nos capítulos da tese, destacam-se ainda as seguintes linguagens para construção de ontologias:

- **F-logic** (*Frame Logic*): foi uma das primeiras linguagens a permitir a representação de conceitos, taxonomias, relações binárias, funções, instâncias, axiomas e regras, integrando frames e lógica de primeira ordem. Esta linguagem fornece a base lógica para linguagens baseadas em frames e orientadas a objectos (*object-oriented*) com vista à representação dos dados e do conhecimento;
- **NKRL** (*Narrative Knowledge Representation Language*): trata-se de uma linguagem de representação do conhecimento baseada em frames desenvolvida especificamente para descrever modelos semânticos de documentos multimédia;
- **KIF** (*Knowledge Interchange Format*) é um formato do intercâmbio do conhecimento baseado em *S-expression* ou sexp (*S stands for symbolic*) para representar dados semi-estruturados num formato textual legível para os humanos. Esta sintaxe foi criada para a lógica de primeira ordem e permite especificar ontologias;
- **Ontolingua**: trata-se de uma linguagem que combina os paradigmas das linguagens baseadas em frames e da lógica de primeira ordem. É uma linguagem com primitivas de modelagem expressivas, o que pode causar alguns problemas na construção de mecanismos de inferência;
- **LOOM**: é uma linguagem de representação do conhecimento baseada na lógica de primeira ordem e um ambiente para a construção de ontologias e aplicações inteligentes;
- **CML** (*Conceptual Modeling Language*): linguagem semi-formal com notação gráfica para descrever ontologias, proposta na metodologia CommonKADS (Schreiber et al., 1994).

- **OCML** (*Operational Conceptual Modelling Language*): linguagem utilizada em aplicações de gestão do conhecimento, desenvolvimento de ontologias, comércio electrónico e sistemas baseados em conhecimento;
- **OML** (*Ontology Markup Language*): linguagem baseada na lógica descritiva e em grafos conceptuais que permite a representação de conceitos organizados em taxonomias, relações e axiomas;
- **FOML** (*Formal Ontology Markup Language*): é uma linguagem de marcação baseada em XML que permite a ligação de documentos Web com ontologias formais, tendo por objecto final a aquisição automática de conhecimento de domínios específicos;
- **XOL** (*Ontology Exchange Language*): linguagem para a construção de ontologias através da especificação de conceitos, taxonomias e relações binárias. Foi desenvolvida com base nas linguagens OKBC-Lite (*Open Knowledge Base Connectivity*), Ontolingua e OML especificamente para o intercâmbio de ontologias no domínio da biomédica. Esta estrutura da sua sintaxe divide-se em quatro partes: cabeçalho, classes/subclasses, *slots* (propriedades) e instâncias;
- **SHOE** (*Simple HTML, Extensions*): linguagem de representação do conhecimento baseada em HTML que usa marcas ou etiquetas XML, para representar a significado dos dados ou a semântica das páginas Web. As marcações tanto podem ser usadas na construção de ontologias como na anotação das páginas Web;

Existem actualmente na Web alguns **repositórios de ontologias**, dos quais se destacam: DAML (*http://www.daml.org/ontologies*), disponibilizando cerca de três centenas de ontologias implementadas em DAML+OIL; Swoogle (*http://pear.cs.umbc.edu/swoogle*), que mais não é do que um motor de busca específico para a WS que permite pesquisar mais de quatro mil ontologias, nomeadamente esquemas RDF(S) e ontologias OWL; SchemaWeb (*http://www.schemaweb.info/default.aspx*), que permite pesquisar esquemas RDF e ontologias OWL no seu directório; Servidor Ontolingua (*http://www-ksl-svc.stanford.edu:5915*), que após o registo como utilizador identificado, permite acesso a uma livraria onde podemos encontrar algumas dezenas de ontologias escritas em Ontolingua; e a Biblioteca WebOnto que disponibiliza mais de uma centena de ontologias criadas com a ferramenta WebOnto (*http://kmi.open.ac.uk/projects/webonto*).

Anexo E – Ferramentas para as Ontologias

Para a construção de ontologias e para a anotação de páginas (que podem ou não ser baseadas em ontologias) destacam-se as seguintes ferramentas:

- ***Protégé***: é um ambiente *open source* interactivo para o projecto de ontologias que oferece uma interface gráfica e intuitiva para a edição de ontologias, para além de disponibilizar uma arquitectura para a criação de estruturas baseadas em conhecimento. Esta é uma das ferramentas gráficas de edição de ontologias e aquisição de conhecimento mais utilizadas na modelação de domínios do conhecimento, uma vez que permite a edição de ontologias em diversas linguagens para a WS de forma personalizada. Com Protégé, a sintaxe deixa de ser uma preocupação para o utilizador, podendo este canalizar a sua atenção para a representação dos conceitos e suas relações. De referir ainda que é uma ferramenta aberta baseada em Java que permite a adição de novas funcionalidades recorrendo aos *plug-ins* apropriados (*http://protege.stanford.edu*);

- ***TopBraid Composer***: é uma das mais recentes plataformas para o desenvolvimento de ontologias para a Web Semântica e para a construção de aplicações semânticas. O Composer está em conformidade com as normas W3C, nomeadamente RDF, RDFS, OWL, SWRL e SPARQL. Oferece suporte para o desenvolvimento, controlo e teste dos modelos do conhecimento (ontologias) e das suas instâncias. Fornece uma estrutura flexível e extensível através de APIs para a construção de aplicações cliente/servidor semânticas. Para além da definição de classes, subclasses e propriedades, as interfaces gráficas do Composer permitem a declaração de construtores e restrições OWL e regras SWRL (*http://www.topbraidcomposer.com*).

- ***SemanticWorks***: ferramenta comercial destinada à edição de ontologias e outros documentos em RDF, RDFS e OWL. Lida com a expressividade fornecida pelas sub-linguagens *OWL-Lite*, *OWL-DL* e *OWL-Full* e permite a sua edição em modo de texto (*http://www.altova.com*);

- ***SWOOP***: ferramenta hipermédia de visualização e edição rápida e fácil de ontologias baseadas na linguagem OWL. Esta ferramenta tem vindo a ser bem aceite em termos de usabilidade, uma vez que usa como metáfora para o seu ambiente a interface e funcionalidade de *browser* Web (*http://www.mindswap.org/2004/SWOOP*);

- ***Network Inference Construct***: é uma ferramenta gráfica de modelação baseada nas normas do W3C (em particular, OWL) que permite a construção de ontologias de uma forma fácil, eficiente e colaborativa. A facilidade da utilização desta ferramenta na criação de modelos ontológicos é comparada à utilização da ferramenta Microsoft Visio (*http://www.networkinference.com/products/construct_business.html*);
- ***OilEd***: editor de ontologias *open source* para a construção de ontologias utilizando a linguagem OIL. A versão mais recente do OilEd suporta também (sob a forma de protótipo) a leitura e escrita de ontologias OWL em RDF/XML (*http://oiled.man.ac.uk*). Não se trata de um ambiente completo para o desenvolvimento de ontologias, pelo que é considerado o "bloco de notas" dos editores de ontologias. Contudo, este editor inclui também a ferramenta FaCT reasoner para verificar a consistência e a classificação automática da ontologia (*http://www.cs.man.ac.uk/fact*);
- ***OntoEdit***: ambiente gráfico para edição, codificação, alteração, visualização e verificação de ontologias. As ontologias podem ser implementadas em XML, *F-Logic*, RDF(S) e DAML+OIL. De referir que têm vindo a ser desenvolvidos esforços para incluir raciocinadores ou mecanismos de inferência através de *plug-ins* (*http://www.ontoprise.de/products/ontoedit_en*);
- ***WebOnto***: ferramenta gráfica para a criação, edição e visualização de ontologias representadas na linguagem de modelação de conhecimento OCML. Esta ferramenta não é mais do que um applet de Java acoplado a um servidor Web personalizado que permite aos utilizadores visualizar e editar modelos de conhecimento ou ontologias via Web (*http://webonto.open.ac.uk*);
- ***KSL Ontolingua frame editor***: é uma ferramenta que suporta de forma distribuída e colaborativa a edição visualização e criação de ontologias baseadas na linguagem Ontolingua. Permite acesso a uma biblioteca de ontologias, tradutores para linguagens e a um editor de ontologias (*http://www-ksl-svc.stanford.edu:5915/FRAME-EDITOR*);
- **GFL *(Graphic F-Logic)***: é uma ferramenta gráfica que usa a popular linguagem F-Logic para a representação visual das ontologias e das consultas a base de dados. GFL combina as melhores característica das redes semânticas, dos gráficos conceptuais e do UML (*Unified Modeling Language*). Podemos usufruir também do *Ontobrowser*

que mais não é do que uma implementação limitada de Java do GFL baseado na biblioteca de TouchGraph (*http://ecoinformatics.uvm.edu/dmaps/*);

- ***KAON-TextToOnto***: ferramenta que proporciona um ambiente de aprendizagem e construção de ontologias a partir de textos. Os textos podem estar em linguagem natural ou formatos em HTML. O extractor de informações armazena os resultados em XML. Uma outra solução disponibilizada é a *KAON Ontology Framework* que é uma infra-estrutura de criação e gestão de ontologias destinada a aplicações comerciais (*http://kaon.semanticweb.org*);
- ***SMORE***: é uma ferramenta que cria marcações OWL para páginas Web, ou seja, permite a anotação de documentos HTML em OWL usando ontologias Web. Embora de forma limitada, permite também a criação ou edição de ontologias usando termos dos documentos Web (*http://www.mindswap.org/2005/SMORE*);
- ***OntoMat-Annotizer***: é uma ferramenta interativa *user-friendly* de anotação de páginas Web que auxilia o utilizador na criação e manutenção de ontologias baseadas em OWL (*http://annotation.semanticweb.org/ontomat/index.html*);
- ***Ontomarkup Annotation Tool***: tal como o próprio nome indica, é uma ferramenta baseada em ontologias para a anotação de documentos semânticos. Os utilizadores desta ferramenta têm vindo a adoptar ambientes baseados em ontologias mais eficazes no projecto e desenvolvimento de Websites, tais como: *OntoWeaver* e *KMi semantic Web* (*http://kmi.open.ac.uk/projects/akt/technologies.cfm*). Uma alternativa mais simples é a ***MnM***: uma ferramenta de anotação semi-automática que permite anotar páginas Web com os conteúdos semânticos. MnM integra um *browser* Web com um editor de ontologias e fornece APIs para ligação aos servidores de ontologias e para integração com ferramentas extracção da informação;
- ***SHOE Knowledge Annotator***: é um programa de Java que permite aos utilizadores a marcação de páginas Web com conhecimento SHOE sem terem que se preocupar com o código HTML. Esta ferramenta também está disponível *on-line* através de um applet de Java (*http://www.cs.umd.edu/projects/plus/SHOE/KnowledgeAnnotator.html*);
- ***Annotation System***: sistema baseado em três arquitecturas complementares: um cliente que pode realizar e visualizar anotações, um servidor de aplicações que gere todos os pedidos do cliente e um servidor de base de dados para armazenar as

anotações realizadas pelos utilizadores das aplicações cliente/servidor disponibilizadas (*http://www.ncb.ernet.in/groups/dake/annotate/index.shtml*);

- ***OntoAnnotate***: é uma ferramenta semi-automática de anotação que permite reunir o conhecimento dos documentos e páginas Web, com vista a criar novos documentos semânticos ricos em metadados. Neste sentido, esta ferramenta serve-se do motor de inferência subjacente (*OntoBroker*) para aceder à ontologia e à base de conhecimento. De referir que o OntoAnnotate permite a anotação em documentos HTML estáticos, documentos Word e documentos Excel;
- ***OntoBroker***: uma vez que as ontologias são o meio para estabelecer uma base conceptual concisa para comunicar o conhecimento, a ferramenta OntoBroker tem como objectivo principal a exploração de estruturas semânticas ricas com vista a suportar o acesso da máquina ao conhecimento explícito e implícito, através do processamento inteligente de ontologias. Esta ferramenta mais não é do que um motor de inferência baseado em ontologias e pode ser usado juntamente com *OntoAnnotate* (*http://www.ontoprise.de/products/ontobroker_en*).
- **Annotea**: é um projecto LEAD (*Live Early Adoption and Demonstration*) do W3C no âmbito do desenvolvimento avançado da WS (SWAD - *Semantic Web Advanced Development*). O contributo desta ferramenta incide na partilha de metadados baseados em anotações Web (comentários, notas, explicações), em registos dos favoritos (*bookmarks*) e em combinações de ambos (*http://www.w3.org/2001/Annotea*);

Uma vez que a OWL é a linguagem recomendada pelo W3C para a criação de ontologias adequadas à WS, a grande maioria das linguagens e ferramentas descritas tem vindo a fornecer suporte (por exemplo através de *plug-ins*) na migração das ontologias para a versão OWL.

Anexo F – Plataformas para o Desenvolvimento de Aplicações Baseadas em Agentes

As principais ferramentas e plataformas ou infra-estruturas que suportam o desenvolvimento de aplicações baseadas em agentes são:

- **AgentBuilder**: ambiente integrado de software que permite desenvolvimento de agentes inteligentes e de aplicações baseadas em agentes. O recurso a esta ferramenta permite que utilizadores com fracos conhecimentos sobre as linguagens de programação de agentes inteligentes possam desenvolver rápida e facilmente aplicações baseadas em agentes. Esta ferramenta pode ser aplicada na construção de sistemas multiagente para aplicações de comércio electrónico, de filtragem de e-mail, de monitorização e controlo e de outras aplicações que envolvam processos P2P (http://www.agentbuilder.com);
- **AgentSpace**: plataforma que permite o suporte, desenvolvimento e gestão de aplicações dinâmicas e distribuidas baseadas em agentes. Para tal, é disponibilizado um servidor (processo de Java onde os agentes são executados), uma API (biblioteca de classes e interfaces) e um cliente (applet de Java que permite a gestão e monitorização de agentes). O cliente e servidor são executados sobre a máquina virtual Java e utilizam funcionalidades providenciadas pelo ORB Java da Voyager (*http://berlin.inesc.pt/agentspace/index.html*);
- **AgentTcl ou D'Agents** (*Dartmouth Agents*) – inicialmente a plataforma AgentTcl apenas suportava a construção de agentes na linguagem Tcl (*Tool Command Language*). Desde que foi possível utilizar agentes móveis construídos noutras linguagens (Tcl, Java, Python e *Scheme agents*) passou a designar-se *D'Agents*. Embora seja uma aplicação para UNIX e TCP/IP, é possível instalá-la noutras plataformas e usufruir de uma ambiente de criação de aplicações baseadas em agentes móveis (*http://agent.cs.dartmouth.edu*);
- **Aglets ou ASDK** (*Aglets Software Development Kit*) - plataforma de agentes móveis e biblioteca de classes Java que facilita o desenvolvimento de aplicações baseadas em agentes. O termo Aglet corresponde a um agente de Java (*Agile* + *Applet*) capaz de se mover autónoma e espontaneamente de um *host* para outro (*aglet contexts*). O Java Aglet, originalmente criado pela IBM (*IBM Tokio Research Laboratory*), apresentou-

se como uma extensão para o movimento de código, uma vez que para além do código, o *aglet* transporta também o estado do agente, o que lhe permite migrar para os *hosts* que bem entender de forma a cumprir os seus objectivos. As primitivas de comunicação baseiam-se na invocação remota de métodos (RMI) ou na serialização de objectos, dependendo da versão do JDK (*Java Development Kit*). Actualmente, é disponibilizada pela *sourceforge.net* como uma plataforma de código aberto recomendada para a Internet *Aglets Software Development Kit* (*http://aglets.sourceforge.net*);

- **Ajanta** (*Ajanta mobile agent system*) - fruto do projecto *Mobile Agents Research Project da University of Minnesota*, este sistema destina-se ao desenvolvimento de aplicações para a Internet baseadas em agentes. Os agentes são vistos como objectos móveis activos que encapsulam código e o contexto da execução junto com dados. Esta plataforma baseia-se na linguagem Java e tira proveito de algumas das suas potencialidades, nomeadamente: a segurança, o método de invocação remoto (RMI), a serialização dos objectos e a reflexão (*http://www.cs.umn.edu/Ajanta/*);

- **ARA** (*Agents for Remote Access*) - é uma plataforma para a execução e portabilidade segura de agentes móveis em redes heterogéneas, desenvolvida pelo *AG SystemSoftware* (*Fachbereich Informatik - Technische Universität Kaiserslautern*). Comparando-a com plataformas similares, esta plataforma tem como principal objectivo tirar o máximo proveito da funcionalidade oferecida pelos agentes móveis e respectivos modelos de programação (*http://wwwagss.informatik.uni-kl.de*);

- **BEE-GENT** (*Bonding and Encapsulation Enhancement aGENT*) - disponibilizada pela *Toshiba*, é uma infra-estrutura baseada completamente em agentes (as aplicações assumem-se como agentes e todas as mensagens são transportadas por agentes) que permite desenvolver sistemas distribuídos a partir de aplicações existentes. Ao contrário de alguns dos sistemas que fazem apenas algum uso dos agentes, esta plataforma "agentifica" completamente a comunicação que ocorre entre as aplicações de software. O modelo multiagente desta plataforma baseia-se na linguagem Java e utiliza como linguagem de comunicação a linguagem ACL (*http://www.toshiba.co.jp/rdc/beegent/index.htm*);

- **Concordia** - é uma plataforma baseada na linguagem Java para a implementação e gestão de aplicações de agentes móveis, desenvolvida pela *Mitsubishi Electric Research*

Laboratories. Esta plataforma é composta por diversos componentes que combinados entre si fornecem um ambiente robusto e completo para aplicações distribuídas. Na perspectiva da Concordia, um agente é um programa em Java composto por um conjunto de objectos Java. Embora as primitivas de comunicação se baseiem na invocação de métodos locais, é possível *multicast* desde que seja usado o *AgentGroup*. Um dos principais objectivos do projecto inerente a esta plataforma é fornecer um sistema de agentes móveis para utilizadores móveis, o que permitirá alargar o espectro da rede computacional da organização (*http://www.merl.com/projects/concordia*);

- **Enago Mobile** - segundo a empresa *IKV++ Technologies AG (Innovation + Know-How + Vision Technologies AG)*, este produto assume-se como um robusto ambiente de programação e de execução (*runtime*) para aplicações que usam agentes móveis baseados na tecnologia Java. Após a descontinuidade da plataforma *Grasshoper*, disponibilizada pela mesma empresa até Outubro de 2004, podemos afirmar que este ambiente é o seu sucessor (*http://www.ikv.de/content/Produkte/enago_mobile_e.htm*);

- **JACK ou JDE** (*JACK Development Environment*) - é a terceira geração de um sistema de agentes, da responsabilidade do fornecedor de software AOS (*Agent Oriented Software*), que permite o desenvolvimento de sistemas de software distribuídos baseados em agentes inteligentes. Para tal, fornece as ferramentas necessárias à construção, execução e integração de sistemas multiagente usando a linguagem Java com algumas extensões inerentes a conceitos orientados a agentes. Esta plataforma disponibiliza também um mecanismo de inferência genérico, baseado no modelo BDI - *Belief, Desire and Intention Model* (crença, desejo, intenção), para que os agentes inteligentes possam raciocinar, isto é, planear as tarefas necessárias para atingir os objectivos. Em suma, JACK fornece a arquitectura e as capacidades para o desenvolvimento e execução de agentes de software em aplicações distribuídas (*http://www.agent-software.com*);

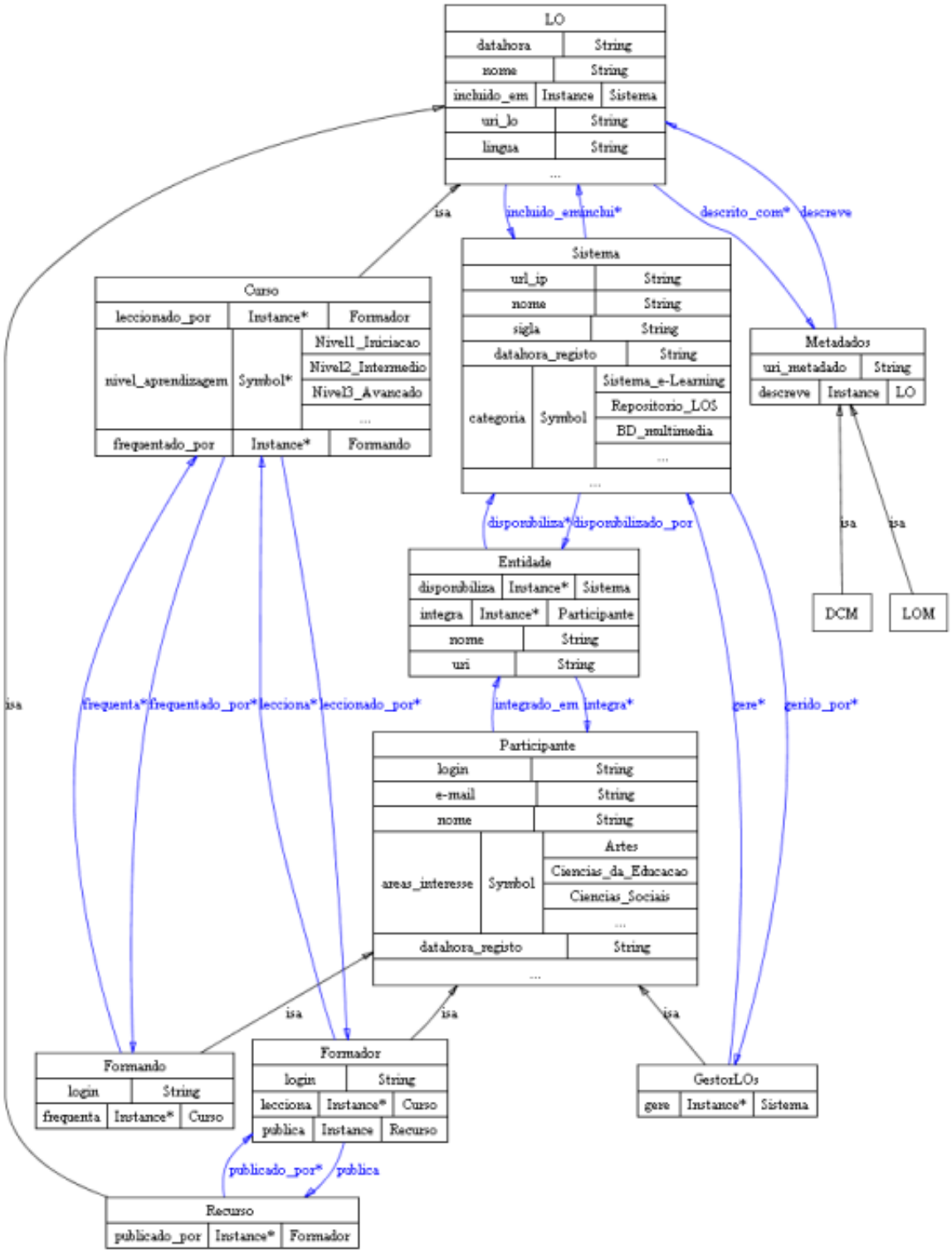
- **JADE** (*Java Agent DEvelopment Framework*) - é uma plataforma de software disponibilizada pela TILab (*Telecom Italia Lab*) para desenvolver aplicações baseadas em agentes de acordo com as especificações FIPA para sistemas multiagente inteligentes e interoperáveis. Assume-se como o *middleware* ou plataforma que suporta o desenvolvimento de agentes inteligentes para aplicações P2P (*http://jade.tilab.com*);

- **Mole** (*Mobile Agent System Mole*) - a investigação levada a cabo no *Institute of Parallel and Distributed Systems of the University of Stuttgart* resultou num sistema de agentes móveis que teve como linguagem de implementação a linguagem orientada a objectos Java, tanto no desenvolvimento do sistema, como na construção dos agentes que o compõem (*http://mole.informatik.uni-stuttgart.de*);
- **Plangent** - fruto de um dos vários projectos da *Toshiba* no âmbito da tecnologia para os agentes, este ambiente permite o desenvolvimento de agentes inteligentes que operam em sistemas baseados na linguagem Java. As principais características dos agentes *Plangent* são: autonomia, mobilidade e adaptabilidade. A estas capacidades adiciona-se a possibilidade de recorrerem a uma base de conhecimento existente em cada nó da rede, para além da sua própria base de conhecimento, a fim de executarem as suas tarefas (*http://www.toshiba.co.jp/rdc/plangent/index.htm*);
- **TACOMA** (*Tromsø And COrnell Moving Agents*) - é uma plataforma de sistema operativo para suportar a actividade dos agentes. Resultou dos esforços do *Institutt for Informatikk - University of Tromsø* e baseia-se em Unix e TCP/IP para mover os agentes que podem ser escritos em diversas linguagens (C, Tcl/Tk, Perl, Python e *Schema*). Uma vez que não suporta o desenvolvimento de agentes escritos em Java, este sistema recorre ao TCP para transportar o código e o estado dos agentes de um *host* para o outro. Assim, os agentes podem encontrar-se num *host* e trocar dados usando a primitiva *meet* (*http://www.tacoma.cs.uit.no*);
- **Voyager** - desenvolvida pela *ObjectSpace* para suportar o desenvolvimento de aplicações distribuídas é, actualmente, disponibilizada pela *Recursion Software*. Esta plataforma de desenvolvimento de aplicações baseadas em agentes móveis é considerada um ORB de Java que usa o modelo de objectos da linguagem Java, possibilitando a migração dos objectos como agentes na rede. A filosofia desta plataforma encara um agente simplesmente como um tipo especial de objecto que pode mover-se independentemente, pode continuar a executar o seu plano enquanto se move e comportar-se tal como qualquer outro objecto. Esta plataforma fornece diversos mecanismos de comunicação, tais como: suporte ORB, RMI, CORBA (*Common Object Request Broker Architecture*) e DCOM (*Distributed Component Object Model*); invocações síncronas, *one-way*, *future-relay* e *multicast* de eventos e mensagens. Actualmente, a Voyager assume-se como a plataforma de

desenvolvimento Java mais completa no suporte à tecnologia de agentes móveis inteligentes (*http://www.recursionsw.com/voyager.htm*).

Esta listagem apresentou de forma sucinta as ferramentas e plataformas mais mencionadas na bibliografia. Existem muitas outras ferramentas, plataformas, infra-estruturas e iniciativas no âmbito do desenvolvimento de sistemas multiagente, nomeadamente de agentes inteligentes e de agentes móveis.

Anexo G – Ontologia para o Sistema WSE



Anexo H – Ontologia OWL para o Sistema WSE

```
<?xml version="1.0" ?>
<rdf:RDF
```

Namespaces usados:

```
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:xsd="http://www.w3.org/2001/XMLSchema#"
    xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
    xmlns:owl="http://www.w3.org/2002/07/owl#"
    xmlns="http://www.vgportal.ipb.pt/wse.owl#"
    xml:base="http://www. vgportal.ipb.pt/wse.owl">
```

Resumo da definição de classes e respectivas propriedades (incluindo as relações):

```
<owl:Ontology rdf:about="" />
<owl:Class rdf:ID="Formador">
<owl:Class rdf:ID="Entidade">
<owl:Class rdf:ID="DCM">
<owl:Class rdf:ID="Formando">
<owl:Class rdf:ID="Recurso">
<owl:Class rdf:ID="LOM">
<owl:Class rdf:ID="Curso">
<owl:Class rdf:ID="Sistema">
<owl:Class rdf:about="#LO">
<owl:Class rdf:about="#Metadados">
<owl:Class rdf:about="#Participante">
<owl:Class rdf:ID="GestorLOs">
<owl:ObjectProperty rdf:ID="incluido_em">
<owl:ObjectProperty rdf:ID="frequentado_por">
<owl:ObjectProperty rdf:ID="disponibilizado_por">
<owl:ObjectProperty rdf:ID="descrito_com">
<owl:ObjectProperty rdf:ID="leccionado_por">
<owl:ObjectProperty rdf:ID="integra">
<owl:ObjectProperty rdf:about="#frequenta">
<owl:ObjectProperty rdf:about="#lecciona">
<owl:ObjectProperty rdf:ID="gere">
<owl:ObjectProperty rdf:ID="publicado_por">
<owl:ObjectProperty rdf:about="#descreve">
<owl:ObjectProperty rdf:about="#gerido_por">
<owl:ObjectProperty rdf:about="#publica">
<owl:ObjectProperty rdf:about="#integrado_em">
<owl:ObjectProperty rdf:about="#disponibiliza">
<owl:ObjectProperty rdf:about="#inclui">
<owl:DatatypeProperty rdf:ID="uri">
<owl:DatatypeProperty rdf:ID="nome">
<owl:DatatypeProperty rdf:ID="descricao">
<owl:DatatypeProperty rdf:about="#e-mail">
<owl:DatatypeProperty rdf:about="#uri_foaf">
<owl:DatatypeProperty rdf:ID="areas_interesse">
<owl:DatatypeProperty rdf:ID="categoria">
<owl:DatatypeProperty rdf:ID="nivel_aprendizagem">
<owl:DatatypeProperty rdf:ID="datahora_registo">
<owl:DatatypeProperty rdf:ID="uri_metadado">
<owl:DatatypeProperty rdf:ID="url_ip">
<owl:DatatypeProperty rdf:about="#lingua">
<owl:DatatypeProperty rdf:about="#login">
<owl:DatatypeProperty rdf:ID="uri_lo">
<owl:DatatypeProperty rdf:ID="tipo_los">
<owl:DatatypeProperty rdf:ID="sigla">
<owl:DatatypeProperty rdf:ID="datahora">
```

Instâncias:

```
<LOM rdf:ID="OntoWSE_Instance_uri">
...

</rdf:RDF>
```

Classe Sistema (a declaração das classes participante, entidade, metadados e LO é análoga):

```
  <owl:Class rdf:ID="Sistema">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Servidor externo ou
remoto de um sistema de e-Learning, repositorio de LOs ou de um outro sistema que
disponibilize objectos de aprendizagem.
    </rdfs:comment>
  </owl:Class>
```

Subclasse Curso (a declaração das subclasses recurso, formador, formando, gestorLOs, DCM e LOM é análoga):

```
  <owl:Class rdf:ID="Curso">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Disciplina ou curso
de formacao, normalmente um LO de tamanho grande e com elevada granularidade</rdfs:comment>
    <rdfs:subClassOf>
      <owl:Class rdf:about="#LO"/>
    </rdfs:subClassOf>
  </owl:Class>
```

Declaração da propriedade “nome” aplicada a várias classes:

```
  <owl:DatatypeProperty rdf:ID="nome">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Nome ou designacao consoante a classe a que se aplica a propriedade</rdfs:comment>
    <rdfs:range rdf:resource="http://www.w3.org/2001/XMLSchema#string"/>
    <rdfs:domain>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Thing"/>
          <owl:Class rdf:about="#Participante"/>
          <owl:Class rdf:about="#Sistema"/>
          <owl:Class rdf:about="#LO"/>
          <owl:Class rdf:about="#Entidade"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:domain>
  </owl:DatatypeProperty>
```

Declaração da relação “lecciona” e “leccionado_por” aplicada às classes Formador e Curso:

```
  <owl:ObjectProperty rdf:about="#lecciona">
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string"
    >Um professor pode leccionar um ou mais cursos. Util para definir que professores podem
anotar semanticamente esses cursos ou disciplinas</rdfs:comment>
    <rdfs:domain>
      <owl:Class>
        <owl:unionOf rdf:parseType="Collection">
          <rdf:Description rdf:about="http://www.w3.org/2002/07/owl#Thing"/>
          <owl:Class rdf:about="#Formador"/>
        </owl:unionOf>
      </owl:Class>
    </rdfs:domain>
    <owl:inverseOf rdf:resource="#leccionado_por"/>
  </owl:ObjectProperty>
```

Declaração de uma restrição simples para a propriedade “login” da classe Formador:

```
<owl:Class rdf:ID="Formador">
    <rdfs:subClassOf>
      <owl:Restriction>
        <owl:hasValue rdf:datatype="http://www.w3.org/2001/XMLSchema#string">p00000
         </owl:hasValue>
        <owl:onProperty>
          <owl:DatatypeProperty rdf:ID="login"/>
        </owl:onProperty>
      </owl:Restriction>
    </rdfs:subClassOf>
    <rdfs:comment rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
      Formador ou Professor que usa o Sistema WSE</rdfs:comment>
    <rdfs:subClassOf>
      <owl:Class rdf:ID="Participante"/>
    </rdfs:subClassOf>
  </owl:Class>
```

Mecanismo para declarar que duas classes (ou duas instâncias) são idênticas e que duas classes são diferentes. Este mecanismo é útil para alinhar a ontologia para o sistema WSE com a ontologia para os sistemas de e-Learning:

```
  <owl:Class rdf:ID="Aluno"/>
  <owl:Class rdf:ID="Professor"/>
…
  <owl:Class rdf:ID="Formando">
    <owl:sameAs>
      <Aluno rdf:ID="Aluno_XYZ">
        <owl:sameAs rdf:resource="#Formando"/>
      </Aluno>
    </owl:sameAs>
    <owl:differentFrom>
      <Professor rdf:ID="Professor_ABC">
        <owl:differentFrom rdf:resource="#Formando"/>
      </Professor>
    </owl:differentFrom>
  </owl:Class>
```

Exemplo parcial de uma instância de um curso ou disciplina descrito por metadados LOM:

```
<LOM rdf:ID="Metadados_Ergonomia">
 <descreve>
  <Curso rdf:ID="Ergonomia">
  <datahora rdf:datatype="http://www.w3.org/2001/XMLSchema#string">2006/09/16</datahora>
  <tipo_los rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Disciplina
  </tipo_los>
  <leccionado_por>
  …
  </leccionado_por>
  <uri_lo rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
                        https://www.easy-learning.ipb.pt/course/view.php?id=10
  </uri_lo>
  <lingua rdf:datatype="http://www.w3.org/2001/XMLSchema#string">pt_utf8</lingua>
  <nome rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Ergonomia</nome>
  <descricao rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Disciplina leccionada ao
Curso de Especialização Tecnológica - Desenvolvimento de Produtos Multimédia com carga horária
de 40 horas. Abordagem Ergonómica de Sistemas. Esta unidade temática tem como principal
finalidade explorar os principais conceitos e directrizes no âmbito Interacção Humano-
Computador (IHC), nomeadamente ao nível da optimização, confiabilidade, usabilidade e
acessibilidade do sistema.
  </descricao>
  <descrito_com rdf:resource="#Metadado_programa_uri" />
  <frequentado_por rdf:resource="#Alunos" />
  <nivel_aprendizagem rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
       Curso_Completo
  </nivel_aprendizagem>
  <tipo_los rdf:datatype="http://www.w3.org/2001/XMLSchema#string">Pacote_SCORM
  </tipo_los>
  <incluido_em rdf:resource="#Ergonomia" />
  <tipo_los rdf:datatype="http://www.w3.org/2001/XMLSchema#string">LO_avaliacao
  </tipo_los>
  <incluido_em rdf:resource="#Ergonomia" />
  </Curso>
  </descreve>
  <uri_metadado rdf:datatype="http://www.w3.org/2001/XMLSchema#string">
       https://www.easy-learning.ipb.pt/mod/metadatalom/view.php?id=1725
  </uri_metadado>
</LOM>
```

Anexo I – Geração (semi)automática de metadados DCM

Simple DC – Elemento usado nos metadados *Simple Dublin Core*
Auto S-Q – Elemento gerado automaticamente no *Simple Dublin Core* e/ou no *Qualified Dublin Core*
✍ – Elemento gerado automaticamente em que pontualmente há necessidade de proceder a alterações

Elementos DCMES *15 core elements*	**Sub-elementos DCM** *7 new elements + 33 Refinements*	**Valor predefinido ou gerado**	**Simple DC?**	**Auto? S-Q**
Title		Nome do recurso ou actividade (BD Moodle)	✓	✓
	alternative	Nome fornecido pelo utilizador quando inicia a anotação + nome do módulo (classe do LO)		✓✍
Creator		Formador que tem sessão activa para editar metadados e respectivo departamento e instituição (BD Moodle)	✓	✓
Subject		Extracção de palavras do sumário, texto completo ou do próprio ficheiro textual	✓	✓✍
Description		Resumo do LO (+ introdução caso exista na BD Moodle)	✓	✓✍
	abstract	Resumo do LO (BD Moodle)		✓
	tableOfContents			
Publisher		Formador que tem sessão activa para editar metadados e respectivo departamento e instituição (BD Moodle)	✓	✓
Contributor		Formador que tem sessão activa para editar metadados e respectivo departamento e instituição (BD Moodle)	✓	✓
Date		Data LO (BD Moodle)	✓	✓
	created	Data LO (BD Moodle)		✓✍
	valid	Data LO (BD Moodle)		✓✍
	available	Data LO (BD Moodle)		✓✍
	issued	Data LO (BD Moodle)		✓✍
	modified	Data LO (BD Moodle)		✓
	dateAccepted	Data LO (BD Moodle)		✓✍
	dateCopyrighted	Data do sistema actual		✓
	dateSubmitted	Data LO (BD Moodle)		✓✍
Type		Valor seleccionado automaticamente de acordo com o tipo de LO	✓	✓
Format		Extracção do formato MIME ou *content-type* do ficheiro do LO	✓	✓
	extent	Extracção do tamanho ou *content-length* do ficheiro do LO (função PHP get_headers)		✓
	medium	Extracção do nome do servidor e outras configurações do sistema		✓

(Continua na página seguinte)

(Continuação da página anterior)

Elementos DCMES *15 core elements*	**Sub-elementos DCM** *7 new elements + 33 Refinements*	**Valor predefinido ou gerado**	**Simple DC?**	**Auto S-Q**
Identifier		URL do LO	✓	✓
	Bibliographic Citation			
Source			✓	
Language		Linguagem (BD Moodle)	✓	✓
Relation		Seleccionadas automaticamente 3 tipos de relação	✓	✓✍
	isVersionOf	Geradas automaticamente 3 URLs inerentes às relações: isPartOf - curso; isReferencedBy - metadado; requires - moodle easy-learning.		
	hasVersion			
	isReplacedBy			
	replaces			
	isRequiredBy			
	requires			✓
	isPartOf			✓
	hasPart			
	isReferencedBy			✓
	references			
	isFormatOf			
	hasFormat			
	conformsTo			
Coverage			✓	
	spatial			
	temporal			
Rights		String predefinida (Informações da BD e configurações do Moodle)	✓	✓
	accessRights	String predefinida (Informações da BD Moodle)		
	license			
	Audience	*Token* inferido de acordo com a categoria do curso ou disciplina do LO. Se elemento vazio, então associa: LO oculto = *teacher* LO visível = *learner*		✓
	mediator			
	education Level			
	Provenance			
	Rights Holder	Dados do formador (BD Moodle)		✓✍
	Instructional Method			
	Accrual Method			
	Accrual Periodicity			
	Accrual Policy			

Anexo J – Geração (semi)automática de metadados LOM

Simple LOM– Elemento usado no *profile* de metadados *Simple LOM* (alguns elementos da especificação)
Auto S - C – Elemento gerado automaticamente no *Simple LOM* e/ou no *Complete LOM* (todos os elementos)
✍ – Elemento gerado automaticamente em que pontualmente há necessidade de proceder a alterações

Elementos LOM	Sub-elementos LOM	Min-Máx	Valor predefinido ou gerado	Simple LOM?	Auto? S - C
General		**0 – 1**			
Identifier		0 – n			
	Catalog	0 – 1	URI		✓
	Entry	0 – 1	URL do recurso ou da actividade (URL do LO)	✓	✓
Title	String	0 – 1	Nome LO (BD Moodle)	✓	✓
Language	Unspecified	0 – n	Linguagem (BD Moodle)	✓	✓
Description	String	0 – n	Resumo LO BD Moodle)	✓	✓
Keyword	String	0 – n	Extracção estatística	✓	✓✍
Coverage	String	0 – n		✓	
Structure	Source Value	0 – 1	*Token* inferido de acordo com tipo de LO		✓
Aggregation Level	Source Value	0 – 1	*Token* inferido de acordo com tipo de LO e estrutura		✓
Life Cycle		**0 – 1**			
Version	String	0 – 1	v1.0, excepto para pacotes scorm ou IMS-CP em que é apresentada a versão do mesmo	✓	✓✍
Status	Source Value	0 – 1	LO oculto = *unavailable* LO visível = *final*	✓	✓
Contribute		0 – n			
	Role	0 – 1	Formador da sessão actual = *creator*		✓
	Entity	0 – n	Dados do formador no formato *vCard* (BD Moodle)	✓	✓
	Date	0 – 1	Data LO (BD Moodle)		✓

(Continua na página seguinte)

(Continuação da página anterior)

Elemento LOM	Sub-elementos LOM	Min-Máx	Valor predefinido ou gerado	Simple LOM?	Auto? S - C
Meta-Metadata		**0 – 1**			
Identifier		0 – n			
	Catalog	0 – 1	URI		✓
	Entry	0 – 1	URL do metadado do LO	✓	✓
Contribute		0 – n			
	Role	0 – 1	Formador da sessão actual = *creator*		✓
	Entity	0 – n	Dados do formador no formato *vCard* (BD Moodle)		✓
	Date	0 – 1	Data do sistema actual		✓
Metadata Schema	None	0 – n	*Complete LOM* = v1.0 *Simple LOM* = v0.5	✓	✓
Language	None	0 – 1	Linguagem (BD Moodle)	✓	✓
Technical		**0 – 1**			
Format	None	0 – n	Extracção do formato MIME ou *content-type* do ficheiro do LO	✓	✓
Size	None	0 – 1	Extracção do tamanho ou *content-length* do ficheiro do LO (função PHP get_headers)		✓
Location	None	0 – n	URL do LO	✓	✓
Requirement	OrComposite	0 – n			
OrComposite		0 – n			
	Type	0 – 1	*Browser* (Informações do sistema do utilizador)		✓
	Name	0 – 1	Nome do *browser* usado pelo formador (Informações do sistema do utilizador)		✓
	Minimum Version	0 – 1	Versão do Moodle aquando da criação do LO		✓

(Continua na página seguinte)

(Continuação da página anterior)

Elemento LOM	Sub-elementos LOM	Min-Máx	Valor predefinido ou gerado	Simple LOM?	Auto? S - C
	Maximum Version	0 – 1	Versão do Moodle actual (aquando da edição dos metadados do LO)		✓
Instalation Remarks	None	0 – 1	Comentário predefinido	✓	✓
Other Plataform Requirements	None	0 – 1	Versão do módulo		✓
Duration	None	0 – 1			
Educational		**0 – n**			
Interactivity Type	Source Value	0 – 1	Normalmente, um recurso = *expositive* e uma actividade = *active*. Módulos como o *book* ou pacotes SCORM são considerados *mixed*.	✓	✓✍
Learning Resource Type	Source Value	0 – n	Tipos de módulos de LOs e sua interligação com os *tokens* sugeridos	✓	✓
Interactivity Level	Source Value	0 – 1	*Token* inferido de acordo com o tipo de LO	✓	✓✍
Semantic Density	Source Value	0 – 1	*Token* inferido de acordo com tipo de LO. Exemplo: Recurso página de texto ou sumário = densidade baixa e *book* = densidade alta (mais do que um elemento para expressar o conteúdo)	✓	✓
Intended End User Role	Source Value	0 – n	LO oculto = *teacher* LO visível = *learner*	✓	✓
Context	Source Value	0 – n	*Token* inferido de acordo com a categoria do curso ou disciplina do LO	✓	✓
Typical Age Range	String	0 – n	*Token* associado de acordo com o *Context*	✓	✓✍
Difficulty	Source Value	0 – 1		✓	
Typical Learning Time	Duration Description	0 – 1		✓	
Description	String	0 – n		✓	

(Continua na página seguinte)

(Continuação da página anterior)

Elemento LOM	Sub-elementos LOM	Min-Máx	Valor predefinido ou gerado	Simple LOM?	Auto? S - C
Language	None	0 – n	Linguagem (BD Moodle)	✓	✓
Rights		**0 – 1**			
Cost	Source Value	0 – 1	Custo (BD Moodle)	✓	✓
Copyright And Other Restrictions	Source Value	0 – 1	Yes é o valor predefinido	✓	✓
Description	String	0 – 1	String predefinida (Informações da configuração do sistema e BD Moodle)	✓	✓
Relation		**0 – n**			
Kind	Source Value	0 – 1	Geradas 3 relações automaticamente (isPartOf – curso; isReferencedBy – metadado; requires – moodle easy-learning)	✓	✓✍
Resource	Identifier Description	0 – 1			
Identifier		0 – n			
	Catalog	0 – 1	URI		✓
	Entry	0 – 1	Gerados 3 URL de acordo com as 3 relações	✓	✓✍
Description	String	0 – n	String predefinida (Informações da BD Moodle) de acordo com os 3 tipos de relações	✓	✓✍
Annotation		**0 – n**			
Entity	None	0 – 1	Dados do formador no formato *vCard* (BD Moodle)		✓
Date	DateTime Descrition	0 – 1	Data do sistema actual		✓
Description	String	0 – 1			

(Continua na página seguinte)

(Continuação da página anterior)

Elemento LOM	Sub-elementos LOM	Min-Máx	Valor predefinido ou gerado	Simple LOM?	Auto? S - C
Classification		**0 – n**			
Purpose	None	0 – 1		✓	
Taxon Path	Source Taxon	0 – n			
Source	String	0 – 1		✓	
Taxon		0 – n			
	Id	0 – 1			
	Entry	0 – 1			
Description	String	0 – 1		✓	
Keyword	String	0 – n			
Total de elementos				35-58	28-46

Anexo K – Inquéritos de Avaliação

Inquérito à utilização do *Sistema Moodle easy-Learning*
Professores

O *Sistema Moodle easy-Learning* atravessou um período experimental de utilização. Com o objectivo de avaliar o grau de satisfação dos utilizadores com o sistema e, mais concretamente com os módulos de Programas, Sumários e Metadados (informação sobre os objectos de aprendizagem, recursos ou actividades), pedimos-lhe encarecidamente que nos dê a sua opinião sobre as funcionalidades desta plataforma de b-Learning.

Parte I

Numa escala de 1 a 5 (1 = nada importante; 2 = pouco importante, 3 = importante; 4 = muito importante; 5 = fundamental), responda por favor aos pedidos/questões abaixo. Deixe vazio, caso não saiba responder, não tenha opinião ou não se aplique à utilização que fez do sistema até à data.

1. Classifique cada um destes factores no que diz respeito ao contributo que fornecem para a sua satisfação com o *Sistema Moodle easy-Learning*?

A usabilidade do sistema de e-Learning (amigável e fácil de usar) As funcionalidades do sistema de e-Learning A aparência e a organização dos objectos de aprendizagem A possibilidade de criar recursos ou actividades on-line A possibilidade de disponibilizar actividades de colaboração entre alunos O módulo "Programa da Disciplina"
O módulo "Sumários de Aulas" A existência de metadados para descrever automaticamente os objectos de aprendizagem A existência de um bloco de pesquisa adicional para procurar conteúdos: Procurar LO-metadados A disponibilidade e competência do Administrador do Sistema no esclarecimento de dúvidas
A ausência de problemas técnicos e de dificuldades de utilização da plataforma Outros:

2. Quando comparado com o ensino presencial, o *Sistema Moodle easy-Learning* influenciou positivamente os alunos no que diz respeito ao seu nível de:

Motivação	
Aprendizagem	
Aproveitamento	
Interacção com conteúdos (e materiais de aprendizagem publicados)	
Interacção com os outros alunos	
Outra:	

3. Já utilizou os módulos de metadados (LOM e/ou DCMES) para descrever os objectos de aprendizagem de uma disciplina?

Sim ☐ Não ☐

Se não, active o modo de edição na página principal de uma das suas disciplinas e adicione metadados aos recursos ou actividades existentes. De seguida, continue o preenchimento do inquérito.

Se sim, classifique a importância destes factores no que diz respeito ao contributo que os módulos de metadados implicam no *Sistema Moodle easy-Learning*?

A visibilidade dos metadados associados aos objectos de aprendizagem de uma disciplina.	
A visibilidade dos metadados de um objecto de aprendizagem mesmo que não tenha autorização para ver esse objecto de aprendizagem (ficar a saber do que trata o objecto de aprendizagem embora não o possa visualizar ou utilizar)	
A eficácia dos módulos na descrição dos objectos de aprendizagem	
A eficácia dos módulos ao permitir a pesquisa de objectos de aprendizagem	
Outro:	

4. Como classifica a importância de cada um dos seguintes factores no que diz respeito à utilidade dos metadados para os intervenientes do processo educativo?

Úteis para complementar ou esclarecer dúvidas sobre os objectos de aprendizagem	
Úteis para cruzar informação sobre os temas, conteúdos e conceitos leccionados nas várias disciplinas de um curso (úteis para definir estratégias de interdisciplinaridade)	
Úteis para saber que conteúdos são leccionados em disciplinas da mesma turma (perspectiva aluno)	
Úteis para suportar a tomada de decisão de potenciais alunos (obter informação sobre as temáticas leccionadas nas disciplinas para eventualmente adicionar essa disciplina ao seu plano de estudos)	
Úteis para perceber a que se referem os objectos de aprendizagem devolvidos pelo bloco de pesquisa	
Outra:	

5. Já utilizou o bloco de pesquisa de recursos/actividades (Procurar LO-metadados)?

Sim ☐ Não ☐

Se não, localize esse bloco na página principal do *easy-Learning* e tente procurar informação relevante acerca de recursos que descreveu, quer através da pesquisa básica, quer através da pesquisa avançada.
Se sim, avalie a importância da utilização que pode fazer desse bloco:

Pesquisa de objectos de aprendizagem nas minhas disciplinas	
Pesquisa de objectos de aprendizagem em todas as disciplinas do *easy-Learning*	
Resultados devolvidos (os objectos devolvidos correspondem à pesquisa solicitada)	
Outra:	

Parte II

1. Antes de usar o easy-Learning, alguma vez utilizou um sistema de e-Learning?

Não ☐ Sim como formando ☐ Sim como formador ☐

Qual? _____

2. Conhecia alguma norma ou esquema de metadados?

Não ☐ Sim (LOM) ☐ Sim (DCMES) ☐ Sim (Outra: _____) ☐

3. Tinha alguma experiência de catalogação e indexação de recursos de informação?

Nenhuma ☐ Pouca ☐ Alguma ☐ Muita ☐ Muita com formação na área ☐

4. Há quanto tempo usa metadados para classificar recursos digitais?

Nunca ☐ Só agora com o *easy-Learning* ☐ há menos de 1 ano ☐ há mais de 1 ano ☐
Se aplicável, indique o sistema que usou: _____

5. Usou o módulo LOM e/ou DCMES nas suas disciplinas antes de iniciar este inquérito?

Sim ☐ Não ☐

Se sim, indique:
5.1.Se tinha que alterar regularmente alguns dos elementos gerados automaticamente e, se sim, quais.

_____
_____
_____

5.2. Se para além dos elementos gerados automaticamente, existe algum que considere que poderia ser automaticamente preenchido.

_____
_____
_____

5.3. Dos seguintes sistemas de classificação, indique (com um **x**) quais usou na adição de metadados:

ACM (*Association for computing Machinery – Computing Classification System*)	
ARIADNE	
LCSH (*Library of Congress Subject Headings*)	
MESH (*Medical Subject Headings*)	
NLM (*National Library of Medicine Classification*)	
DDC (*Dewey Decimal Classification*)	
LCC (*Library of Congress Classification*)	
UDC (*Universal Decimal Classification*)	
Outro:	

5.4. Que tipo de funcionalidade preferia para a criação ou adição de metadados?

Módulo de geração de metadados totalmente automático, sem qualquer intervenção humana	
Módulo com geração (semi)automática de metadados, permitindo edição por parte do autor	
Módulo para a criação manual de metadados com validação automática	
Módulo para a criação manual de metadados sem validação automática	
Outro:	

5.5. Dos elementos de metadados das norma LOM e DCMES, quais os elementos que considera mais relevantes para a pesquisa de informação?

_____
_____
_____
_____
_____

5.6. E quais acha que devem ser de preenchimento automático, semi-automático e manual?

_____
_____
_____
_____
_____

MUITO OBRIGADO PELO SEU TEMPO ☺!

Inquérito à utilização do *Sistema Moodle easy-Learning*
Alunos

O *Sistema Moodle easy-Learning* atravessou um período experimental de utilização. Com o objectivo de avaliar o grau de satisfação dos utilizadores com o sistema e, mais concretamente com os módulos de Programas, Sumários e Metadados (informação sobre os objectos de aprendizagem, recursos ou actividades), agradecemos encarecidamente a sua opinião sobre as questões abaixo.

Numa escala de 0 a 5, em que 0 significa não sei responder ou sem opinião, 1 é muito pouco importante e 5 é muito importante:

1. Como classifica cada um destes factores no que diz respeito ao contributo que fornecem para a sua satisfação com o *Sistema Moodle easy-Learning*?

A usabilidade do sistema de e-Learning (amigável e fácil de usar) As funcionalidades do sistema de e-Learning A aparência e a organização dos objectos de aprendizagem A interacção com os objectos de aprendizagem disponibilizados (outros para além de PDFs) As actividades de interacção e colaboração com outros alunos e com o docente O dinamismo e apoio do docente
O módulo adicional de programa da disciplina O módulo adicional de sumários de aulas A existência de metadados para descrever e complementar os objectos de aprendizagem A existência de um bloco de pesquisa adicional para procurar conteúdos: Procurar LO-metadados
A disponibilidade e competência do Administrador do Sistema no esclarecimento de dúvidas A ausência de problemas técnicos e de dificuldades de utilização da plataforma Outro:

2. Quando comparado com o ensino presencial, como classifica a influência do *Sistema Moodle easy-Learning* no que diz respeito ao seu nível de:

Motivação	
Aprendizagem	
Aproveitamento	
Acesso aos conteúdos educativos da disciplina	
Outra:	

3. Como classifica cada um destes factores no que diz respeito ao contributo que os módulos de metadados implicam no *Sistema Moodle easy-Learning*?

A visibilidade dos metadados juntamente com os objectos de aprendizagem de uma disciplina.	
A visibilidade dos metadados de um objecto de aprendizagem mesmo que não tenha autorização para ver esse objecto de aprendizagem (ficar a saber do que trata o objecto de aprendizagem embora não o possa visualizar ou utilizar)	
A eficácia dos módulos na descrição dos objectos de aprendizagem	
A eficácia dos módulos ao permitir a pesquisa de objectos de aprendizagem	
Outro:	

4. Aquando da visualização dos recursos/actividades das suas disciplinas ou cursos de formação estavam disponíveis alguns metadados?

Sim ☐ Não ☐

Se sim, meça a utilidade dos mesmos:

Úteis para complementar ou esclarecer dúvidas sobre os objectos de aprendizagem	
Úteis para cruzar informação sobre os temas, conteúdos e conceitos leccionados nas várias disciplinas que frequento (interdisciplinaridade)	
Úteis para saber que conteúdos são leccionados em disciplinas nas quais não estou inscrito	
Úteis para suportar a tomada de decisão de potenciais alunos (obter informação sobre as temáticas leccionadas nas disciplinas para eventualmente adicionar essa disciplina ao meu plano de estudos)	
Úteis para perceber a que se referem os objectos de aprendizagem devolvidos pelo bloco de pesquisa	
Outra:	

5. Já utilizou o bloco de pesquisa de recursos/actividades (Procurar LO-metadados)?

Sim ☐ Não ☐

Se não, localize esse bloco na página principal do easy-Learning e tente procurar informação relevante para si, quer através da pesquisa básica, quer através da pesquisa avançada.
Se sim, meça a utilização que faz do mesmo:

Pesquisa de objectos de aprendizagem nas minhas disciplinas	
Pesquisa de objectos de aprendizagem em todas as disciplinas do easy-Learning	
Resultados devolvidos (os objectos devolvidos normalmente correspondem à pesquisa solicitada)	
Outra:	

6. Antes de usar o easy-Learning, alguma vez utilizou um sistema de e-Learning como aluno ou formando?

Sim ☐ Não ☐

Qual? _____

MUITO OBRIGADO PELO SEU TEMPO ☺!