LOD: uma abordagem para desenvolvimento de objetos de aprendizagem multimídias e interativos

Marco Aurélio Graciotto Silva

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de depósito: 05/03/2012

Assinatura: _____

LOD: uma abordagem para desenvolvimento de objetos de aprendizagem multimídias e interativos

Marco Aurélio Graciotto Silva

***Orientador*: Prof. Dr. José Carlos Maldonado**

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP como parte dos requisitos para obtenção do título de Doutor em Ciências – Ciências de Computação e Matemática Computacional. VERSÃO REVISADA

USP – São Carlos
Março de 2012

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

G731l

Graciotto Silva, Marco Aurélio
LOD: uma abordagem para desenvolvimento de objetos de aprendizagem multimídias e interativos / Marco Aurélio Graciotto Silva; orientador José Carlos Maldonado. -- São Carlos, 2012.
221 p.

Tese (Doutorado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2012.

1. objeto de aprendizagem. 2. televisão digital. 3. desenvolvimento baseado em modelos. 4. engenharia de software. I. Maldonado, José Carlos, orient. II. Título.

Agradecimentos

Uma pesquisa não se faz sozinho, uma única pesquisa não faz o doutor. A formação envolve o todo e a todos, nunca sozinho. Por esses anos de USP, sinto-me recompensado pelas oportunidades que tive e pelo o que aprendi, e me esforçarei em retribuir a contento este mundo que tanto acredita em nós, futuros doutores.

Quanto à formação acadêmica, agradeço primeiramente ao meu orientador, o Prof. Maldonado, que depositou confiança em meu trabalho, aceitando-me no grupo de prolíficos pesquisadores que formou em suas décadas de dedicação acadêmica.

Agradecimentos especiais também à Profª. Ellen, que, na prática, coorientou este trabalhado, contribuindo com ricas discussões sobre o tema desta tese, permitindo a elucidação de dúvidas, criação de várias outras e o delineamento de soluções para os inúmeros problemas que existem na modelagem de objetos de aprendizagem.

O convívio no grupo de pesquisa, em nosso Laboratório de Engenharia de Software (LabES) e no seu jovem e irmão Laboratório de Computação Aplicada à Educação (CAEd), também foi fonte de valiosas discussões não apenas sobre objetos de aprendizagem, mas sobre todos os temas sobre a área, em um espaço de mútua ajuda e crescimento pessoal e do grupo. Não enumerarei todos os colegas de laboratórios, são muitos desde que ingressei no ICMC (com o LabES ainda na sala 1-006), mas saibam que todos colaboraram na formação desta pessoa e estão devidamente registrados em minhas memórias.

Meus agradecimentos também aos professores, alunos e funcionários do CePES (FACIN) e da PUCRS Virtual, em especial ao Prof. Avelino Zorzo e à Profª. Lucia Giraffa, pela acolhida e atenção dedicados durante o meu estágio na PUCRS (PROCAD).

A universidade não é apenas pesquisa, mas uma formação holística. Para um doutorando, estar preparado para o ingresso na academia não significa apenas pesquisa, mas também educação e administração. Neste último ano e meio, como representante discente de todos os pós-graduandos do ICMC, reunido com mais de século de experiência universitária da maior instituição universitária do mundo e um dos maiores programa de pós-graduação em Computação do Brasil, pude aprender um pouco de como funciona nossa universidade e contribuir para o seu melhor funcionamento. Espero, no futuro próximo, repassar e aplicar

esse valioso conhecimento em outras instituições de ensino. Meu imenso obrigado aos membros da Comissão de Pós-Graduação e aos alunos da pós-graduação que tanto prezo.

Este trabalho contou com o apoio do Projeto QualiPSo (IST-FP6-034763) e das agências de fomento CNPq (Processos 142381/2009-8 e 555565/2009-2) e CAPES (Projeto PROCAD 191/2007).

Encerrando os cumprimentos na área acadêmica, gostaria de estender meus agradecimentos aos funcionários do ICMC, em especial aos da Secretaria da Pós-Graduação e do Setor de Convênios, pelo cordial e efetivo serviço prestado.

Nada disso seria possível se não fosse o apoio incansável de minha família: Apolo e Rosa, meus pais, Victor Augustus e Melissa, meus irmãos. Obrigado pela paciência e pela fé em mim depositadas.

Finalmente, meus eternos agradecimentos ao amor de minha vida, minha Aretha. Obrigado Deus por me mostrar minha cara metade e lhe dar forças para aturar um doutorando em fase final de escrita de tese.

Sumário

1 Introdução **1**

2 Objetos de aprendizagem: caracterização dos aspectos educacionais e computacionais **11**

2.1 Conceitos de educação e o papel dos objetos de aprendizagem 12

2.1.1 Teorias educacionais 13

2.1.2 Características do processo instrucional 14

2.1.2.1 Formalidade 14

2.1.2.2 Modalidade 15

2.1.2.3 Acesso 15

2.1.2.4 Formas de interação 16

2.1.2.5 Formas de avaliação 16

2.1.3 Características de projeto instrucional 19

2.1.4 Educação com informática 22

2.2 Objetos de aprendizagem 22

2.2.1 Propriedades dos objetos de aprendizagem 23

2.2.2 Taxonomia de objetos de aprendizagem 24

2.2.3 Representação de objetos de aprendizagem 27

2.3 Abordagens para desenvolvimento de objetos de aprendizagem 29

2.3.1 Métodos 29

2.3.1.1 Métodos precursores baseados em hipertexto 30

2.3.1.2 Métodos baseados em linguagens de modelagem educacional 31

2.3.1.3 Métodos baseados em grafos de composição 33

2.3.1.4 Métodos baseados em Rede de Petri 34

2.3.1.5 Métodos baseados em máquinas de estados 36

2.3.1.6 Métodos de desenvolvimento baseados em modelos (MDD) . 41
2.3.2 Processos 44
2.3.2.1 SP-DEM 45
2.3.2.2 ISDMeLO 46
2.3.2.3 RIVED 47
2.3.2.4 Sophia 48
2.3.2.5 MISA 49
2.4 Considerações finais 50

3 Objetos de aprendizagem multimídias e interativos para televisão digital 53
3.1 Televisão digital 54
3.1.1 Sistema Brasileiro de Televisão Digital 58
3.1.2 Plataforma procedimental e Ginga-J 59
3.1.3 Plataforma declarativa e Ginga-NCL 61
3.2 Interatividade em televisão digital 65
3.3 Objetos de aprendizagem para televisão digital 68
3.4 Considerações finais 71

4 Requisitos para desenvolvimento de objetos de aprendizagem multimídias e interativos 73
4.1 Requisitos educacionais para objetos de aprendizagem 74
4.1.1 Requisitos quanto à educação 74
4.1.2 Requisitos quanto ao processo instrucional 77
4.1.3 Requisitos quanto ao projeto instrucional 79
4.2 Requisitos computacionais para objetos de aprendizagem 80
4.2.1 Requisitos de taxonomia 81
4.2.2 Requisitos de representação 82
4.2.3 Requisitos para abordagens de desenvolvimento 85
4.2.3.1 Requisitos para métodos de modelagem de objetos de aprendizagem 85
4.2.3.2 Requisitos para processos de desenvolvimento de objetos de aprendizagem 88
4.3 Requisitos para objetos de aprendizagem multimídias e interativos 90
4.3.1 Protocolo da revisão sistemática 91

4.3.2 Resultados 93
4.3.3 Extração e análise de dados 94
4.3.4 Consolidação da análise 98
4.4 Requisitos para objetos de aprendizagem para TVD 100
4.5 Considerações finais 102

5 LOD: Abordagem para desenvolvimento de objetos de aprendizagem 105
5.1 LODM: Learning Object Development Method 106
5.1.1 Modelo conceitual 107
5.1.2 Modelo instrucional 114
5.1.3 Modelo de interação 121
5.1.4 Geração de objetos de aprendizagem 124
5.2 LODP: Learning Object Development Process 126
5.2.1 SPLOD: modelo de referência para objetos de aprendizagem 126
5.2.1.1 Processos de acordo 127
5.2.1.2 Processos organizacionais 130
5.2.1.3 Processos de projeto 132
5.2.1.4 Processos técnicos 135
5.2.2 LODP: atores 138
5.2.3 LODP: processos 138
5.2.3.1 Processo de estabelecimento da comunidade 139
5.2.3.2 Processos de habilitação e manutenção da comunidade . . . 141
5.2.3.3 Processos de projeto 143
5.2.3.4 Processos técnicos 145
5.2.4 LODP: ciclo de vida 148
5.3 LODE: Learning Object Development Environment 150
5.3.1 Geração de apresentações multimídia 154
5.3.2 Geração de documentos NCL 157
5.4 Considerações finais 165

6 Estudos de viabilidade 167
6.1 Estudo 1 – Apresentação multimídia 168
6.1.1 Planejamento 168
6.1.2 Resultados 169

6.1.3 Análise 175
6.2 Estudo 2 – Programa multimídia e interativo de televisão 175
6.2.1 Planejamento 176
6.2.2 Resultados 177
6.2.3 Análise 179
6.3 Análise da abordagem LOD 180
6.3.1 Resultados em relação aos requisitos educacionais 181
6.3.2 Resultados em relação aos requisitos computacionais 182
6.3.3 Resultados em relação aos requisitos de interação 183
6.3.4 Trabalhos relacionados 184
6.4 Considerações finais 185

7 Conclusões **187**

Referências **194**

Lista de figuras

1.1 Método baseado em evidências para a definição de novas tecnologias de software 8

2.1 Taxonomias de objetivos educacionais para os domínios cognitivo, afetivo e psicomotor. 17
2.2 Modelo do projeto instrucional segundo o ADDIE. 20
2.3 Etapas do projeto instrucional do modelo de Dick e Carey 20
2.4 Modelo conceitual da especificação IMS LD 32
2.5 IMA-CID: Modelos utilizados em cada perspectiva da abordagem e a forma com que eles se relacionam. 36
2.6 Modelo conceitual (IMA-CID) sobre o critério de análise de mutantes 37
2.7 IMA-CID: Elementos do modelo instrucional. 39
2.8 Modelo instrucional (IMA-CID) sobre o critério de análise de mutantes . . . 40
2.9 IMA-CID: Elementos do modelo didático. 40
2.10 Processo de transformação de modelos e geração de objeto de aprendizagem personalizável 42
2.11 DSL para descrição arquitetura de um CLO 43
2.12 Exemplo com transformações de modelos conforme o método MDLD 44
2.13 Estrutura geral do SP-DEM 45
2.14 Estragégia *top-down* utilizada pelo método ISDMeLO. 46
2.15 RIVED: Processo de desenvolvimento de objetos de aprendizagem 48

3.1 Camadas de um sistema de televisão digital. 55
3.2 Arquitetura do Ginga 58
3.3 Núcleo do Ginga e partes comuns aos subsistemas Ginga-J e Ginga-NCL . . 59
3.4 Componentes da API Java do Ginga-J. 60
3.5 Componentes do subsistema Ginga-NCL 61

3.6 Modelo de aninhamento de contextos (NCM) implementado pela NCL. . . . 62

4.1 Elementos educacionais associados a um objeto de aprendizagem. 75
4.2 Tipos de objetos de aprendizagem que uma abordagem de desenvolvimento deve contemplar. 82
4.3 Elementos requeridos para a representação de um objeto de aprendizagem. . 83
4.4 Elementos requeridos em um método de modelagem de objetos de aprendizagem. 86
4.5 Elementos requeridos em um processo de desenvolvimento de objetos de aprendizagem. 89
4.6 Projeção dos estudos recuperados da ACM e respectiva classificação 93
4.7 Agrupamento dos estudos selecionados de acordo com os termos extraídos do título, resumo e evidências. 94
4.8 Contextos em que os objetos de aprendizagem são utilizados e atores que os utilizam. 95
4.9 Modalidade de ensino em que os objetos de aprendizagem foram utilizados. . 95
4.10 Formas de interação utilizadas em objetos de aprendizagem. 96
4.11 Tipos de interação e técnicas de elicitação de requisitos de interação 97
4.12 Técnicas de projeto e formas de implementação de objetos de aprendizagem. 98
4.13 Formas de avaliação da qualidade dos objetos de aprendizagem e de sua interação. 98

5.1 LODM: Visão geral do método. 107
5.2 LODM: Diagrama de classe dos elementos do modelo conceitual 109
5.3 Modelo conceitual sobre teste de software para um curso introdutório sobre o tema. 109
5.4 LODM: Diagrama de classe contendo os elementos do modelo instrucional. . 116
5.5 LODM: Diagrama dos elementos do perfil instrucional. 118
5.6 LODM: Perfis de itens de informação aplicáveis a um modelo instrucional. . 118
5.7 Modelo instrucional para um objeto de aprendizagem sobre teste de software. 119
5.8 Valores definidos para os *tagged values* associados ao item de informação `Software failure` do modelo instrucional da Figura 5.7. 120
5.9 LODM: Elementos do modelo de interação. 122
5.10 LODM: Diagrama dos elementos do perfil de interação. 122
5.11 Modelo de interação para um objeto de aprendizagem sobre teste de software. 123
5.12 Elementos de um modelo de referência para processos. 127

5.13 SPLOD: Visão geral do modelo de referência para desenvolvimento de objetos de aprendizagem. 128
5.14 SPLOD: Grupo de processos de acordo 128
5.15 SPLOD: Grupo de processos organizacionais 130
5.16 SPLOD: Grupo de processos de projeto 133
5.17 SPLOD: Grupo de processos técnicos 136
5.18 LODP: Visão geral. 139
5.19 LODP: Processo para estabelecimento de comunidade 141
5.20 LODP: Processos para manutenção de comunidade 142
5.21 LODP: Processos de projeto 144
5.22 LODP: Processos técnicos 146
5.23 LODP: Ciclo de vida de um objeto de aprendizagem. 149
5.24 LODE: Diagrama dos principais pacotes. 151
5.25 LODE: Pacotes de mineração de dados e de transformação. 151
5.26 LODE: Principais classes do pacote de transformadores. 152

6.1 JaBUTi-TT: Modelo conceitual sobre a JaBUTi e sua relação com as atividades de teste de software. 170
6.2 JaBUTi-TT: Modelo instrucional sobre a JaBUTi e sua relação com as atividades de teste de software. 171
6.3 JaBUTi-TT: Página *wiki* com recursos relacionados a itens de informação associados ao conceito de requisito de teste. 171
6.4 JaBUTi-TT: Quadro que representa um estado do modelo de interação. . . . 172
6.5 JaBUTi-TT: Quadro sobre um tipo de oráculo de teste. 173
6.6 JaBUTi-TT: Quadro sobre a visualização de requisitos de teste. 174
6.7 TestingAriane501: Modelo conceitual (LODM). 177
6.8 TestingAriane501: Modelo instrucional – primeira parte (LODM). 178
6.9 TestingAriane501: Modelo instrucional – segunda parte (LODM). 179
6.10 TestingAriane501: Modelo de interação (LODM). 180

Lista de tabelas

2.1 Componentes de um objeto de aprendizagem personalizável e conjunto de operações permitidas. 43

2.2 Correspondência entre o modelo ADDIE e os processos fundamentais do SP-DEM 46

4.1 Relação entre a interatividade e os fatores analisados. 99

5.1 Termos extraídos de um texto didático sobre teste de software e suas respectivas frequências. 112

6.1 JaBUTi-TT: Quantidade de elementos dos modelos conceitual e instrucional. 172

Lista de fragmentos de código

3.1 Exemplo de documento NCL. 63
3.2 Exemplo de âncora de conteúdo em um documento NCL. 64
3.3 Exemplo de definição de um conector causal e de um elo deste tipo de conector. 64
3.4 Exemplo de definição de um conector causal e de um elo deste tipo de conector. 64
3.5 Exemplo de um elo que define um parâmetro para a associação com o conector. 65
5.1 Modelo em LaTeX/Beamer para a definição de um conceito. 154
5.2 LODE: Definição de propriedades de navegação em uma apresentação LaTeX/Beamer. 156
5.3 LODE: Exemplo de definição de transição entre estados. 156
5.4 LODE: Exemplo de representação de estados em um documento LaTeX/Beamer. 157
5.5 LODE: Geração de contexto a partir de estado básico do modelo de interação. 158
5.6 LODE: Descritor para mídias com tempo de vida infinito. 158
5.7 LODE: Geração de um contexto a partir de um estado composto do tipo AND do modelo de interação. 158
5.8 LODE: Regiões e descritores para mídias reproduzidas simultaneamente. . . 159
5.9 LODE: Geração de um contexto a partir de um estado composto do tipo XOR do modelo de interação. 160
5.10 LODE: Ligação que estabelece o acionamento de um contexto ao término de outro. 161
5.11 LODE: Definição do conector onEndStart. 161
5.12 LODE: Geração de um contexto de adaptação a partir de um estado composto do tipo XOR DD do modelo de interação. 162
5.13 LODE: Regras para seleção de opções do contexto adaptativo que representa um estado XOR DD. 162
5.14 LODE: Definição do contexto para escolha de opção e acionamento do respectivo contexto. 163
5.15 LODE: Definição dos descritores referentes às opções para um estado XOR DD. 164
5.16 LODE: Definição de ligações referentes às opções para um estado XOR DD. 164
5.17 LODE: Definição de ligações referentes às opções para um estado XOR DD. 165

6.1 JaBUTi-TT: Código do quadro sobre visualização de requisitos de teste . . . 174

Lista de acrônimos

AAC-HEv1 *Advanced Audio Coding – High Efficiency version 1*

AAC-HEv2 *Advanced Audio Coding – High Efficiency version 2*

AAC-LC *Advanced Audio Coding – Low Complexity*

ACM *Association for Computing Machinery* 93

ADDIE *Analysis, Design, Development, Implementation, and Evaluation* 19

ADL *Advanced Distributed Learning* 29

AHAM *Adaptive Hypermedia Application Model* 33

AICC *Aviation Industry CBT Committee* 29

API *Application Program Interface* 59

ATSC *Advanced Television Systems Committee* 55

AVS *Audio Video Standard*

CCSL Centro de Competência em Software Livre 7

CDC *Java Connected Device Configuration* 59

CDT *Component Display Theory* 38

CLO *Customized Learning Object*

CMI *Computer-Managed Instruction*

CSS *Cascading Style Sheets*

CXL *Concept Mapping Extensible Language* 112

DAVIC *Digital Audio Video Council*

DC *Dublin Core* 27

DSL *Domain Specific Language* 41

DTMB *Digital Terrestrial Multimedia Broadcast* 55

DVB *Digital Video Broadcasting* 56

EaD Educação a Distância

EHDM *Educational Hyperdocuments Design Method* 30

EIT *Event Information Table* 176

EML *Educational Modelling Language*

ESA *European Space Agency* 177

FP *Java Foundation Profile* 59

FSM *Finite State Machine* 38

GEM *Globally Extensible MHP* 58

GLO *Generative Learning Object*

HACP *HTTP AICC Communication Protocol* 83

HAVi *Home Audio Video Interoperability*

HDM *Hypertext Design Model* 30

HMBS *Hypermedia Model Based on Statecharts* 30

HTML *Hypertext Markup Language* 27

IEEE *Institute of Electrical and Electronics Engineers* 93

ILO *Intensional Learning Object* 33

IMA-CID *Integrated Modelling Approach – Conceptual, Instructional and Didactic* 36

IMS *IMS Global Learning Consortium* 29

IMS CC *IMS Common Cartridge* 84

IMS CP *IMS Content Packaging* 44

IMS LD *IMS Learning Design* 28

IMS LTI *IMS Learning Tools Interoperability* 84

IMS SS *IMS Simple Sequencing* 28

INCT-SEC Instituto Nacional de Ciências e Tecnologia em Sistemas Embarcados Críticos 7

IPTV *Internet Protocol Television* 4

ISD *Instrucional System Design* 20

ISDB *Integrated Service Digital Broadcasting* 55

JXTA *Java Juxtapose*

LMS *Learning Management System*

LOCPN *Learning Object production with Color Petri Net* 35

LOD *Learning Object Development* 167
LODE *Learning Object Development Environment* 105
LODE-TV *Learning Object Development Environment for Television*
LODM *Learning Object Development Method* 105
LODP *Learning Object Development Process* 105
LOM *Learning Object Metadata* 27
LWUIT *Java LightWeight User Interface Toolkit* 60
MDD *Model-Driven Development* 41
MDE *Model for Distance Education* 30
MDLD *Model-Driven Learning Design* 43
MERLOT *Multimedia Educational Resource for Learning and Online Teaching*
MHP *Multimedia Home Plataform* 69
MISA *Méthode d'ingénierie de systèmes d'apprentissage* 49
NCL *Nested Context Language* 57
NCM *Nested Context Model* 62
OER *Open Educational Resources*
OFDM *Orthogonal frequency-division multiplexing*
OOCM *Object Oriented Course Modeling* 35
OOHDM *Object-Oriented Hypermedia Design Model* 30
OUEML *Open Universiteit Educational Modeling Language* 31
OU *Open University* 78
OUNL *Open University of the Netherlands* 31
P2P *Peer-to-peer* 69
PBP *Java Personal Basis Profile* 59
PDF *Portable Document Format* 153
PICO Problema ou População, Intervenção, Comparação e Resultados esperados
PIF *Package Interchange File* 29
PIP *Picture in Picture*
PoEML *Perspective-oriented Educational Modeling Language* 32
PP *Java Personal Profile* 59

PSI	*Program Specific Information*	
QualiPSo	*Quality Platform for Open Source Software*	7
RDF	*Resource Description Framework*	27
RIVED	Rede Internacional Virtual de Educação	35
RLO	*Reusable Learning Object*	
RMDM	*Relationship Management Data Model*	30
RMI	*Java Remote Method Invocation*	
RMM	*Relationship Management Methodology*	30
SBTVD	Sistema Brasileiro de Televisão Digital	8
SCORM	*Sharable Content Object Reference Model*	34
SCORM RTE	*SCORM Run-Time Environment*	83
SDTV	*Standard Definition Television*	
SP-DEM	*Standard Process – Development of Educational Modules*	46
SPEM	*Software Process Engineering Metamodel*	192
SPLOD	*Standard Process for Learning Objects Development*	105
STB	*Set-Top Box*	
TVD	Televisão Digital	4
UML	*Unified Modeling Language*	117
UoL	*Unit of Learning*	
URL	*Uniform Resource Locator*	
WIMP	*Windows, Icons, Menus, and Pointing Devices*	
XMI	*XML Metadata Interchange*	150
XML	*eXtensible Markup Language*	27

Resumo

Objetos de aprendizagem (OA) compõem uma das abordagens adotadas para tratar a crescente complexidade da educação com auxílio de computadores, promovendo o reúso e a qualidade de materiais didáticos. Tais benefícios podem ser ampliados ao incluir questões de projeto instrucional e associá-las às características multimídias e interativas dos dispositivos computacionais, sistematizando-se a produção de objetos de aprendizagem. Nesse contexto, esta tese define uma abordagem para o desenvolvimento de objetos de aprendizagem multimídias e interativos considerando o cenário de televisão digital. Essa abordagem, denominada LOD (*Learning Object Development*) consiste de um modelo de processo ou processo-padrão para desenvolvimento de objetos de aprendizagem (SPLOD – *Standard Process for Learning Object Development*), a partir do qual se define um processo (LODP – *Learning Object Development Process*) adequado para a construção de objetos de aprendizagem como recursos educacionais abertos (análogos a software livre); de um método de desenvolvimento dirigido a modelos, denominado LODM (*Learning Object Development Method*), compreendendo a modelagem conceitual, instrucional e de interação do objeto de aprendizagem; e de um conjunto de ferramentas que estabelecem um protótipo de ambiente para o desenvolvimento de objetos de aprendizagem (LODE – *Learning Object Development Environment*) considerando o método LODM e o processo LODP. A abordagem LOD foi preliminarmente avaliada quanto à engenharia de objetos de aprendizagem multimídia e interativos, representados por apresentações multimídia e aplicações interativas para televisão digital para a plataforma Ginga do Sistema Brasileiro de Televisão Digital (SBTVD). Os resultados apontam para os benefícios da abordagem integrada para o desenvolvimento, com a geração de objetos de aprendizagem multimídias e interativos de forma mais ágil e sistemática, além de promover o reúso desde os primeiros passos da modelagem dos objetos de aprendizagem.

Palavras-chave: LOD, objeto de aprendizagem, método, processo, interatividade, requisitos, MDD, SBTVD, Ginga.

Abstract

LOD: an approach to the development of multimedia and interactive learning objects

Learning objects are an approach to address the increasingly complexity of computer-supported education, fostering reuse and learning material quality. Such benefits can be improved by considering instructional design and the multimedia and interactivity features of current computing devices, providing a systematic method to develop learning objects. Considering such context, this thesis defines an approach to the development of multimedia and interactive learning objects, considering the digital television scenario. This approach is named LOD (Learning Object Development) and is composed of (1) a reference process (SPLOD – Standard Process for Learning Object Development) from which we defined a process for learning objects development (LODP – Learning Object Development Process) suitable for engineering of open educational resources; (2) a model-driven development method (LODM – Learning Object Development Method) that integrates conceptual, instructional and interaction modeling of learning objects; and (3) a set of tools that establish a prototype for an environment for learning object development named LODE (Learning Object Development Environment). The LOD approach was evaluated with multimedia and interactive learning objects development realized as multimedia presentations and interactive digital television applications aimed at the Brazilian Digital Television System and its Ginga middleware. The results evince the benefits of the integrated approach, providing an agile and systematic development and fostering the reuse since the initial models of the learning object engineering process.

Keywords: LOD, learning object, method, process, interactivity, requirements, MDD, ISDB-Tb, Ginga.

Capítulo 1

Introdução

A educação é um dos principais meios de desenvolvimento da sociedade. A criação, o compartilhamento e o aprendizado de conhecimentos e habilidades permitem que cada indivíduo e grupo desempenhem seu papel na comunidade de forma eficiente e eficaz. No entanto, para que esse efeito seja significativo, a educação deve ser ampla e constante.

Observam-se atualmente diversas iniciativas de ensino e aprendizagem no âmbito formal e informal. Enquanto governo, é política pública o ensino formal nos níveis básico (fundamental e médio) e superior, preparando-se as pessoas para fins técnicos (indústria e comércio) ou acadêmicos. O setor terciário também educa, realizando treinamentos e certificando profissionais para a execução de tarefas específicas. A sociedade como um todo, com apoio do governo e outros setores, desenvolve atividades de educação continuada, permitindo a constante atualização das pessoas não apenas em questões técnicas ou acadêmicas, mas também de novos conhecimentos necessários no cotidiano em decorrência dos avanços científicos proporcionados pela própria educação.

Um item comum a essas formas de educação é a necessidade de disseminar os conceitos e habilidades, retratados como conteúdos ou material didático. Antigamente, o limite era a fala humana; depois a escrita transmitida no papel; no século passado a mídia (televisão) e a computação. Hoje a realidade apresenta a convergência de todas essas abordagens: os computadores exercem o papel de rápida disseminação de informações antes restrito às mídias

radiofônicas e televisivas; a televisão digitaliza-se para entreter e educar em grupo (no modo tradicional de "assistir" televisão, em um ambiente comum com pessoas ao redor do aparelho fixo) ou individualmente (nos cada vez mais presentes e sofisticados telefones celulares); o papel transforma-se em mídia eletrônica, aliando o prazer da leitura, a comodidade de armazenar milhares de livros e o conforto de um dispositivo de reduzidas dimensões e pouco peso.

Entretanto, o desenvolvimento de atividades de aprendizagem e dos recursos educacionais necessários a sua execução não se faz sem dificuldades. Observando os avanços na compreensão do ser humano quanto aos aspectos cognitivos, afetivos e psicomotores e os efeitos deles na aprendizagem, representados nas teorias educacionais, e dos métodos de instrução para fomentar essa aprendizagem, constata-se que o desenvolvimento de material didático e das atividades possui um custo elevado e resultados insatisfatórios para todos os atores envolvidos: aprendizes, professores e instituições.

Uma razão para esse cenário indesejado é a intersecção entre a educação e a tecnologia de informação. Em situações anteriores, era possível educar sem considerar aspectos computacionais ou abrangendo apenas itens simples, passíveis de obtenção por treinamentos. Essa não é mais a realidade. A convergência tecnológica expõe problemas importantes de computação que não podem ser sobrepujados apenas com boa vontade. Logo, torna-se necessária a integração entre especialistas de áreas distintas, complementando as competências de forma sinérgica e colaborativa (QUADROS; MARTINS, 2005).

As aplicações educacionais – tais como os tutores inteligentes, simuladores, micromundos, ambientes virtuais de aprendizagem, dentre outros – são um importante resultado da atuação conjunta de educação e informática. Embora os benefícios proporcionados por elas não tenham causado a revolução que se imaginava no passado, seria impensável educar hoje sem essas ferramentas. A questão não é propor um novo modelo que sobrepuja esse, mas de evoluir gradativamente o atual e, principalmente, desenvolver ferramentas que sustentem essa evolução.

Nessa perspectiva, uma iniciativa é o desenvolvimento sistemático de material didático a ser utilizado em aplicações educacionais inseridas no contexto do processo instrucional. A instrução, por si, é suprida de métodos instrucionais e de processos que a organizam com foco na perspectiva educacional (DICK *et al.*, 2005). Em contrapartida, em computação define-se a figura do objeto de aprendizagem para materializar o material didático, utilizado na instrução, em um recurso computacional que promove o reúso e reduz o custo de planejamento e desenvolvimento das atividades de aprendizagem (WILEY, 2000, p. 7).

O conceito de objeto de aprendizagem é inspirado no paradigma de orientação a objeto de Engenharia de Software. Ele possui um conjunto de dados (conteúdos didáticos) e um comportamento definido em função de um objetivo educacional específico. Semanticamente, a

coesão (proporcionada pelo foco em um objetivo bem definido e provavelmente simples) e baixo acoplamento (decorrente da independência do objeto quanto a outros objetos de aprendizagem para satisfazer o objetivo) seriam suficientes para escolher objetos apropriados para uma aplicação educacional, mas, a partir do momento em que os requisitos das aplicações são elaborados e o universo de objetos disponíveis amplia-se, torna-se necessário o estabelecimento de mecanismos que auxiliem os interessados (desenvolvedores, professores e aprendizes, para citar os principais) a selecionar e utilizar os objetos mais adequados.

A primeira solução adotada foi a definição de metadados para a descrição das características dos objetos de aprendizagem, tanto da perspectiva educacional quanto computacional. Essas informações são necessárias também para a escolha dos objetos pelos interessados, mas, quando formalmente especificadas, possibilitam a automatização dessa tarefa e permitem outras mais sofisticadas, tal como a personalização do objeto de aprendizagem em função do contexto da atividade educacional.

Embora os metadados satisfaçam algumas questões, as especificidades da educação a distância (seja essa total ou parcial, tal como em educação híbrida ou *blended learning*) requerem um grau de automatização de atividades além daquele praticado com metadados. Exemplos disso são a especificação de sequências de entrega e a execução de objetos de aprendizagem em função das atividades planejadas para atingir um conjunto de objetivos educacionais, assim como a realização de avaliações diagnósticas, formativas e sumativas. Técnicas de especificação formal, normalmente utilizadas para a definição de software e hipertextos, emprestam suas capacidades para modelar essas situações e permitir a representação, execução e interação dos objetos de aprendizagem.

Ao final, observa-se um paralelo entre objetos de aprendizagem e software, seja pelos paradigmas adotados (orientação a objeto e a componentização), os modelos computacionais (as diversas técnicas de especificação formal e linguagens de modelagem) e, ultimamente, quanto aos processos de desenvolvimento e controle de qualidade (DOUGLAS, 2001; BOYLE, 2003; BARBOSA, 2004).

Motivação

As iniciativas de ensino com uso da Internet, mais especificamente da Web (*e-learning*), aproveitam-se das propriedades multimídia e da agilidade e facilidade de comunicação para construir ambientes de ensino a distância ou híbrido. Nesse cenário, os objetos de aprendizagem são utilizados principalmente para a representação de recursos educacionais (documentos e apresentações) e avaliações associados a uma sequência de atividades (por si também um objeto de aprendizagem).

No entanto, é necessário recordar que a Internet não está disponível para todos (e isso

não é um fato válido apenas no Brasil). Embora seja possível garantir o acesso à grande rede, em especial às aplicações Web, em ambientes públicos e escolas, educação não se faz em um espaço determinado de tempo e espaço. Principalmente para educação continuada (*life-long learning*), deve-se facilitar o acesso às oportunidades de aprendizado, oferecendo-as nos diferentes meios de comunicação.

A televisão é vista como uma forma de englobar todos nessa iniciativa educacional. Trata-se de um veículo presente na maioria dos espaços (aproximadamente 96% das residências brasileiras possuem ao menos um televisor fixo) e acreditado. Diversas iniciativas lograram sucesso no uso da televisão na educação formal, tal como a série de telecursos e outros programas educativos transmitidos em rede aberta (BARRETO, 2009). No entanto, o ensino por televisão era limitado pela ausência de interação com os aprendizes: a transmissão de informações ocorria apenas em uma via e não existia qualquer programação possível no lado do usuário (no máximo a possibilidade de gravar um programa e assisti-lo em um momento posterior).

A Televisão Digital (TVD) define importantes mudanças nesse cenário. Além de melhorar aquilo que já existia da televisão (qualidade de imagem e som), ela permite a interação na forma de atividades definidas por software transmitido com os programas de televisão e a comunicação com outros usuários pelo canal de retorno. Além disso, a TVD não se restringe a aparelhos fixos, mas também pode ser recebida em dispositivos móveis, o que propicia o seu acesso em diversos ambientes.

Os objetos de aprendizagem devem tomar partido das características da televisão digital, facilitando a adaptação de programas para ensino e definindo novas abordagens para tirar proveito das características do sistema. Por exemplo, ao contrário das ações de *e-learning*, o foco da televisão é a mídia em transmissão contínua (vídeo e áudio), o que requer mecanismos distintos para sincronização.

Deve-se atentar também ao emprego dos padrões utilizados na televisão digital para a programação procedural e, principalmente, declarativa dos aplicativos interativos no contexto da Internet, na hoje denominada *Internet Protocol Television* (IPTV).

Finalmente, e não menos importante, deve-se alavancar o reúso de objetos de aprendizagem interativos, permitindo o desenvolvimento colaborativo e aberto deles. Esse ponto está em consonância com esforços internacionais para ampliar o acesso à educação, seja pelo desenvolvimento colaborativo de recursos educacionais abertos (ALBRIGHT, 2005), pela livre distribuição da material didático referente a cursos – tal como proposto no projeto *OpenCourseWare* (OCW Consortium, 2002; TERRELL; CAUDILL, 2012) – ou pela realização de cursos informais nos moldes do *Khan Academy* (KHAN *et al.*, 2006) ou dos cursos abertos de *Stanford* lançados em 2011. Trata-se da construção de atividades elaboradas de aprendizagem e acessíveis de forma não discriminatória, cunhadas pela interação social dos usuários.

A criação de objetos de aprendizagem abertos, multimídia e interativos de qualidade é possível com o emprego de processos, métodos e ferramentas adequados, que conciliem a liberdade necessária para definir e modelar as atividades e objetos de aprendizagem com os instrumentos para desenvolvê-los de forma controlada, observando os atributos de qualidade de produto e de processo.

Na perspectiva de processos, o desenvolvimento de objetos de aprendizagem conjuga a técnica computacional com a instrucional em grupos multidisciplinares. Em um esforço para construir repositórios de objetos, projetos internacionais (*e.g.*, MERLOT) e nacionais (*e.g.*, RIVED) definiram processos que combinam atividades de projeto instrucional com aquelas de desenvolvimento de software, facilitando a comunicação entre o professor (ou projetista instrucional) e o programador.

Entretanto, observa-se que tais processos buscam reduzir o desenvolvimento de objetos de aprendizagem à pura e simplesmente programação. Tal visão é equivocada, fato esse reconhecido desde o estabelecimento da área de Engenharia de Software. O desenvolvimento de software requer, no mínimo, atividades de análise de requisitos, o desenvolvimento de modelos, programação, atividades de validação, verificação e testes, com atenção a atributos de qualidade para que, além do produto final ter qualidade, que o processo seja executado com custos e prazos controlados e razoáveis. Tais considerações também são válidas no contexto de desenvolvimento de objetos de aprendizagem.

Recentemente, trabalhos nessa linha foram desenvolvidos para o estabelecimento de modelos de processos de desenvolvimento, também denominados de processos-padrão (BARBOSA, 2004; BARBOSA; MALDONADO, 2011a). Um processo-padrão define os processos, atividades e resultados esperados quanto ao desenvolvimento de um produto (HUMPHREY, 1989). Um processo-padrão para objetos de aprendizagem define, além da integração com a instrução, a execução de atividades cobrindo todo o ciclo de vida de um objeto de aprendizagem, desde sua concepção até a operacionalização, com especial atenção àquelas específicas aos requisitos, modelos e construção dos objetos (BARBOSA, 2004). A partir de um processo-padrão, é possível definir processos específicos de forma a atender os modelos estabelecidos no padrão e de acordo com a maturidade da organização que o executará. Assim, assegura-se que serão realizadas as atividades necessárias ao desenvolvimento de objetos de aprendizagem de qualidade condizentes com as capacidades disponíveis.

Os processos são uma solução-macro para a questão do desenvolvimento de objetos de aprendizagem, mas é necessário considerar, também, os métodos específicos (e que atendem parte dos processos definidos em um processo-padrão). Tais métodos devem-se conciliar os interesses educacionais e computacionais, conferindo qualidade ao produto. Por exemplo, soluções que se baseiam em especificações escritas em linguagem natural, tal como um roteiro de aprendizagem, embora acessíveis a todos os atores, potencialmente apresentam problemas

quanto à correção, completeza, existência de ambiguidades, inexistência de rastreabilidade dos cenários e atos definidos com os objetivos educacionais. Embora facilitar a comunicação seja importante, abrir mão de todos os demais aspectos técnicos não é uma solução adequada.

Alguns métodos de desenvolvimento evoluíram nessa questão ao estabelecer linguagens para modelagem educacional e adotando representações formais ou, ao menos, padronizadas para a definição de atividades de aprendizagem e do uso dos objetos de aprendizagem. Ao invés de um roteiro servir de entrada para a programação, define-se um passo intermediário para permitir a correta e completa especificação dos casos de uso, formalizando-o e mitigando erros decorrentes da interpretação da linguagem natural.

Considerando tais aspectos, este trabalho estuda a aplicação de técnicas de Engenharia de Software no desenvolvimento de objetos de aprendizagem, definindo um processo e um método adequado aos requisitos de objetos de aprendizagem interativos e multimídia com ênfase em televisão digital e na perspectiva de recursos de aprendizagem livres.

Objetivos

Esta pesquisa de doutorado objetiva a definição de subsídios e de uma abordagem para o desenvolvimento de objetos de aprendizagem para plataformas multimídia e interativas, visando à televisão digital como plataforma alvo. As metas desta pesquisa são:

- Estabelecimento de requisitos para o desenvolvimento de objetos de aprendizagem multimídias e interativos.
- Especificação de um método para a modelagem de objetos de aprendizagem multimídia e interativos.
- Especificação de um processo-padrão para o desenvolvimento de objetos de aprendizagem a partir do processo de referência da ISO 12207:2008 (ISO/IEC, 2008a) e incorporando os aspectos estudados em trabalhos relacionados.
- Implementação de um protótipo de ambiente para desenvolvimento de objetos de aprendizagem abertos.

Os requisitos e o processo-padrão constituem importantes subsídios para a definição de abordagens para o desenvolvimento de objetos de aprendizagem, com especial atenção para questões de interatividade relacionadas à televisão digital nesta tese. O método de desenvolvimento, concebido para atender os processos técnicos do processo-padrão e dos requisitos para objetos de aprendizagem para televisão digital, adota a abordagem de desenvolvimento baseada em modelos, integrando visões conceituais, instrucionais e de interação.

Tais resultados inserem-se na perspectiva do grupo de pesquisa para o estabelecimento de centros de ensino e treinamento em software livre e em sistemas embarcados críticos, no escopo do Centro de Competência em Software Livre (CCSL) de São Carlos, fruto do projeto de pesquisa *Quality Platform for Open Source Software* (QualiPSo), e do Instituto Nacional de Ciências e Tecnologia em Sistemas Embarcados Críticos (INCT-SEC), respectivamente.

Método

O desenvolvimento de objetos de aprendizagem é uma tarefa multidisciplinar e interdisciplinar. A multidisciplinaridade é caracterizada pelo uso coordenado de produtos de coleções de métodos de áreas distintas de conhecimento (BLEVIS; STOLTERMAN, 2009). Isto é patente, apenas para citar um exemplo, ao observar a integração entre currículo (que define os objetivos de aprendizagem que devem ser atingidos pelos aprendizes) e a instrução (que define e utiliza objetos de aprendizagem que satisfazem os objetivos provenientes do currículo). A interdisciplinaridade é uma abordagem para um problema em particular que integra os resultados da combinação de métodos de diversas áreas de conhecimento, de forma transcendental, conforme necessário para alcançar um determinado objetivo maior (BLEVIS; STOLTERMAN, 2009). No caso desta tese, os métodos da área de Engenharia de Software são integrados aos métodos de Projeto Instrucional para a definição de uma abordagem para o desenvolvimento de objetos de aprendizagem multimídia e interativos.

Esta pesquisa foi conduzida com base no método de definição de novas tecnologias de software proposto por Mafra *et al.* (2006), esse por sua vez uma extensão dos trabalhos de Shull *et al.* (2001). O objetivo do método é permitir o desenvolvimento e a maturação das tecnologias, desde a concepção acadêmica até a sua introdução na indústria. Conforme ilustrado na Figura 1.1, o método organiza-se em duas etapas: a definição inicial da tecnologia (MAFRA *et al.*, 2006) e o refinamento da tecnologia (SHULL *et al.*, 2001).

A primeira etapa consiste na identificação de evidências em estudos primários a partir da realização de uma revisão sistemática. As evidências obtidas são então utilizadas para a proposta de uma nova tecnologia, o que minimiza as dificuldades e incertezas do processo de sua definição. Prossegue-se, então, com a segunda etapa, de coleta de evidências sobre a aplicação da técnica-alvo de pesquisa e a identificação de seus pontos fortes e fracos, a partir dos quais se podem propor alterações na tecnologia (SHULL *et al.*, 2001).

Organização do texto

O restante desta tese organiza-se da seguinte forma. O Capítulo 2 apresenta os conceitos das perspectivas educacionais e computacionais sobre objetos de aprendizagem. Ele se divide

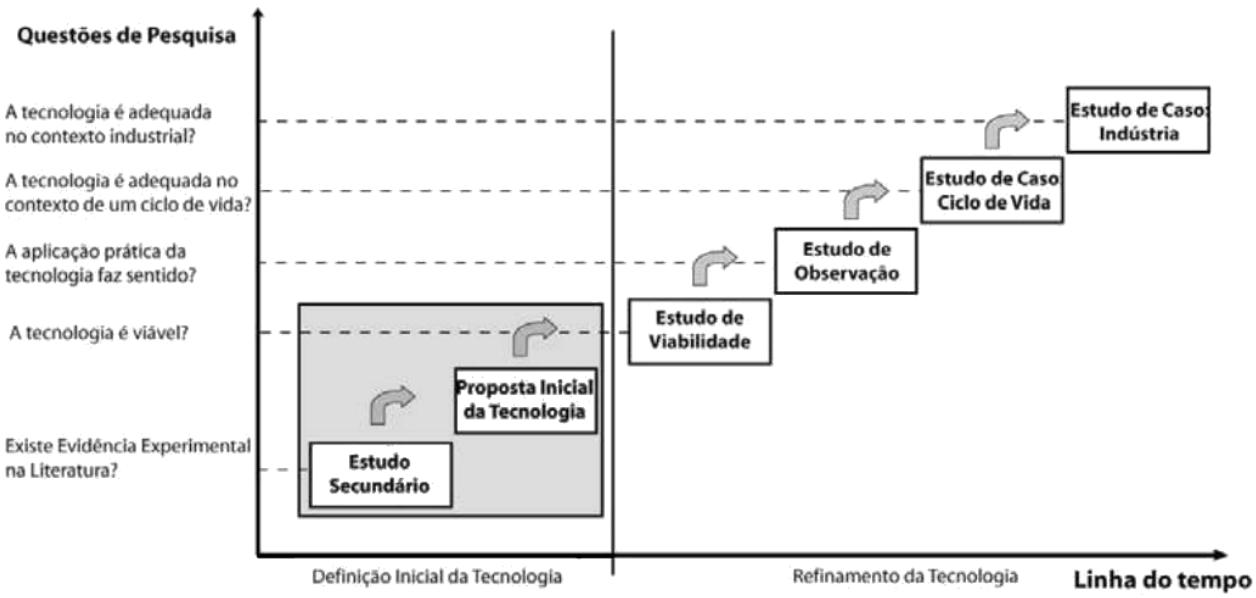


Figura 1.1: Método baseado em evidências para a definição de novas tecnologias de software (MAFRA *et al.*, 2006, p. 245).

em duas grandes seções. A primeira (Seção 2.1) trata de conceitos básicos de educação, processo e projeto instrucional. Na Seção 2.2, discorre-se sobre a definição, propriedades, representação e abordagens técnicas de desenvolvimento: métodos e processos.

A interatividade de objetos de aprendizagem e a sua utilização no contexto de televisão digital são exploradas no Capítulo 3. As características comuns a todos os sistemas e aquelas particulares ao Sistema Brasileiro de Televisão Digital (SBTVD) são apresentadas, bem como a interatividade da televisão digital e as características de objetos de aprendizagem para televisão digital.

A partir dos elementos descritos nos Capítulos 2 e 3, definem-se no Capítulo 4 os requisitos associados a objetos de aprendizagem multimídias e interativos e seu desenvolvimento. Apresentam-se os requisitos da perspectiva educacional (Seção 4.1), computacional (Seção 4.2) e de interação (Seções 4.3 e 4.4). A interatividade é investigada a partir dos elementos identificados no Capítulo 3 e das evidências coletadas em uma revisão sistemática sobre interatividade em objetos de aprendizagem.

A partir dos requisitos estabelecidos no Capítulo 4, define-se uma abordagem para o desenvolvimento de objetos de aprendizagem interativos no Capítulo 5. A abordagem estabelecida nesta tese, denominada LOD (*Learning Object Development*), compreende um método dirigido a modelos, um processo-padrão e um processo específico ao desenvolvimento de objetos e aprendizagem para televisão digital.

O emprego da abordagem é preliminarmente avaliado no Capítulo 6 em dois estudos no contexto de educação em teste de software. Os resultados desta tese são sumarizados no Capítulo 7, apontando-se as principais contribuições, limitações e trabalhos futuros.

Convenções adotadas no texto

As seguintes convenções são adotadas no texto desta tese. Palavras em letras mono-espaçadas representam identificadores ou palavras reservadas de uma linguagem de modelagem ou de programação. Palavras em idioma estrangeiro (inglês), que não sejam identificadores ou palavras reservadas, estão em itálico. Os acrônimos são expandidos em seu primeiro uso; a lista de acrônimos contém, como esperado, a expansão de todos aqueles utilizados no texto. Alguns termos em português estão acompanhados de sua tradução em inglês (principalmente quando o termo em inglês é bem conhecido na literatura). Modelos e listagens de código originais deste trabalho estão em inglês, dado que, no modelo de desenvolvimento aberto adotado para a abordagem LOD, todos os artefatos são descritos em inglês para facilitar o acesso.

Capítulo 2

Objetos de aprendizagem: caracterização dos aspectos educacionais e computacionais

Um objeto de aprendizagem corresponde a qualquer entidade, digital ou não, que possa ser usada, reutilizada ou referenciada pelas tecnologias que apoiem o aprendizado (IEEE, 2002). Baseado nessa definição, Wiley (2000, p. 7) propõe uma menos abrangente, e bastante adotada pela comunidade científica, na qual um objeto de aprendizagem consiste em qualquer recurso digital que possa ser reutilizado como apoio à aprendizagem. Esta definição também é utilizada neste trabalho.

O princípio central dos objetos de aprendizagem é permitir que o material educacional seja fracionado e suas partes combinadas e reutilizadas em diferentes contextos e cenários de aprendizagem, preferencialmente de forma automática, de acordo com um projeto instrucional (WILEY, 2000, p. 11). Nesse sentido, como características associadas destacam-se a reusabilidade, a interoperabilidade, a acessibilidade e a extensibilidade dos objetos desenvolvidos (DOWNES, 2001).

Assim, um objeto de aprendizagem agrega aspectos educacionais e computacionais. No âmbito educacional, ele é aplicado em atividades de aprendizagem planejadas para o desenvolvimento de conhecimentos e habilidades e executadas com auxílio de ferramentas computacionais em um determinado contexto e ambiente. Na perspectiva computacional, aplicam-se princípios de computação, em especial os da área de Engenharia de Software, para viabilizar o desenvolvimento e a utilização dos objetos de aprendizagem.

Este capítulo descreve ambas as perspectivas. Na Seção 2.1, trata-se da relativa à educação. Além dos conceitos básicos, apresentam-se as principais teorias de aprendizagem (Seção 2.1.1), as características do processo educacional quanto à instrução (Seção 2.1.2) e as iniciativas para sistematizar o desenvolvimento de atividades educacionais na forma de projetos instrucionais (Seção 2.1.3). A perspectiva computacional é abordada a partir da Seção 2.2, definindo-se as propriedades observáveis de um objeto de aprendizagem na Seção 2.2.1 e, a partir dessas, taxonomias de objetos de aprendizagem, conforme sumarizadas na Seção 2.2.2. Na Seção 2.2.3 discutem-se as formas com que os objetos de aprendizagem, em seus diferentes tipos, são representados. Considerando o tipo de objeto de aprendizagem, abordagens de desenvolvimento, embasadas na Engenharia de Software, são apresentadas na Seção 2.3, abrangendo métodos e processos.

2.1. Conceitos de educação e o papel dos objetos de aprendizagem

Educação é formar uma pessoa para o pleno exercício de seu papel na sociedade a que pertence, em um constante ciclo de aprendizado e ensino. Uma das principais formas de educação ocorre a partir da interação entre alunos e professores em um ambiente (espaço) e momento (tempo) comuns. Porém, ela apresenta uma importante limitação quanto à quantidade de informação e o alcance da comunicação e das interações. Avanços tecnológicos, desde os livros até as nuvens computacionais, permitiram transpor essas barreiras, ampliando o acesso à educação.

Não foi apenas a forma de aprender que sofreu alterações ao longo da história: as demandas impostas pela sociedade também acompanharam essa evolução, exigindo o domínio de conhecimentos e habilidades cada vez maiores, em espaços de tempos curtos. Assim, essa vivência em um mundo globalizado, alavancada pelo uso intenso de tecnologias de informação e comunicação, passou a exigir um indivíduo de competências múltiplas, que trabalhe em equipe, seja capaz de aprender e se adaptar a situações novas. Em decorrência disso, a formação precisa enfatizar a aquisição de habilidades de aprendizagem e interdisciplinaridade, sem negligenciar o desenvolvimento do espírito científico e das competências de pesquisa (BELLONI, 2006, p. 5).

As tecnologias são importantes instrumentos para a educação, mas não são suficientes. A utilização dessas ferramentas dá-se eficazmente apenas com a compreensão do ser humano e, mais precisamente, de como aprendemos. Teorias educacionais refletem o grau de entendimento que temos sobre os processos cognitivos e afetivos, sofrendo e causando reflexos nas atividades de aprendizagem. Na subseção a seguir, serão tratadas as principais teorias educacionais e os princípios que elas definem quanto ao processo de aprendizagem.

2.1.1. Teorias educacionais

A educação pode ser desenvolvida de acordo com uma teoria educacional, a qual reúne determinado número de leis e princípios pertinentes à filosofia, pedagogia, psicologia educacional e epistemologia. A partir dessas, as principais teorias educacionais desenvolvidas são o comportamentalismo, o cognitivismo e o construtivismo.

O comportamentalismo considera a aprendizagem como a modificação relativamente permanente do comportamento a partir de uma experiência (CATANIA, 1999). O comportamento é um reflexo visível do estado de uma pessoa, portanto passível de medição. O principal ponto de preocupação é a função que define esse estado, definida a partir das consequências de relações particulares entre os eventos ambientais e ações específicas. Tais eventos e ações são representados por estímulos (propriedades do ambiente) e respostas (propriedades do comportamento).

A "máquina de ensinar" de Skinner (1954) é um exemplo de aplicação desta teoria. O aluno interage com uma máquina, a qual fornece uma questão a ser respondida. Logo após fornecer a resposta, a máquina mostra a solução da questão, permitindo a imediata comparação da resposta fornecida e a esperada. O primeiro estímulo, a questão, tem uma resposta por parte do aluno, que serve de estímulo à máquina, que provê a resposta correta. A sequência de perguntas formuladas é projetada de modo a promover o gradual aprendizado do aluno, que, por sua vez, realiza-a no seu ritmo. Essa foi uma das primeiras formas de instrução programada (*programmed instruction*) disponibilizada e, nota-se, realizada sem auxílio computacional. O posterior advento do computador permitiu a criação de dispositivos aperfeiçoados, com um nível de personalização de instrução maior do que aquele possível na época de Skinner.

Enquanto o comportamentalismo parte dos aspectos mensuráveis do comportamento humano, o cognitivismo discute os processos não observáveis do comportamento, o que envolve a estrutura do conhecimento na mente humana. O cognitivismo reconhece a importância de elementos do comportamentalismo, tal como o processo de reforço e repetição resultantes das operações – estímulo e reforço – realizadas. Porém, ao invés de considerar o aprendiz como uma caixa-preta, observando apenas o comportamento visível, busca a compreensão de como as estruturas cognitivas do aprendiz são alteradas. Assim, foca-se no entendimento dos processos de aquisição e reorganização das informações.

Em busca de um modelo de compreensão da mente humana, estabelece-se um paralelo com um sistema de processamento de informação (computador). As informações são organizadas em esquemas, os quais podem ser criados, alterados, combinados e estendidos. O sujeito obtém as informações por intermédio de seu sistema sensorial, do qual são transferidas para a memória de curto prazo e, posteriormente, para a memória de longo prazo. As informações podem ser recuperadas de qualquer uma dessas memórias e o aprendizado é efetivado quando

as informações encontram-se na memória de longo prazo. A retenção de uma informação depende (1) das relações dela com informações anteriores e (2) da ativação em contextos diferentes.

Embora o cognitivismo considere não apenas as manifestações externalizadas do sujeito, mas também o aspecto cognitivo, ele assume um caráter simplista de processamento de informações. A visão construtivista, por sua vez, é a de que o sujeito constrói seu conhecimento e desenvolve suas habilidades a partir de suas percepções quanto às experiências anteriores, suas estruturas mentais e credos (JONASSEN, 1991). Em outras palavras, ao invés dele apenas armazenar as informações de forma estruturada, o sujeito reinterpreta-as de acordo com seu estado e contexto em que se insere, o que torna sua aprendizagem única. Ao mesmo tempo, ela possui uma parte significativa em comum com aquela de outros indivíduos, compartilhada por um processo de negociação social (JONASSEN, 1994).

2.1.2. Características do processo instrucional

O processo educacional é estabelecido por um ciclo de instrução (ensino) e aprendizagem. Na seção anterior, foram apresentadas teorias educacionais que estabelecem princípios pelos quais o aprendizado acontece. Nesta seção, a instrução, ou seja, o conjunto de atividades feitas propositalmente para facilitar a aprendizagem (REIGELUTH; CARR-CHELLMAN, 2009, p. 6), é caracterizada. Afinal, um objeto de aprendizagem é um instrumento utilizado para guiar o aprendiz em sua meta de aprender.

O processo instrucional, definido a partir do conjunto de princípios de uma ou mais teorias educacionais, possui algumas facetas que devem ser consideradas: a formalidade (formal e informal), a modalidade (presencial ou a distância), o acesso (fechado ou aberto), as interações e as avaliações do aprendizado.

2.1.2.1. Formalidade

Educação é um termo amplo, que reflete toda experiência de aprendizado de uma pessoa. Isto inclui experiências pessoais, conhecimentos passados pela família, habilidades adquiridas no trabalho, conhecimentos aprendidos na escola. Essas diferentes formas de educação podem ser classificadas em formal e informal (GASPAR, 2002).

A educação formal é realizada de forma sistemática e em um ambiente específico para esse fim. Geralmente é normatizada pelos países, definindo-se níveis e requisitos para sua execução. Cada nível direciona-se, geralmente, a uma faixa etária. Em complemento a esse cenário regrado, a educação informal acontece ao longo da vida, seja em situações usuais do cotidiano ou para uma finalidade específica (SMITH, 2011). Ela não tem um local ou tempo específico, sendo oferecida de forma flexível aos interessados.

2.1.2.2. Modalidade

A educação dá-se de três formas: presencial, a distância e híbrida (*blended learning*). A modalidade presencial é a mais conhecida e praticada. Nela, professores e alunos estão presentes no mesmo espaço e tempo para a realização do ensino. Antagônica à presencial, a educação a distância (EaD) é aquela em que os atores – professores e alunos – encontram-se, predominantemente, separados uns dos outros (MORAN, 2002). Enfoca-se na responsabilidade do aluno em aprender, de forma predominantemente autônoma, com intermediação e colaboração do professor remotamente, por meios de comunicação e ferramentas computacionais.

Na modalidade híbrida (*blended learning*), ocorre uma combinação de educação presencial e a distância. Por exemplo, aulas e atividades práticas podem ser realizadas remotamente enquanto que as avaliações podem ocorrer de forma presencial.

2.1.2.3. Acesso

Enquanto presencial, a educação é realizada com grupos de pessoas, tipicamente em uma sala de aula. Os objetivos educacionais de cada grupo de aprendizes são equivalentes, mesmo que cada indivíduo tenha, para si, objetivos distintos daqueles do grupo. Além disso, o espaço, o tempo e o ritmo do aprendizado, em média, são similares. Essa forma de educação, inflexível, é aquela que denominamos de educação fechada.

A educação aberta é aquela que não apresenta barreiras aos aprendizes (ROMISZOWSKI; ROMISZOWSKI, 1998), ou, mais precisamente, que as restrições impostas aos aprendizes estão em constante revisão e removidas sempre que possível. Trata-se de uma educação personalizada, em geral independente e individualizada, que satisfaz os requisitos do indivíduo quanto a sua necessidade de conhecimento, disponibilidade de tempo e não-comprometimento de suas atividades profissionais (ROWNTREE, 1992, p. 15). Rowntree (1992) define a abertura da educação a partir de três fatores:

- Quem: Diz respeito dos requisitos quanto ao aprendiz, ou seja, o quanto se exigirá dele durante o processo educacional e quais as habilidades esperadas para que o processo educacional transcorra. Algumas perguntas que caracterizam esse fator: Quão fácil é para alguém tornar-se um aprendiz, sem restrições de idade, qualificação, saúde ou trabalho? Quanto o sucesso do aprendizado depende do aprendiz possuir as habilidades de aprendizado, motivação e cultura que se espera dos usuários típicos do sistema?
- O que: Trata do conteúdo ou habilidades a serem aprendidos. Quão livre o aprendiz é para decidir o conteúdo e como ele será avaliado?
- Como: Modo com que o aprendizado ocorrerá. Este fator compreende a liberdade do aprendiz para a escolha do momento (quando?), local (onde?) e o ritmo em que a

atividade de aprendizagem será realizada. Além disso, a questão também engloba os métodos de ensino e aprendizagem a serem utilizados e as formas de interação com outros aprendizes e com os instrutores.

A educação aberta não implica em educação a distância, mas, observando os fatores discutidos, percebe-se que a abertura, utilizando-se dos mecanismos da educação a distância, é possível e facilitada. As ferramentas podem ser acessadas de qualquer local e a comunicação pode ser síncrona ou assíncrona, o que permite a execução de cursos abertos. Todavia, cabe ressaltar que a educação a distância não é sinônimo de educação aberta. Por exemplo, pode-se restringir o oferecimento de conteúdo no formato de vídeos em tempo real (conferências), com avaliações em datas previamente estabelecidas e em local físico.

2.1.2.4. Formas de interação

A instrução pressupõe, como condição mínima, a realização de atividades por partes do aprendiz (*e.g.*, a leitura de um livro). Essa atividade pode ser realizada de forma autônoma, mas o fato é que o aprendizado pressupõe alguma forma de interação: com o conteúdo, com professores, com outros alunos (MOORE, 1989) e com a comunidade (e a realidade social) a qual o aluno pertence

Existem várias formas de caracterizar interação, conforme será discutido e caracterizado nos Capítulos 3 e 4. Apenas para ilustrar os principais conceitos, uma interação pode (1) não existir e, consequentemente, não causar o aprendizado; (2) ser restrita a um material inerte como um livro; (3) requerer a interação com um instrutor, tal como em uma aula narrativa tradicional; (4) ser realizada com outros alunos em um grupo de discussão moderado pelo professor; (5) ocorrer com a comunidade durante o aprendizagem pela resolução de um problema identificado no meio em que o aluno vive.

Além dessa dimensão relacionada aos atores envolvidos na instrução, deve-se também considerar o valor da interação em prol dos objetivos de aprendizagem estabelecidos. Uma interação pode ser inócua ao não contribuir para o desenvolvimento pessoal, simples ao contribuir de forma individual para os objetivos de aprendizado de cada indivíduo, ou rica pela colaboração entre as pessoas e a constatação de progresso e até superação dos objetivos de todos os aprendizes.

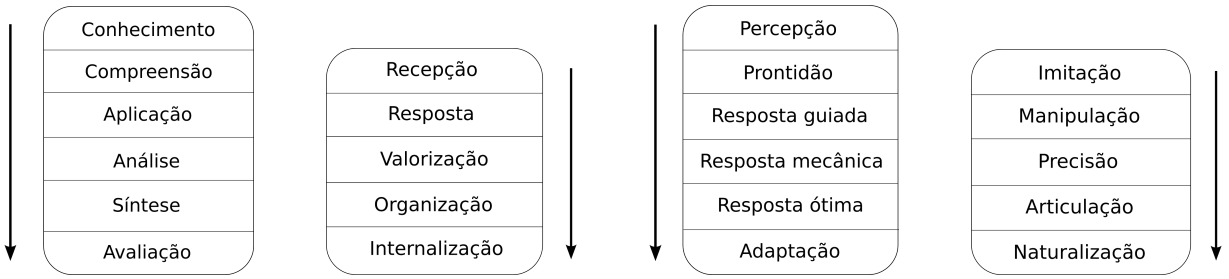
2.1.2.5. Formas de avaliação

A avaliação é um mecanismo para aferir o aprendizado do aluno por parte da intervenção do ensino e, a partir da compreensão de sua evolução, manter ou alterar os mecanismos utilizados de modo a permitir que cada aluno alcance os objetivos de aprendizagem da

atividade educacional. Exemplos clássicos de avaliação são provas, exercícios, projetos de final de curso, estudos de caso, grupos de foco, dentre outros.

Portanto, ao tratar de avaliação, é necessário compreender também o significado dos objetivos educacionais. Um objetivo educacional é o resultado esperado pelo aluno após o aprendizado. De forma simples, poder-se-ia afirmar que a aquisição de um conhecimento específico sobre um determinado assunto é um possível objetivo. No entanto, como descobrir que ele foi satisfeito? Torna-se necessário caracterizar adequadamente os objetivos, classificando-os e especializando-os. Considerando-se os domínios psicológicos, pode-se adotar uma classificação com base nos domínios cognitivos, afetivos e psicomotores.

A taxonomia de Bloom atua no domínio cognitivo e define o valor do conhecimento e habilidades intelectuais com base no comportamento esperado do aluno (BLOOM *et al.*, 1956). Ela define uma hierarquia de seis classes – conhecimento, compreensão, aplicação, análise, síntese e avaliação –, assumindo desde um valor baixo (habilidade cognitiva de baixa ordem, o simples saber) a alto (capacidade de avaliação crítica), conforme exposto na Figura 2.1a. Em seu nível inferior, tem-se o conhecimento por conhecimento: domina-se o significado do conceito, mas apenas isto. Um nível acima, compreende-se o conceito em relação aos demais conceitos. Em aplicação, o conhecimento é utilizado na resolução de uma situação real. A análise compreende a capacidade de entender as partes que compõem o conhecimento e as causas e efeitos daquele conhecimento aplicado, criando ligações com outros conceitos e reorganizando seu mapa mental. A síntese abrange o entendimento global do conhecimento e suas consequências, permitindo a generalização ou especialização desse e a produção de novos conhecimentos. Finalmente, em avaliação aborda-se a crítica e avaliação sobre a real função e validade do conhecimento, julgando-o com base em valores e opiniões pessoais. Cabe destacar que, para cada uma dessas seis classes, os objetivos da classe superior faz uso dos comportamentos encontrados nas classes inferiores (BLOOM *et al.*, 1956, p. 18).



(a) Taxonomia do domínio cognitivo segundo Bloom *et al.* (1956).

(b) Taxonomia do domínio afetivo segundo Krathwohl *et al.* (1964).

(c) Taxonomia do domínio psicomotor segundo Simpson (1966).

(d) Taxonomia do domínio psicomotor segundo Dave (1968).

Figura 2.1: Taxonomias de objetivos educacionais para os domínios cognitivo, afetivo e psicomotor.

Em continuação à taxonomia de Bloom, acrescentou-se a taxonomia de Krathwohl quanto

ao domínio afetivo, descrevendo os objetivos de aprendizagem que enfatizam o tom sentimental, a emoção ou o grau de aceitação e rejeição. Tais objetivos são expressos na forma de interesses, atitudes, apreciações, valores e tendências (KRATHWOHL *et al.*, 1964). Conforme exposto na Figura 2.1b. cinco níveis são definidos para esse domínio: recebimento/recepção, resposta, valorização, organização, internalização. O primeiro, recepção, descreve a consciência e a sensibilidade quanto à existência de certas ideias, materiais e fenômenos. A resposta corresponde às atitudes que se tem face ao recebimento, compreendendo a participação do estudante. Àquilo que se responde, atribui-se um determinado valor, seja pela proficiência, suporte ou debate. A harmonização dos valores com princípios pessoais confere a classificação de organização. Finalmente, em internalização, os valores harmonizados são usados para agir de acordo com valores anteriormente estabelecidos.

O domínio psicomotor inclui os movimentos físicos, coordenação e uso de habilidades motoras. Considerando, por exemplo, a taxonomia de Simpson (1966), apresentada na Figura 2.1c, as categorias são: percepção, prontidão, resposta guiada, resposta mecânica, resposta ótima e adaptação. Em percepção, órgãos sensoriais são utilizados para perceber objetos, quantidades e relações. A prontidão caracteriza o preparo mental do aluno para agir. A resposta guiada é um dos primeiros estágios de fato de aprendizado, em que se imita um terceiro. Após muita prática, podem-se repetir as ações psicomotoras de forma autônoma. O aperfeiçoamento dessa resposta leva a uma resposta clara e ótima (SIMPSON, 1966, p. 25–30). Um modelo alternativo é o de Dave (1968), que estabelece cinco categorias: imitação, manipulação, precisão, articulação e naturalização, conforme disposto na Figura 2.1d.

Estabelecidos os objetivos educacionais e as várias facetas com que eles podem (e devem) ser analisados, procede-se aos tipos de avaliação que podem ser empregados para aferir o quanto cada aluno atingiu de acordo com os objetivos propostos. A literatura define três tipos de avaliação: diagnóstica, formativa e sumativa. A avaliação diagnóstica é realizada no início de um ciclo de atividades de aprendizagem para verificar o nível inicial dos aprendizes quanto aos objetivos educacionais. A avaliação formativa é conduzida durante as atividades de aprendizagem para verificar o progresso do aprendiz, permitindo o diagnóstico de dificuldades de aprendizagem e de ensino. Ao final de um ciclo de atividades de aprendizagem, realiza-se a avaliação sumativa para aferir se os objetivos educacionais foram satisfeitos.

Finalmente, os objetivos de aprendizagem, em suas distintas classificações, e os tipos de avaliação devem ser alinhados às teorias educacionais. Por exemplo, na visão comportamentalista, definições simples de requisitos de aprendizagem e avaliações objetivas podem ser suficientes para a avaliação de um aprendiz. No entanto, considerando o cognitivismo e o construtivismo, não faria sentido definir um objetivo sem relacioná-lo com outros. Na verdade, cada conceito ou habilidade-fim deve estar associado com conceitos e habilidades pré-existentes e relacionados entre si, formando um todo (AEBIL, 1982, p. 261-282). Por esses princípios, um mapa conceitual (NOVAK; GOWIN, 1984) é uma opção que

permite avaliar não apenas os conhecimentos e habilidades aprendidos, mas também a forma com que os aprendizes os desenvolveram por intermédio das interações com outros alunos e professores.

2.1.3. Características de projeto instrucional

Um método instrucional define a melhor maneira para realizar a instrução e a alavancar o aprendizado (NEWBY *et al.*, 1996; REIGELUTH, 1983). Pelas características apresentadas nas seções anteriores, é evidente que definir e executar um método instrucional não é uma tarefa trivial. Um passo intermediário é, portanto, desejável para a realização da instrução, que inclui a escolha e combinação de métodos adequados para alcançar os objetivos de aprendizagem levando em consideração os princípios de uma teoria educacional. O projeto instrucional reúne esses dispositivos, permitindo a sistematização do projeto do processo instrucional ao abstrair componentes do processo educacional e definir como esses componentes comunicam-se entre si para alcançar um objetivo (no caso, os objetivos educacionais).

O projeto instrucional (*instructional design*) é um processo para o desenvolvimento sistemático de especificações instrucionais e que utiliza teorias de aprendizagem e de instrução para garantir a qualidade de instrução. Ele compreende a análise das necessidades e objetivos dos aprendizes, o desenvolvimento de materiais didáticos e sistemas associados, a implantação e execução das atividades instrucionais, e a avaliação da instrução e das atividades dos aprendizes.

O modelo *Analysis, Design, Development, Implementation, and Evaluation* (ADDIE) foi um dos primeiros modelos de projeto instrucional desenvolvidos e serve de referência para os demais modelos. Ele descreve cinco grupo de processos: análise, projeto, desenvolvimento, implementação e avaliação da instrução (BRANSON *et al.*, 1975). No grupo de análise, definem-se atividades para a determinação dos objetivos educacionais e dos conhecimentos e habilidades dos aprendizes. Em projeto, ocorre o detalhamento dos objetivos educacionais, definição dos instrumentos de avaliação e material instrucional. A criação ou seleção do material didático realiza-se pelas atividades de desenvolvimento. A implementação trata da instrução propriamente dita, com os aprendizes. A efetividade da instrução é verificada por atividades de avaliação.

Os principais componentes do ADDIE definem uma sequência de execução, realizada iterativamente, conforme exposto na Figura 2.2. A cada ciclo, ocorre uma revisão da instrução com base nos resultados obtidos na etapa de avaliação.

A partir do modelo de referência ADDIE, definiram-se vários outros modelos de projeto instrucional: modelo de Dick e Carey (DICK *et al.*, 2005), modelo de Kemp (MORRISON *et al.*, 2010), 3PD (SIMS; JONES, 2003), dentre outros. Essa diversidade explica-se pelo fato de cada modelo assumir uma ou mais teorias educacionais, características de formalidade,

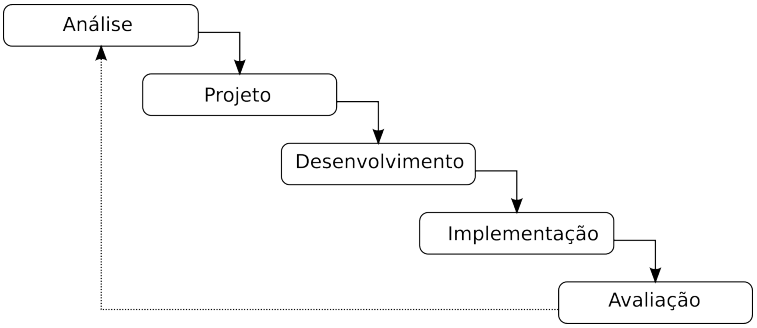


Figura 2.2: Modelo do projeto instrucional segundo o ADDIE.

modalidade, acesso e avaliação, e incorporar diferentes experiências práticas. Apesar dessas diferenças, a essência dos modelos é a mesma apresentada pelo ADDIE.

Um modelo proeminente de projeto instrucional é o de Dick e Carey (2005): o *Instrucional System Design* (ISD). Ele possui características de várias teorias pedagógicas (comportamentalismo, cognitivismo e construtivismo) e agrega a experiência daqueles autores em anos de aplicação do modelo. O ISD divide-se em dez etapas, executadas em sequência, tal como o ADDIE, porém com atividades para corrigir as etapas anteriores de acordo com os resultados da avaliação, configurando um ciclo de vida iterativo, conforme indicado na Figura 2.3.

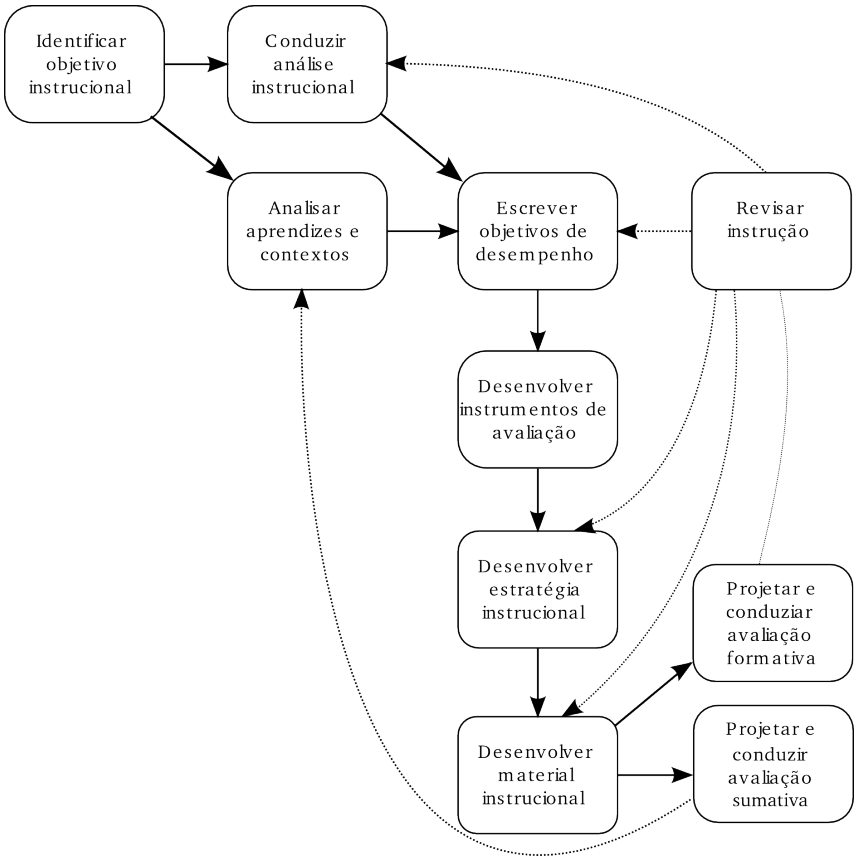


Figura 2.3: Etapas do projeto instrucional do modelo de Dick e Carey (DICK *et al.*, 2005).

- Identificar objetivos instrucionais: Determinar as habilidades que o instrutor deseja que os aprendizes adquiram após o término da instrução.
- Conduzir análise instrucional: Determinar o que as pessoas fazem ou demonstram quando desempenham o objetivo instrucional. Esta etapa também determina habilidades, conhecimentos e atitudes que são requeridos dos aprendizes para o início da instrução (pré-requisitos).
- Analisar aprendizes e contextos: Análise do contexto em que o aprendiz aprenderá e utilizará as habilidades. Nesta etapa, determinam-se: as habilidades, preferências e atitudes atuais dos aprendizes; as características de configuração instrucional; e as condições nas quais as habilidades serão utilizadas.
- Escrever objetivos de desempenho: A partir da análise instrucional e da análise do aprendiz e seu contexto, especifica-se o que exatamente o aprendiz, após completar a instrução, será capaz de fazer, sob que condições e os critérios para avaliar o desempenho correto da habilidade.
- Desenvolver instrumentos de avaliação: Definir as avaliações para medir a habilidade dos aprendizes quanto aos objetivos instrucionais estabelecidos.
- Desenvolver estratégia instrucional: Identificar a estratégia instrucional a ser utilizada para que os alunos alcancem os objetivos previamente estabelecidos.
- Desenvolver e selecionar material instrucional: Estabelecida a estratégia instrucional, desenvolver o material instrucional e as avaliações a serem utilizadas.
- Projetar e conduzir avaliação formativa: Coletar dados da instrução para identificar pontos de melhoria.
- Projetar e conduzir avaliação sumativa: Passo final do ciclo do processo instrucional (do ponto de vista do aprendiz). Avalia-se, de forma absoluta, o valor da instrução.
- Revisar instrução: Revisar a instrução a partir da interpretação dos dados obtidos da avaliação formativa. Executada a cada ciclo de instrução, ela identifica deficiências encontradas pelos aprendizes na realização dos objetivos instrucionais e as relaciona com erros na instrução. A revisão da instrução é utilizada para validar e corrigir: a análise instrucional e as suposições sobre as habilidades; suposições sobre atitudes e conhecimentos prévios dos estudantes; objetivos de desempenho; e o material instrucional.

Uma abordagem alternativa para o projeto instrucional é a Prototipação Rápida (*Rapid Prototyping*). O propósito do método é criar a estrutura lógica do produto final, sem os gastos de um ciclo completo de desenvolvimento (JONES *et al.*, 1992). As fases de projeto e desenvolvimento são executadas simultaneamente enquanto a de avaliação é feita durante todo o processo. Esse modelo oferece uma eficiente comunicação entre o projetista e os usuários devido ao retorno rápido das alterações realizadas no projeto. A união da completeza e consolidação do ISD ao desenvolvimento ágil e iterativo da prototipação rápida

são características a serem observadas e conciliadas com o desenvolvimento de objetos de aprendizagem.

2.1.4. Educação com informática

O desenvolvimento de teorias educacionais fez-se em período anterior ao desenvolvimento e disseminação dos computadores. Não obstante, viu-se nos computadores uma importante ferramenta para aplicá-las e, usufruindo-se das capacidades de armazenamento e processamento disponíveis, otimizar o processo de instrução. As primeiras iniciativas de educação com informática focavam em sistemas tutoriais e exercícios repetitivos (ATKINSON; WILSON, 1968), de forma similar à instrução programada de Skinner (1954). Essas primeiras iniciativas possuem um embasamento comportamentalista. No campo construtivista, um dos pioneiros no uso de informática na educação foi Seymour Papert, que criou a linguagem LOGO (PAPERT, 2004) em colaboração com Jean Piaget, um dos pioneiros do construtivismo.

Atualmente observa-se a predominância dos ambientes virtuais de aprendizagem, os quais prestam o papel de repositório de material didático, provedor de aplicações educacionais (fóruns, *wikis*, diários (*blogs*), portfólios) e meio para coordenar a execução de atividades didáticas. Menos utilizados na prática, mas alvo de pesquisas, os dispositivos móveis (celulares e *tablets*) e televisores, aliando educação com interatividade e entretenimento, são tendências para a educação com apoio computacional.

Objetos de aprendizagem constituem um importante elemento da educação com informática. Eles encapsulam o material didático em uma forma apropriada para utilização em ambientes virtuais de aprendizagem e outras aplicações educacionais, possibilitando inclusive o alinhamento com projetos instrucionais. Na próxima seção, aborda-se a perspectiva computacional dos objetos de aprendizagem.

2.2. Objetos de aprendizagem

A escolha, o desenvolvimento e a utilização de objetos de aprendizagem estão relacionados com todas as etapas do projeto instrucional. A partir dos objetivos instrucionais e do diagnóstico dos aprendizes, ambos provenientes da etapa de análise, é possível escolher objetos adequados para o contexto das atividades de aprendizagem. Caso não seja possível selecionar um objeto, o projeto das atividades constitui um dos elementos a serem considerados na construção de objetos de aprendizagem, assegurando-se, durante a etapa de desenvolvimento, o alinhamento entre projeto instrucional e objetos de aprendizagem. A implementação do projeto instrucional constitui-se na utilização dos objetos selecionados ou construídos e, possivelmente, adaptar-se os objetos ou as sequências de utilização deles em função do progresso do aprendiz quanto aos objetivos associados às atividades. Para a etapa de avaliação, objetos de aprendizagem

podem ser encarregados do oferecimento e correção de exercícios e projetos, repassando para o ambiente de aprendizagem os resultados de avaliações diagnósticas, formativas e sumativas.

A devida integração de objetos de aprendizagem, fomentando-se o reúso em diversas atividades, realiza-se pela identificação de suas propriedades, estabelecimentos de formas de representação e, principalmente, de abordagens para o desenvolvimento: métodos e processos. Tais tópicos são tratados nas próximas subseções.

2.2.1. Propriedades dos objetos de aprendizagem

A literatura relata diversos tipos de objetos de aprendizagem: objetos de aprendizagem reutilizáveis (RLO) (Cisco Systems, Inc., 2001) ou generativos (GLO) (DAMASEVICIUS; STUIKYS, 2008), unidades de aprendizagem (UoL), módulos educacionais (BARBOSA; MALDONADO, 2011b), componente instrucional (MERRILL, 2001), documentos pedagógicos (FORTE *et al.*, 1997), material de aprendizagem (California State University, 1997), objeto de conteúdo compartilhável (ADL, 2009), dentre outros. Cada tipo de objeto enfatiza um conjunto de propriedades que define sua reusabilidade.

A noção mais básica de reusabilidade de um objeto de aprendizagem reside na divisão e composição do objeto a partir de outros objetos, ou seja, a sua granularidade e organização. Um objeto, em sua forma mais simples, não é divisível. Um nível acima, encontram-se os objetos que agregam ou são compostos por objetos indivisíveis. O resultado da composição de objetos indivisíveis também é indivisível, ou seja, não é possível recuperar as partes do todo, enquanto que uma agregação permite a recuperação das partes. Caso seja possível acessar os objetos de uma agregação de forma independente, caracteriza-se uma agregação fraca ou simplesmente uma associação entre objetos. Finalmente, caso os objetos associados não sejam apenas acessados, mas também dinamicamente criados, modificados e removidos, caracteriza-se um objeto de aprendizagem generativo.

Na prática, observa-se que, quanto maior a granularidade, a tendência é a adoção de formas de organização mais complexas. Ao mesmo tempo, a reutilização de um objeto é mais difícil quanto maior for sua granularidade e mais complexa for sua organização: a primeira, por ser naturalmente difícil reutilizar quantidades elevadas de objetos; a segunda, porque a organização está associada a um contexto específico de aprendizagem, o que impede a reutilização de objetos de aprendizagem complexos em contextos distintos.

Além da granularidade e organização (estrutura), um objeto de aprendizagem também possui um comportamento. Objetos mais simples não apresentam qualquer comportamento, sendo passivos quanto ao contexto educacional em que estão inseridos ou a outros objetos de aprendizagem com que estão relacionados. Observando-se a estrutura, objetos indivisíveis, de composição ou agregação são geralmente passivos. Uma forma de comportamento branda é a definição da apresentação do conteúdo do objeto: a partir de estímulos externos, o objeto

reage com alterações em seu estado sem, no entanto, alterar a granularidade ou estrutura. Um objeto é considerado ativo quando possui o comportamento determinado pelo seu estado atual e de seus componentes, podendo adaptar suas reações em função do contexto. Objetos com granularidade fina e estruturados dinamicamente (objetos ativos) geralmente possuem um comportamento dinâmico. Cabe destacar que um objeto atômico também pode apresentar um comportamento: por exemplo, uma aplicação educacional utilizada como objeto de aprendizagem.

A estrutura e o comportamento de um objeto de aprendizagem são importantes interesses tratados por objetos de aprendizagem e, não coincidentemente, eles remetem ao paradigma de orientação a objeto de Engenharia de Software, em que os objetos possuem identidade, estado e comportamento. No entanto, tal como em Engenharia de Software, existem outros interesses acerca de objetos de aprendizagem: projeto instrucional, contexto (pré-requisitos educacionais, perfil dos aprendizes), proteção intelectual (e custos associados), dependências (aplicações e padrões), gerenciamento de configuração (versões e alterações sofridas ao longo do tempo), escopo ou abrangência da experiência educacional, dentre outros.

2.2.2. Taxonomia de objetos de aprendizagem

Um objeto de aprendizagem pode ser classificado conforme suas propriedades e a finalidade educacional. A taxonomia mais simples, o *Learnativity Content Model*, estabelece classes de acordo com a granularidade do objeto de aprendizagem, assumindo os princípios de que a reusabilidade é inversamente proporcional à granularidade do objeto e de que o contexto é proporcional à granularidade (WAGNER, 2002). Objetos de aprendizagem, de acordo com essa taxonomia, são classificados como:

- Recursos (*content asset*): Objetos de aprendizagem atômicos e simples (*e.g.*, texto, animações, áudio).
- Objeto de informação: Composição de recursos com uma finalidade educacional (conceito, princípio ou procedimento).
- Objeto de aprendizagem: Sequência de objetos de informação.
- Componente de aprendizagem: Sequência de objetos de aprendizagem.
- Ambiente de aprendizagem: Ambiente computacional composto por software educacional, serviços de infraestrutura e componentes de aprendizagem.

A taxonomia de Wiley (2000) combina as propriedades intrínsecas de um objeto de aprendizagem, descritas na Seção 2.2.1, e algumas propriedades educacionais, tais como a função do objeto e características instrucionais. Quanto à estrutura, considera-se a quantidade e o tipo de objetos de aprendizagem que constituem cada tipo de objeto. Quanto ao comportamento, define-se como o objeto é utilizado e qual a lógica (tipo de atividades)

definida em seu projeto instrucional. A taxonomia também considera a reusabilidade dos objetos da composição e em relação ao contexto em que o objeto de aprendizagem se insere:

- Fundamental: Tipo mais simples de objeto de aprendizagem. Trata-se de um objeto atômico utilizado apenas para exibição. Seu potencial para reúso é alto.
- Combinado-fechado: Objeto que combina alguns objetos de aprendizagem mais simples (porém, os objetos da composição não podem ser reutilizáveis em outros objetos). O projeto instrucional inexiste ou é simples (restrito a avaliações objetivas). Seu potencial para reúso é médio.
- Combinado-aberto: Objeto que combina vários objetos de aprendizagem (e objetos de qualquer tipo). Seu projeto instrucional é elaborado, descrevendo atividades de instrução e de avaliação para um domínio/contexto específico. Sua reusabilidade é média.
- Generativo-apresentação: Objeto que combina uma quantidade variada de objetos de aprendizagem do tipo fundamental ou combinado-fechado. Seu projeto instrucional é razoavelmente simples, sendo restrito à apresentação. Sua reusabilidade é alta.
- Generativo-instrucional: Um objeto de aprendizagem generativo-instrucional pode combinar qualquer quantidade de objetos do tipo fundamental, combinado-fechado e generativo-apresentação. Seu projeto instrucional é elaborado, descrevendo atividades de instrução e de avaliação de forma independente de contexto e domínio. Sua reusabilidade é alta.

Redeker (2003) adota uma estratégia distinta, focando na visão do aprendiz e na interatividade possibilitada pelas atividades definidas no objeto de aprendizagem:

- Unidade de conhecimento: Material didático sobre um conceito. Pode ser classificada em:
  - Receptiva: Unidade passiva.
  - Internamente interativa: Unidade com sequências de interação de até 5 minutos.
  - Cooperativa: Unidade com sequências de tempo ilimitado e com atividades interativas entre os aprendizes.
- Unidade de aprendizagem: Composição de unidades de conhecimento quanto a um tema (um ou mais conceitos).
- Curso parcial: Sequência ou rede de unidades de aprendizagem ou unidades de conhecimento que são um componente de um curso.
- Curso: Sequência ou rede de unidades de aprendizagem.

Conforme visto nos três exemplos de taxonomias aqui apresentados (*Learnativity*, de Wiley e de Redeker), a definição de uma taxonomia única para classificar objetos de aprendizagem não é uma atividade trivial: os interesses são distintos, porém relacionados. Por exemplo, é possível associar objetos mais complexos (tais como os generativos da taxonomia de Wiley) com os cooperativos de Redeker, mas nada impede um objeto simples (Wiley) de ser cooperativo. A diferença, nesse caso, reside no controle que existe no processo e o quanto

esse controle pertence ao professor ou ao ambiente virtual de aprendizagem e ao objeto de aprendizagem.

Um ponto central de objetos de aprendizagem, que os distinguem de materiais didáticos tradicionais, é a reusabilidade. Ela é promovida, em sua forma mais simples, pela descrição dos objetos com metadados e projetos instrucionais. No entanto, cada execução de uma atividade de aprendizagem e dos objetos a ela associada é única, requerendo a adaptação em função do contexto. Algumas variabilidades podem ser tratadas no âmbito da atividade, definindo, por exemplo, sequências alternativas para entrega de objetos de aprendizagem. Porém, eventualmente torna-se necessária a alteração de elementos do projeto instrucional e, consequentemente, dos objetos de aprendizagem, exigindo um esforço considerável.

Um passo importante para tratar essa questão é reconhecer os objetos de aprendizagem como um artefato de engenharia, os quais devem ser criados a partir de requisitos e modelos, empregando técnicas para garantir a qualidade do produto e a sistematização do processo. Dessa forma, o reúso não se restringe à escolha de objetos de aprendizagem por suas características finais, mas pelos artefatos de desenvolvimento, principalmente os modelos, que os geraram. Esses modelos também permitem reutilizar partes do objeto de aprendizagem, considerando que essas são devidamente projetadas e seus aspectos representados modularmente (MEYER *et al.*, 2011). Considerando essas características e observando as taxonomias anteriormente definidas, resume-se a classificação dos objetos de aprendizagem em:

- Objetos de aprendizagem simples ou recursos: O tipo mais simples de objeto de aprendizagem consiste de qualquer recurso passível de utilização em uma situação de aprendizagem. Tais objetos não possuem recursos intrínsecos ao reúso: não existem modelos ou quaisquer artefatos que permitam a seleção ou a adaptação. A reusabilidade estabelece-se pela destreza do usuário em empregá-lo.
- Objetos de aprendizagem com metadados: Os metadados permitem descrever as características do objeto de aprendizagem relativas ao contexto de aprendizagem, facilitando a sua recuperação e utilização em educação. A reusabilidade restringe-se à busca: não existem modelos ou quaisquer artefatos que auxiliem as tarefas de adaptação e engenharia.
- Unidades de ensino: Objeto de aprendizagem construído a partir de um projeto instrucional e que define inteiramente uma situação de aprendizagem. Os modelos do objeto de aprendizagem são reutilizados para a adaptação do objeto para outros contextos e requisitos.

Esta tese trata de objetos de aprendizagem do tipo unidades de ensino, fazendo uso de objetos de aprendizagem simples e com metadados para compor a unidade de ensino a ser disponibilizada em uma aplicação educacional (ambiente virtual de aprendizagem ou televisão digital). Doravante denominados apenas de "objetos de aprendizagem", abordagens para a

representação e o desenvolvimento desse tipo de objetos são descritas nas próximas seções.

2.2.3. Representação de objetos de aprendizagem

Computacionalmente, um objeto de aprendizagem deve expor suas propriedades em um formato automaticamente processável. Os elementos relacionados à estrutura (granularidade e composição) e ao comportamento constituem o conteúdo do objeto de aprendizagem, desenvolvido com um método específico (tal como aqueles discutidos na Seção 2.3). As propriedades relacionadas aos demais interesses de um objeto de aprendizagem são representadas por metadados, responsáveis por descrever as principais características do objeto no intuito de facilitar a sua recuperação e reúso. Dois padrões têm sido amplamente adotados para a definição dos metadados: o *Dublin Core* (DC) (DCMI, 2010) e o *Learning Object Metadata* (LOM) (IEEE, 2002):

- Dublin Core: Descreve mídias digitais como um todo e não define elementos específicos para educação. No entanto, sua simplicidade torna-o uma alternativa interessante para a descrição de objetos de aprendizagem, principalmente aqueles que são mídias simples. Ele define os seguintes elementos: título, assunto, descrição, tipo, origem, relação, cobertura, criador, editora, contribuidores, direitos, data, formato, identificador e idioma.
- LOM: Especifica um esquema hierárquico de dados que define a estrutura de metadados do objeto de aprendizagem. São definidas nove categorias: geral, ciclo de vida, meta-metadados, técnico, educacional, direitos, relação, anotação e classificação. Cada categoria, por sua vez, define um conjunto de elementos, que possuem um nome, descrição, cardinalidade, ordem, exemplo, tipos de dados e valores permitidos.

Além dos elementos, que constituem um modelo conceitual, são necessárias formas para a representação física dos metadados. Para o LOM, define-se um mapeamento para documentos *eXtensible Markup Language* (XML) (IEEE, 2005). No caso do Dublin Core, eles podem ser representados como documentos XML, metadados no cabeçalho de documentos *Hypertext Markup Language* (HTML) ou tuplas *Resource Description Framework* (RDF).

Recentemente, iniciou-se o desenvolvimento da família de padrões ISO/IEC 19788 para corrigir falhas e limitações observadas nos padrões Dublin Core e LOM e, ao mesmo tempo, preservar a compatibilidade com eles. Em especial, a nova especificação define de forma mais rígida os tipos e formatos dos valores de cada elemento do conjunto de metadados, bem como uma estrutura não-hierárquica para a representação dos elementos, o que facilita a descrição em RDF e o desenvolvimento de aplicações para a Web semântica (BOURDA *et al.*, 2010).

Enquanto os metadados são utilizados para representar diversos aspectos de um objeto de aprendizagem, o comportamento, em especial a coordenação da seleção e entrega de objetos

de aprendizagem para o aprendiz, é definido em especificações de sequenciamento (IMS, 2003b). Em linhas gerais, o sequenciamento define as atividades de aprendizagem e os recursos (objetos de aprendizagem e serviços) utilizados. Para tanto, definem-se papéis a serem desempenhados pelos aprendizes e professores nas atividades, as ações que eles podem realizar no ambiente de aprendizagem, as condições e regras para entrega e navegação dos recursos e mecanismos para aferir a satisfação de objetivos de aprendizagem.

Os modelos existentes para a representação do sequenciamento diferem quanto aos atores envolvidos, às ações permitidas e às interações e colaborações entre os atores com os objetos de aprendizagem. O *IMS Simple Sequencing* (IMS SS) (IMS, 2003b), por exemplo, propõe um modelo restrito, porém completo, para modelar o sequenciamento no cenário de educação a distância ou híbrida, com um ambiente Web de aprendizagem, sem atentar às colaborações entre os atores e considerando apenas o papel de aprendiz (o papel do professor não é apresentada no modelo). Enquanto isso, o *IMS Learning Design* (IMS LD) (IMS, 2003a) estabelece um arcabouço para a descrição de processos de ensino-aprendizagem que suplanta o IMS SS, permitindo a representação de processos educacionais com mais atores, papéis e colaborações.

O sequenciamento é definido no escopo do objeto de aprendizagem: as interações com o ambiente de aprendizagem restringem-se a interações de navegação. A troca de dados entre os objetos de aprendizagem e com o ambiente é uma questão distinta, que não pode ser antecipada ou prevista da mesma forma que no sequenciamento. A solução é o estabelecimento de um ambiente de execução e uma API com o ambiente virtual de aprendizagem, permitindo a livre programação da interação entre os objetos e o ambiente.

Atualmente, as interfaces (API) são simples: elas permitem o estabelecimento da comunicação entre o objeto de aprendizagem e o ambiente de execução e a troca de dados segundo um modelo de dados. Considera-se que os objetos de aprendizagem estão restritos ao ambiente Web e são iniciados por um ambiente virtual de aprendizagem. A comunicação entre eles ocorre por iniciativa do objeto de aprendizagem, limitado a obter e alterar valores (tal como a nota ou tempo gasto em determinada atividade) e a receber informações quanto ao sucesso das operações realizadas. Alguns trabalhos buscam expandir o modelo de execução para permitir a execução com menor acoplamento ao ambiente virtual de aprendizagem, adotando-se serviços Web (COSTAGLIOLA *et al.*, 2006). Dessa forma, um objeto de aprendizagem pode ser iniciado independentemente e, se necessário, acionar o ambiente de aprendizagem.

Estabelecidos os mecanismos para a representação dos metadados e comportamento (sequenciamento e execução) de um objeto de aprendizagem, resta definir como os dados serão especificados e distribuídos para instalação em ambientes virtuais de aprendizagem. A solução adotada para esse fim consiste na definição de pacotes que descrevem a estrutura

do objeto de aprendizagem e reúnem os recursos que o compõem, os metadados e descrição do sequenciamento. O empacotamento especifica o formato em que todas essas informações serão representadas e armazenadas.

O formato de pacote mais utilizado é o *Package Interchange File* (PIF) (IMS, 2007). Ele contém duas partes: o conteúdo, composto pelo objeto de aprendizagem em si e demais objetos dos quais ele dependa; e um manifesto, o qual contém metadados no formato LOM, a organização e o sequenciamento das atividades, e um inventário dos recursos e objetos de aprendizagem associados ao pacote. O manifesto é especificado em um documento XML, que é armazenado, junto com os recursos do pacote, em um arquivo compactado.

Metadados, sequenciamento, modelo de dados para comunicação com o ambiente de execução e o empacotamento definem os mecanismos para a representação de objetos de aprendizagem. No entanto, não é requisito que um objeto de aprendizagem utilize todos eles. Na próxima seção, apresentam-se os principais tipos de objetos de aprendizagem e os mecanismos de representação utilizados.

2.3. Abordagens para desenvolvimento de objetos de aprendizagem

O desenvolvimento de objetos de aprendizagem, principalmente em sua característica de reusabilidade, pode ser comparado ao desenvolvimento de software. A própria nomenclatura deriva do paradigma de orientação a objeto e do conceito de modularização como promotor do reúso. Analogamente, também podem ser utilizados métodos e processos para o desenvolvimento de objetos de aprendizagem.

2.3.1. Métodos

Observam-se, na literatura, duas gerações de métodos para o desenvolvimento de objetos de aprendizagem. A primeira, descrita na Seção 2.3.1.1, adota a visão de material didático como um hipertexto. Posteriormente, a atenção dirigiu-se aos padrões definidos por organizações como *Advanced Distributed Learning* (ADL), *Aviation Industry CBT Committee* (AICC) e *IMS Global Learning Consortium* (IMS) e, principalmente, à questão de reusabilidade. As técnicas da segunda geração preservam o uso de especificações formais, mas não o enfoque em características típicas de sistemas hipermídia (nós, âncoras, ligações, tipos de ligação, distinção entre estrutura e conteúdo). Além disso, assume-se o caráter interdisciplinar da atividade de desenvolvimento, definindo-se processos. Essa última geração é o tema das demais subseções.

2.3.1.1. Métodos precursores baseados em hipertexto

Principalmente após a criação da Web em 1990, hipertextos foram amplamente utilizados para a elaboração de textos e aplicações com fins didáticos. A literatura relata diversas técnicas para a engenharia de hipertextos, a citar HDM (GARZOTTO *et al.*, 1993), RMM (ISAKOWITZ *et al.*, 1995), OOHDM (SCHWABE; ROSSI, 1995), Trellis (STOTTS; FURUTA, 1989) e HMBS (TURINE *et al.*, 1997), dentre outras. Tais modelos baseiam-se em técnicas de especificação formal tais como máquinas finitas de estados (FSM), *statecharts* (HAREL, 1987) e Redes de Petri (PETRI, 1962).

No entanto, conforme visto na Seção 2.1, existem características intrínsecas à educação que devem ser consideradas. Em vista disso, buscou-se a combinação de modelos já estabelecidos na perspectiva educacional, tal como o mapa conceitual (NOVAK, 1977), e um ou mais modelos de hipertextos, com as adaptações necessárias para capturar ou facilitar a análise de elementos importantes de instrução.

Um exemplo desses métodos é o Daphne (KAWASAKI, 1996; KAWASAKI; FERNANDES, 1996). Ele possui três camadas: uma de acesso guiada pelos objetivos de aprendizagem (camada superior), uma de acesso direto às informações e a camada central, que contém o conjunto de conhecimento a ser trabalhado. A camada central organiza-se em entidades, que são definidas com mapas conceituais. A cada conceito, estabelece-se um hipertexto segundo uma variação do modelo *Relationship Management Data Model* (RMDM) do método *Relationship Management Methodology* (RMM) (GARZOTTO *et al.*, 1993), o qual consiste em conjunto estruturado de informações interligadas por elos estruturais, de perspectiva ou de aplicação.

Semelhantemente, o método *Educational Hyperdocuments Design Method* (EHDM) (PANSANATO; NUNES, 1999) também utiliza mapas conceituais, porém conciliados com o modelo de informação definido por Michener (1978). De cada mapa conceitual, derivam-se diferentes formas de navegação, representadas em um modelo navegacional inspirado no modelo do *Object-Oriented Hypermedia Design Model* (OOHDM) (SCHWABE; ROSSI, 1995), por sua vez uma evolução do *Hypertext Design Model* (HDM).

Leiva (2003) define o *Model for Distance Education* (MDE). Esse modelo utiliza a técnica de mapeamento conceitual e o modelo *Hypermedia Model Based on Statecharts* (HMBS) para especificar formalmente hipertextos. O mapa conceitual é restrito a uma estrutura de árvore de três níveis, constituindo uma hierarquia de conceitos. O material didático é especificado em HMBS, com particular atenção para o estabelecimento de mecanismos de avaliação diagnóstica, formativa e sumativa (LEIVA *et al.*, 2002).

2.3.1.2. Métodos baseados em linguagens de modelagem educacional

Uma linguagem de modelagem educacional (EML) é um modelo semântico de informação e associação que descreve o conteúdo e o processo de um objeto de aprendizagem sob a perspectiva educacional e com o objetivo de promover o reúso e a interoperabilidade (RAWLINGS *et al.*, 2002).

A primeira linguagem de modelagem educacional definida foi a *Open Universiteit Educational Modeling Language* (OUEML) (KOPER, 2001). Ela define um modelo para a descrição de unidades de ensino, utilizando atividades organizadas de forma análoga a uma peça de teatro: cada atividade de aprendizagem é dividida em atos. Os atores em cada ato são os alunos e os professores, que desempenham papéis conforme descritos pela linguagem, de forma semelhantes a um roteiro ou *script*. A cada ato, configura-se também o ambiente (cenário), povoando-o com objetos de aprendizagem. Ao final de cada ato, ocorre a sincronização de todos os atores e respectivas atividades, permitindo a execução do próximo ato.

Adotada com sucesso na *Open University of the Netherlands* (OUNL), a OUEML foi transferida para o consórcio IMS, que a revisou e a transformou no IMS LD (IMS, 2003a). Tal como a OUEML, o IMS LD permite a especificação de modelos em uma linguagem própria, a qual poderia ser executada, tal como um programa de computador, em ambientes virtuais de aprendizagem. O modelo organiza a aprendizagem em unidades compostas de atividades realizadas individualmente ou em grupos para um determinado contexto e objetivos educacionais. A analogia adotada é a mesma peça de teatro da OUEML: pessoas (alunos e professores) desempenham diversos papéis para cada ato da peça; em cada ato, diferentes atividades são realizadas pelos atores; cada atividade produz artefatos e reações em direção à satisfação dos objetivos de aprendizagem (ES; KOPER, 2006).

A estrutura de uma especificação IMS LD define os objetivos de aprendizagem, os pré-requisitos para a execução do objeto, os componentes da atividade instrucional e o *workflow* que rege a atuação e cooperação desses componentes. Um ambiente virtual de aprendizagem, ao executar a especificação, configura o ambiente com os objetos de aprendizagem a partir da descrição estrutural e coordena as atividades de acordo com o *workflow* definido. Este fluxo de trabalho consiste na instanciação de cada parte da atividade de aprendizagem, associação de aprendizes e professores aos papéis especificados e a execução do roteiro de atividades pertinentes (OLIVIER; TATTERSALL, 2005).

O modelo conceitual do IMS LD apresenta-se conforme a Figura 2.4. Dos elementos apresentados, os mais complexos são relativos ao fluxo de trabalho: componentes e recursos. Eles são responsáveis pela coordenação dos aprendizes e instrutores na realização de atividades com os objetos de aprendizagem. A especificação IMS LD organiza-se em três níveis, A, B e C. O primeiro nível (A) restringe-se a coordenar a escolha e a sequência de execução

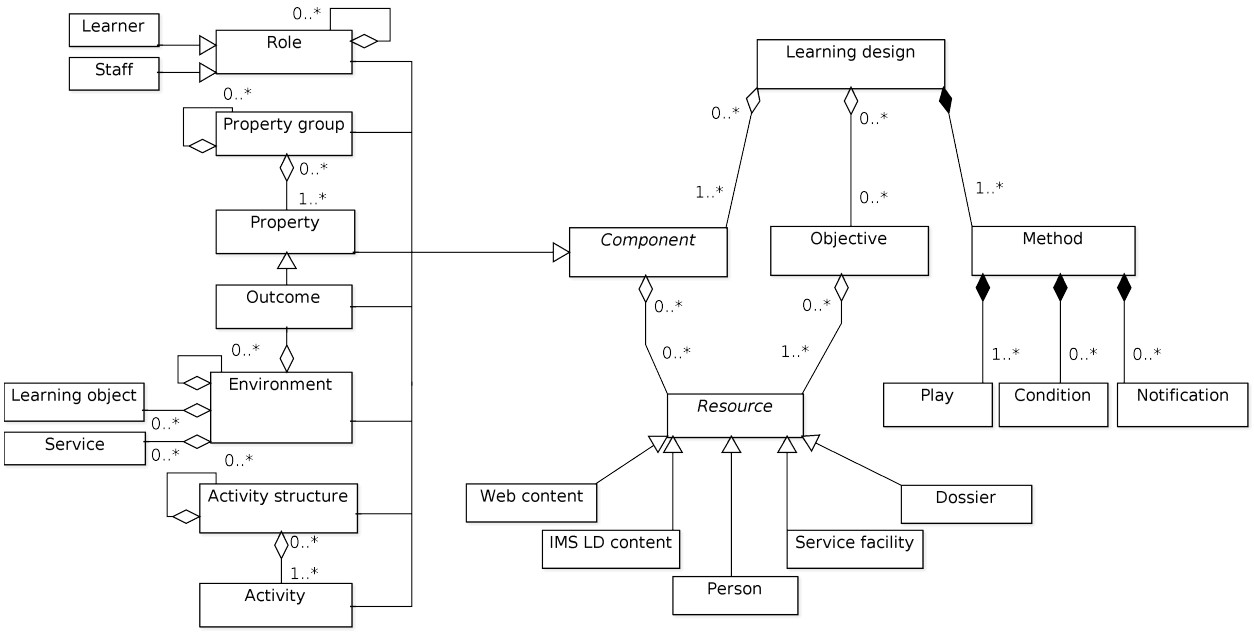


Figura 2.4: Modelo conceitual da especificação IMS LD (IMS, 2003a, p. 9).

dos conjuntos de atividades. Trata-se de uma forma simples de organização, dado que não considera mudanças do ambiente e das pessoas em seus papéis. O nível B permite a personalização da atividade de aprendizagem pelo emprego de propriedades e condições. As propriedades armazenam dados sobre a atividade, as pessoas e os papéis, com um escopo local (valores mantidos apenas durante a execução da atividade) e global (valores mantidos entre execuções). As condições estabelecem regras para a execução das atividades em função das propriedades, do estado de uma atividade ou expiração de um temporizador. Em conjunto, as condições e as propriedades permitem a criação de atividades de aprendizagem adaptáveis. No entanto, o nível B está restrito a alterações de propriedades. Apenas no nível C encontra-se um mecanismo de notificação baseado em eventos.

Embora a IMS LD seja a principal proponente de linguagem de modelagem educacional, ela não é a única. Outra vertente consiste na adoção de linguagens com notação gráfica, o que facilita a comunicação dos modelos para as equipes interdisciplinares encarregadas do desenvolvimento de objetos de aprendizagem. Um exemplo de linguagem que adota essa abordagem é a *Perspective-oriented Educational Modeling Language* (PoEML). Ela incorpora, além de representações gráficas, o conceito de separação de interesses utilizado em programação orientada a aspectos (CAEIRO-RODRíGUEZ, 2008). Dois tipos de interesses ortogonais são estabelecidos: perspectivas e aspectos. Definem-se treze perspectivas: estrutural, funcional, participantes, ambiente, dados, ferramentas, organizações, ordem, temporal, autorização, percepção, interação e causal. Um aspecto define o grau de controle sobre cada uma dessas perspectivas. Quatro tipos de aspectos são definidos: determinante, condicionado, baseado em eventos e baseado em decisões.

Em termos práticos, a PoEML define um modelo para cada perspectiva e os aspectos

definem como esses modelos podem ser alterados quanto à estrutura e ao comportamento. Um aspecto do tipo *determinante* não introduz nenhuma alteração nos modelos enquanto que um aspecto baseado em eventos permite a adição de novos comportamentos nos momentos em que ocorrem eventos significativos no modelo (*e.g.*, o início ou o encerramento de uma atividade).

2.3.1.3. Métodos baseados em grafos de composição

Defude e Farhat (2005) propõem a criação de *Intensional Learning Object* (ILO): um objeto de aprendizagem flexível, criado a partir da composição de outros objetos, os quais são escolhidos de um repositório considerando um conjunto de critérios. Um ILO é definido por três modelos: modelo de domínio, modelo do aprendiz e modelo do objeto de aprendizagem. Esses modelos são inerentes do modelo de hipertexto *Adaptive Hypermedia Application Model* (AHAM) (BRA *et al.*, 1999), por si uma extensão do modelo de referência para hipertextos Dexter (HALASZ; SCHWARTZ, 1994).

O modelo de domínio descreve os conceitos e suas relações do conteúdo a ser ensinado. Ontologias são utilizadas para esse fim, definindo-se os conceitos e dois tipos de relacionamentos: hierárquicos e retóricos (antítese, contraste, extensão, etc) ou específicos do domínio.

O modelo do aprendiz define os conhecimentos prévios do aluno, suas preferências e metas. O conhecimento prévio é obtido de avaliações educacionais anteriores, as preferências (idioma, tempo disponível ou desejável para usar um objeto de aprendizagem, formato de entrega do objeto, etc) são definidas pelo próprio aprendiz, bem como as metas (as quais são selecionadas do modelo de domínio).

O modelo do objeto de aprendizagem define características educacionais (correspondentes aos metadados do LOM), conteúdo e características semânticas. O componente semântico define: (1) a função que o objeto tem em relação aos conceitos do modelo de domínio (funções essas definidas por uma ontologia educacional); (2) pré-requisitos (o grau de conhecimento requerido quanto aos conceitos definidos no modelo de domínio); e (3) uma função de aquisição (que estabelece o quanto de um conceito aprende-se após o uso do objeto de aprendizagem). O conteúdo em si do objeto de aprendizagem é um grafo acíclico de composição o qual possui três tipos de vértices:

- Vértice-Operador (*operator-node*): Define a operação de composição dos vértices a ele relacionados. Três tipos de operações são suportados: SEQ (sequência), ALT (alternativa) e PAR (paralelo).
- Vértice-Consulta (*querynode*): Define uma consulta intencional (e daí o objeto de aprendizagem intencional) com as condições a serem satisfeitas pelos objetos de aprendizagem recuperados do repositório. As condições são definidas em função da semântica do

objeto: pré-requisitos, conteúdo e aquisição.

- Vértice-LO (*LO-node*): Especifica um objeto de aprendizagem.

Instâncias de um grafo acíclico, com a resolução dos vértices-consulta, constituem um novo objeto de aprendizagem. De modo a assegurar um grau mínimo de qualidade dos objetos de aprendizagem intencionais, define-se um conjunto de métricas e um sistema de tipos (DEFUDE; FARHAT, 2005). O sistema de tipos permite a atribuição de restrições quanto às propriedades estruturais do grafo de composição (*e.g.*, limitar a quantidade de vértices do grafo ou a quantidade de arestas de um vértice) e restrições quanto às propriedades semânticas dos conteúdos e do aprendiz (*e.g.*, requerer que um vértice tenha conteúdos do tipo instrução, exemplo e exercício). Dessa forma, todos os modelos definidos para um ILO podem ser utilizados como subsídios para definição de restrições.

2.3.1.4. Métodos baseados em Rede de Petri

Lin *et al.* (2004) estudam a aplicabilidade de Redes de Petri para a modelagem do sequenciamento de conteúdo didático. Além do rigor e formalismo estabelecidos pela linguagem, adequados para expressar a dinamicidade do processo de aprendizado, Redes de Petri possuem uma representação gráfica, que facilita a comunicação, e ferramentas para autoria e simulação dos modelos.

A navegação nos objetos de aprendizado pode ser representada pelas estratégias especificadas no *Sharable Content Object Reference Model* (SCORM) (padrão IMS SS). No âmbito do trabalho de Lin *et al.* (2005), são definidas seis especificações em Rede de Petri para representar as sequências típicas:

- *Flow*: Define um caminho linear de aprendizado. Os diferentes conteúdos didáticos são apresentados em uma única ordem.
- *Choice*: Define um caminho com possibilidade de escolha do conteúdo didático a ser apresentado.
- *Skip*: Permite ignorar um determinado conteúdo didático temporariamente.
- *Limitcondition*: Define uma condição para o acesso a um conteúdo didático.
- *Suspend*: Permite a suspensão temporária das atividades de aprendizado.
- *Rollup*: Avalia o desempenho do aprendiz em relação aos objetivos de aprendizado do objeto de aprendizado.

Para atender essas estratégias, Lin *et al.* (2005) definiram uma versão modificada da Rede de Petri Colorida, incorporando-se um tipo novo de lugar utilizado para o controle da navegação. Os lugares do tipo padrão representam materiais didáticos, tais como lições e avaliações. Os lugares de controle, em conjunto com os demais mecanismos de Redes de

Petri Coloridas, permitem o gerenciamento da navegação entre os materiais didáticos e a implementação das estratégias de navegação.

Nesta mesma linha, Su *et al.* (2006) utilizam redes de Petri coloridas para a especificação de objetos de aprendizagem. A abordagem *Object Oriented Course Modeling* (OOCM) permite a implementação de sequências de atividades. Cada atividade é descrita por uma Rede de Petri hierárquica, definindo especificações de fluxo que implementam o padrão IMS SS. Além disso, define-se uma transformação da Rede de Petri hierárquica em uma árvore de atividades e uma segunda transformação dessa árvore em um objeto de aprendizagem no padrão SCORM.

Ainda na linha de Redes de Petri, Souza *et al.* (2007) desenvolveram a técnica *Learning Object production with Color Petri Net* (LOCPN). Ela difere das demais por não tratar do sequenciamento tal como definido no IMS SS, adotando um modelo mais simplificado quanto à navegação, porém considerando aspectos de interação. A motivação para a criação do modelo foi a redução das imprecisões na comunicação dos aspectos técnicos e pedagógicos entre os diferentes interessados durante a produção e o uso do documento *Roteiro de atividades* do Processo Rede Internacional Virtual de Educação (RIVED).

Um objeto de aprendizado, na concepção da LOCPN, é um conjunto de telas com as quais o usuário interage e navega. As telas são controles em que os conteúdos didáticos são mostrados. A técnica estabelece três requisitos de modelagem: fluxo, interação e navegação. Os fluxos são as relações entre as telas de um objeto de aprendizado. As interações são as possibilidades de interação do usuário, tal como o acionamento de um botão de uma tela. A navegação associa as interações aos fluxos, permitindo a mudança de telas.

Os fluxos classificam-se em direto-interativo, direto-automático e direto-condicional. No fluxo direto-interativo, o acionamento da relação entre as telas ocorre em reação a uma interação do usuário (por exemplo, o acionamento de um botão). O fluxo direto-automático não requer uma ação do usuário: o seu acionamento ocorre espontaneamente, programado pelo sistema. No fluxo direto-condicional, a tela destino da relação é definida por uma condição qualquer, geralmente o estado da tela anterior (por exemplo, a tela que mostra a próxima questão se a ficha contém uma informação que indique que a questão foi respondida corretamente).

A técnica LOCPN, para atender aos requisitos estabelecidos, define uma linguagem formal, baseada em Redes de Petri Coloridas, para a descrição de objetos de aprendizado. O modelo do objeto de aprendizado é definido como uma rede hierarquicamente organizada. Cada tela é especificada por uma Rede de Petri, representando-se os fluxos, as interações e a navegação. Os lugares da rede correspondem a telas (telas normais, *pop-ups* ou caixas de diálogo). Os tipos das fichas definem o comportamentos quanto às telas: a ficha `e`, quando presente em lugares que representam telas, indica que o foco está naquela tela; os demais tipos de ficha representam os controles associados às telas. As transições podem ser marcadas

com pesos, os quais indicam as ações sobre os controles da tela ou mudança de foco. Os fluxos entre as telas são atendidos da seguinte forma:

- O fluxo direto-interativo é acionado por uma ficha correspondente a um controle gráfico no lugar-origem, uma transição cuja guarda é uma ficha de controle gráfico e que produz uma ficha `e`, que transfere o foco para a tela do lugar-destino.
- O fluxo direto-condicional é acionado por uma ficha correspondente a um controle gráfico no lugar-origem, uma transição cuja guarda é uma ficha de controle gráfico e uma condição definida por uma expressão que considera as informações contidas na ficha recebida.
- O fluxo direto-automático é acionado por uma ficha `e`, uma transição cuja guarda é uma ficha `e` e que produz uma ficha `e`.

A modelagem da interação é satisfeita pela definição de modelos de rede para cada tipo de controle de uma tela. Em geral, a condição de guarda das transições é definida em função de uma ação do usuário. Ao ativar um lugar, todas as transições relacionadas ao lugar são habilitadas, mas não executadas. As expressões de guarda são definidas de modo que apenas uma ação direta (interação) do usuário permite a sua satisfação. A navegação é tratada como um caso especial de interação, sendo definida por expressões de guarda com mais de uma condição e, portanto, requerendo mais de uma ficha para a ativação da transição.

2.3.1.5. Métodos baseados em máquinas de estados

A *Integrated Modelling Approach – Conceptual, Instructional and Didactic* (IMA-CID) caracteriza-se por um conjunto de modelos para a representação de conteúdos educacionais (BARBOSA, 2004; BARBOSA; MALDONADO, 2006a), reunindo em uma única proposta diferentes perspectivas de modelagem: conceitual, instrucional e didática. Cada perspectiva é representada por um modelo distinto, conforme sintetizado na Figura 2.5.

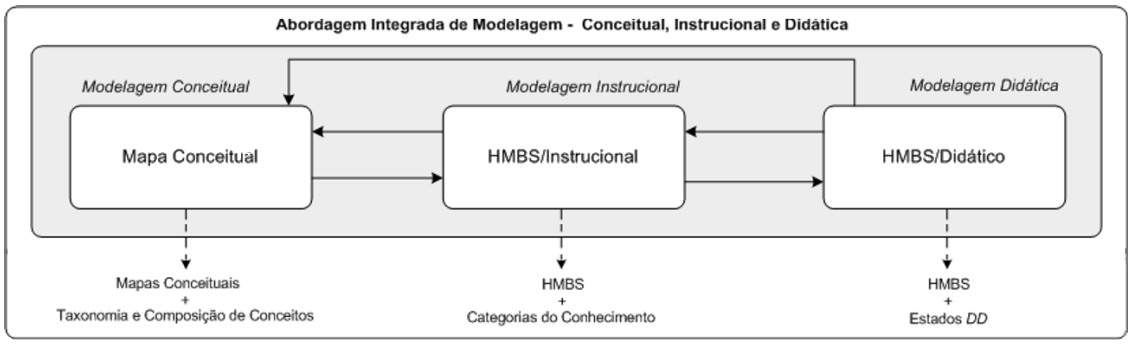


Figura 2.5: IMA-CID: Modelos utilizados em cada perspectiva da abordagem e a forma com que eles se relacionam. (BARBOSA, 2004, p. 105).

O modelo conceitual consiste em uma descrição em alto nível do domínio de conhecimento

que se deseja ensinar. Sua construção envolve a definição dos conceitos relevantes para a compreensão do domínio e a especificação da forma pela qual os mesmos se relacionam. Ele é representado por um mapa conceitual estendido com primitivas para a representação de taxonomia e composição de conceitos. Um mapa conceitual representa diferentes conceitos referentes a um domínio de conhecimento na forma de conceitos relacionados em uma estrutura hierárquica (NOVAK, 1990). Nesse modelo, um conceito é qualquer regularidade ou padrão observado no domínio de conhecimento, correspondendo a objetos ou eventos. O relacionamento entre dois conceitos pode ser de três tipos: classificação (taxonomia), composição e específicos de domínio. O relacionamento de classificação permite a definição de especialização de um conceito, definindo uma taxonomia. A composição presta-se à representação da composição de conceitos, particularmente útil quando as diferentes partes possuem relacionamentos com outros conceitos (que seriam de difícil compreensão caso fosse utilizado apenas o conceito composto/inteiro). O relacionamento específico de domínio é todo aquele que não serve para classificação ou composição, indicando uma relação particular ao domínio de conhecimento sendo modelado e, usualmente, relacionado a uma dependência funcional específica entre os conceitos.

Um exemplo de modelo conceitual sobre teste de software com a técnica de análise de mutantes (DEMILLO, 1980) é apresentado na Figura 2.6. Os conceitos são representados pelos retângulos rotulados e os relacionamentos por arestas rotuladas entre dois conceitos. Relacionamentos de classificação apresentam uma seta não-preenchida no final da aresta, junto ao conceito mais genérico (*e.g.*, entre os conceitos `Mutant` e `Error-Revealing Mutant`, `Dead Mutant` e `Equivalent Mutant`). As composições são identificadas por uma aresta

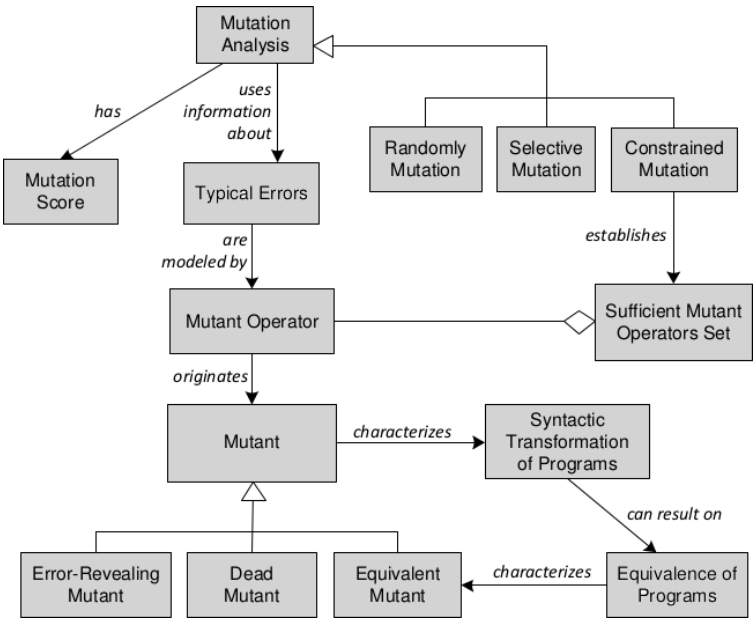


Figura 2.6: Modelo conceitual (IMA-CID) sobre o critério de análise de mutantes (BARBOSA, 2004, p. 186).

com um losango junto ao conceito que faz parte da composição (*e.g.*, `Mutant Operator` é composto por `Sufficient Mutant Operators Set`). Os relacionamentos que possuem uma seta simples e preenchida são específicos de domínio (*e.g.*, `Mutant Analysis` assume `Coupling Effect`).

O *modelo instrucional* define itens de informação e elementos instrucionais. Os itens de informação seguem o modelo do *Component Display Theory* (CDT) (MERRILL, 1983), que classifica os conhecimentos em conceitos, fatos, procedimentos ou princípios:

- Conceito: Símbolos, eventos e objetos que compartilham características e são identificados pelo mesmo nome. Constituem grande parte da linguagem e seu entendimento é essencial para a comunicação.
- Fato: Partes de informação logicamente associadas. Nomes, datas e eventos são exemplos de fatos.
- Procedimento: Conjunto ordenado de passos que visa resolver um problema ou atingir um objetivo.
- Princípio: Explicações e deduções sobre o porquê de determinados acontecimentos e das maneiras específicas com que eles ocorrem. Um princípio (ou lei) resume grande quantidade de fatos e possibilita a previsão de novos fatos (CERVO *et al.*, 2007, p. 19).

Os elementos instrucionais, cuja finalidade é complementar os itens de informação, são divididos em elementos explanatórios, exploratórios e de avaliação. Os elementos explanatórios definem informações complementares utilizadas na explicação de um dado item de informação (exemplos, dicas, sugestões de estudo, referências). Elementos exploratórios permitem a prática e a navegação pelos itens de informação relacionados (exercícios, simulações, utilização de ferramentas). As avaliação diagnósticas, formativas e sumativas do aprendizado são realizadas por elementos de avaliação.

Os itens de informação e elementos instrucionais de um modelo instrucional são representados como parte de uma página de um documento especificado de acordo com o modelo HMBS. Esse modelo permite a especificação de hipertextos (`Hypertext`) como *statecharts* (HAREL, 1987). Explicando brevemente, um *statechart* é uma extensão de *Finite State Machine* (FSM) que permite a composição e aninhamento de estados (definindo uma hierarquia) e concorrência de estados (estados AND). Em HMBS, um hipertexto é representado por um *statechart* (e seus estados) e por páginas. A cada estado atômico ou composto do tipo XOR (que não permite concorrência), associa-se uma página que contém os recursos (textos, imagens) e âncoras que estão associadas aos eventos que disparam a transições entre os estados do *statechart*.

O modelo instrucional da IMA-CID é uma extensão do HMBS em que a página é definida por itens de informação e elementos instrucionais, conforme mostrado na Figura 2.7. A cada um dos itens do modelo instrucional, associam-se recursos (`Resources`), tal como previsto

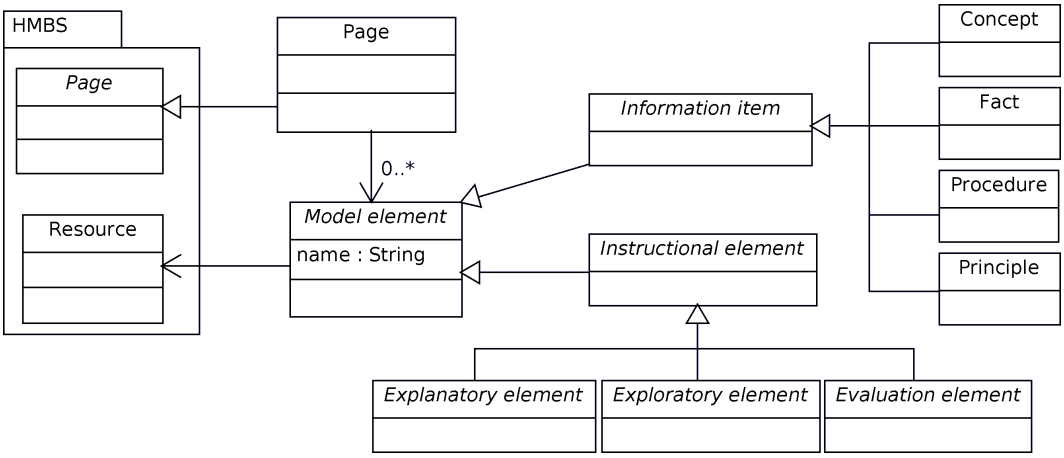


Figura 2.7: IMA-CID: Elementos do modelo instrucional.

no modelo HMBS. Além disso, a IMA-CID considera apenas os aspectos de decomposição hierárquica (ignorando momentaneamente as transições). Em virtude dessas diferenças, essa extensão do HMBS é denominada *HMBS/Instrucional*.

Considerando o modelo conceitual da Figura 2.6, um possível modelo instrucional é aquele apresentado na Figura 2.8. Em síntese, cada estado atômico representa uma página que contém um item de informação ou elemento instrucional. Por exemplo, `SelectiveMutation` é um estado atômico: a categoria de conhecimento (tipo de item de informação) é definido na segunda linha do rótulo e o tipo de recurso encontra-se na terceira linha. Os estados compostos podem ser do tipo `XOR`, *i.e.*, apenas um subestado está ativo, ou `AND`, *i.e.*, todos os subestados estão ativos. Por exemplo, o estado composto `GeneralMutant` é do tipo `AND`, pois contém dois estados que não podem estar ativos simultaneamente: `Mutant` e `MutantClassification`. Enquanto isso, o estado `Mutant` possui quatro estados que ficam simultaneamente ativos. Na prática, isso significa que serão apresentados conceitos e fatos sobre mutantes na forma de textos e figuras.

O modelo didático é responsável por associar os elementos do modelo instrucional, estabelecendo relações de precedência e relacionamentos didáticos entre os itens e elementos, definindo, assim, uma sequência de apresentação. Ele é representado como uma extensão do modelo *HMBS/Instrucional*, incorporando a ideia de estados *DD* (*Dynamically Defined* – Definido Dinamicamente), utilizados como apoio à especificação aberta dos aspectos de navegação. Esse modelo é denominado *HMBS/Didático*, organizando-se da forma exposta na Figura 2.9. Para implementá-lo, são alteradas a sintaxe e a semântica do modelo *HMBS*, acrescentando o estado composto do tipo `DD` (`XOR DD State`) e alterando o comportamento do *statechart* para permitir a transição entre todos os estados contidos em um estado `DD` e a ativação de qualquer estado `DD` da hierarquia de estados-pai `DD`.

A ideia de especificação aberta explorada no modelo didático da IMA-CID refere-se à possibilidade de representar as sequências de apresentação entre conceitos e informações

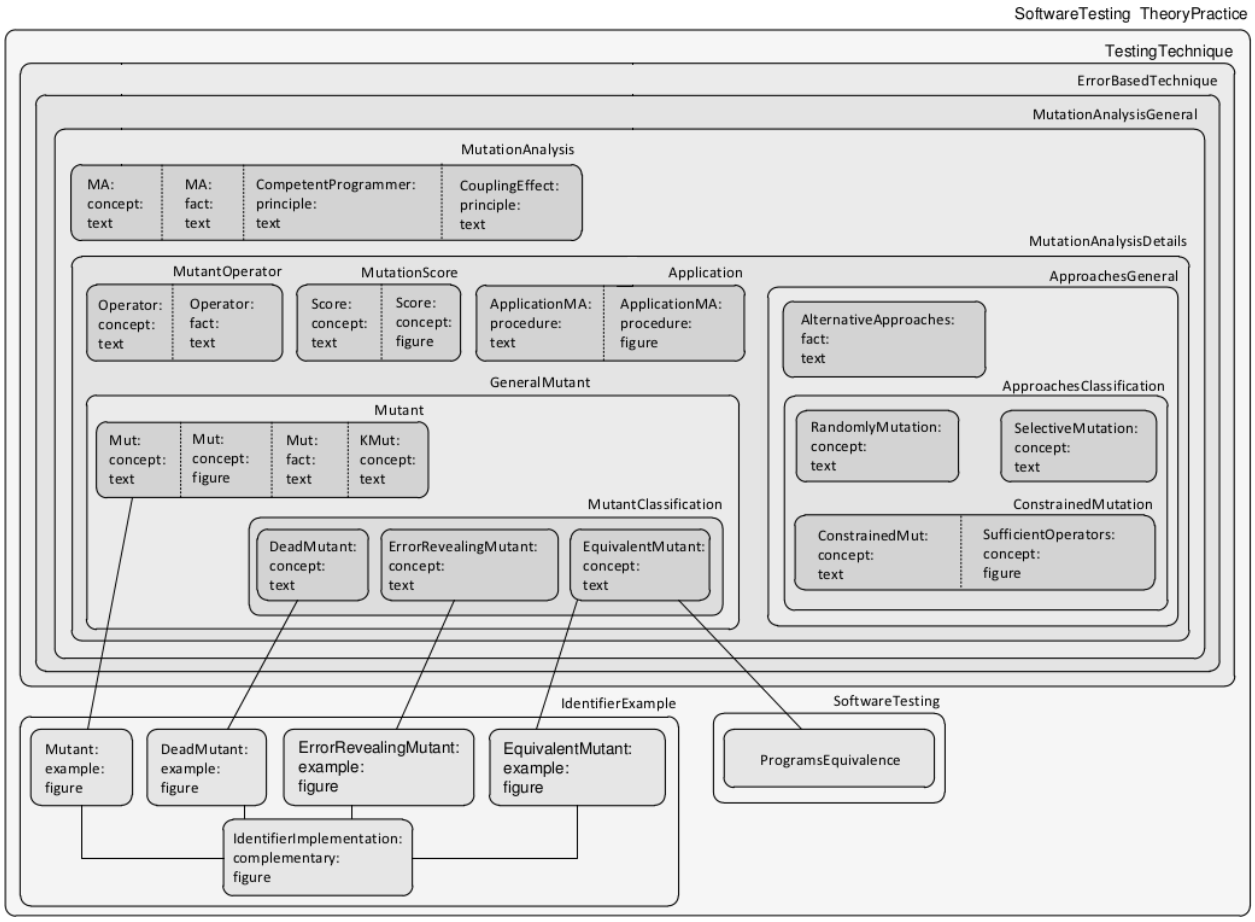


Figura 2.8: Modelo instrucional (IMA-CID) sobre o critério de análise de mutantes (BARBOSA, 2004, p. 189).

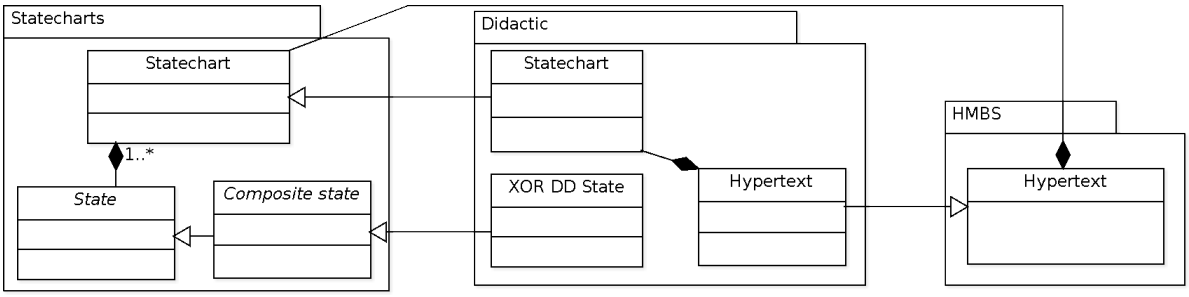


Figura 2.9: IMA-CID: Elementos do modelo didático.

pertinentes de maneira flexível e personalizada. Dependendo de fatores como duração do curso, objetivos de aprendizado e público-alvo, maneiras diferenciadas de apresentação e navegação pelo mesmo conteúdo são exigidas. Uma especificação aberta permite que todas as possíveis sequências de apresentação do conteúdo sejam representadas no mesmo modelo didático. Nesse sentido, a partir de um único modelo, diversas implementações do mesmo conteúdo podem ser geradas em função dos fatores mencionados acima. Quando fisicamente implementada no módulo educacional, ela possibilita que a navegação seja definida pelo próprio usuário, em tempo de execução. Em outras palavras, o usuário tem liberdade para

decidir, dinamicamente, quais tópicos devem ser abordados e em que ordem devem ser apresentados (BARBOSA, 2004).

2.3.1.6. Métodos de desenvolvimento baseados em modelos (MDD)

O desenvolvimento de objetos de aprendizagem pode ser compreendido como o progressivo desenvolvimento de artefatos com base em modelos de conhecimento tácitos e explícitos. Em Engenharia de Software, tal abordagem também é válida, criando-se modelos aos quais são iterativamente acrescentadas informações em diferentes perspectivas até que ocorra a efetiva implementação do modelo completo. Um problema comum nessa abordagem é o abandono do modelo que serviu de base para a criação de um novo, ou seja, considerando uma interação i do processo de desenvolvimento, erros detectados na construção modelo m não são corrigidos no modelo $m-1$ e, principalmente, na iteração $i+1$, a atividade de desenvolvimento ignora o modelo $m-1$, o que aumenta o esforço para representar e analisar os interesses de especialidade do modelo $m-1$, ou, no pior caso, utiliza o modelo com erros.

Em Engenharia de Software, a abordagem de desenvolvimento orientada a modelos (MDD – *Model-Driven Development*) trata esses problemas pela criação e alteração de modelos e a sucessiva transformação entre eles durante todo o ciclo de vida do produto (STAHL; VöLTER, 2006). Dessa forma, modificações feitas no modelo $m-1$ refletem automaticamente no modelo subsequente, eliminando a necessidade de atualizar manualmente o modelo m para refletir o estado do modelo anterior. Além disso, as transformações entre os modelos são automatizadas, capturando o conhecimento antes tácito dos desenvolvedores quanto às relações existentes entre os modelos e a forma com que essas relações são realizadas nos modelos.

A abordagem *Model-Driven Development* (MDD) requer a utilização de linguagens específicas de domínio (DSL - *Domain Specific Language*) para a especificação de modelos em diferentes níveis de abstração e mecanismos de transformação para gerar artefatos (casos de teste, documentação, código-fonte, etc) a partir desses modelos (SCHMIDT, 2006). Uma importante característica de *Domain Specific Language* (DSL) é sua proximidade com o domínio do problema e a facilidade de comunicação com os outros atores do processo, utilizando-se de transformações automatizadas (SCHMIDT, 2006). No contexto de objetos de aprendizagem, o envolvimento de atores de diferentes especialidades e a comunicação entre eles é uma importante questão, o que torna apropriada essa abordagem.

Em continuação à técnica LOCPN (SOUZA *et al.*, 2007), desenvolveu-se uma abordagem para o desenvolvimento de objetos de aprendizagem baseado em modelos, a partir dos conceitos de MDD (SOUZA *et al.*, 2008). Ela ocorre no contexto de um processo organizado em cinco fases: delimitação do escopo pedagógico, engenharia de requisitos, projeto, implementação e verificação/validação. Para a primeira fase, definiu-se a DSL *ExpertDSL* (SOUZA *et al.*, 2008). O público da linguagem é constituído por conteudistas e pedagogos.

Além das diferentes linguagens, a abordagem define: (1) mecanismos de integração, evolução e reutilização dos modelos por ela definidos, de forma a permitir a introdução de modificações nos objetos de aprendizagem sem o comprometimento da qualidade deles; e (2) mecanismos de transformação entre modelos para a geração dos códigos-objeto. A Figura 2.10 expõe essa abordagem, conforme descrito a seguir. Em linhas gerais, o modelo desenvolvido na primeira fase (Delimitação de escopo pedagógico) define o modelo arquitetural do objeto de aprendizagem. Duas transformações são então aplicadas a esse modelo: uma gera uma descrição de atividades de aprendizagem, que é repassada para os desenvolvedores. A segunda transformação produz um modelo de reconfiguração, que receberá, posteriormente, os componentes desenvolvidos pelos desenvolvedores, resultando em um objeto de aprendizagem personalizável (CLO) (SOUZA *et al.*, 2010).

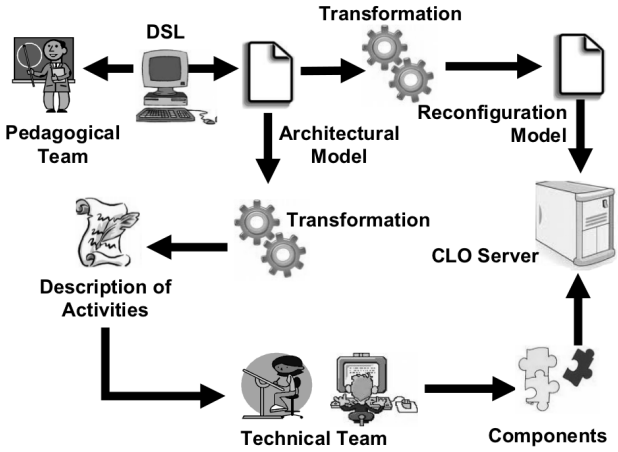


Figura 2.10: Processo de transformação de modelos e geração de objeto de aprendizagem personalizável (SOUZA *et al.*, 2010).

O modelo arquitetural do objeto de aprendizagem, produzido na primeira fase do desenvolvimento, estabelece os pontos do objeto que podem ser alterados. No caso da abordagem de Souza *et al.* (2010), o reúso dá-se pela alteração dos componentes utilizados no objeto, mas preservando-se a estrutura e o projeto instrucional. Quanto aos componentes, limitam-se também as ações a que eles podem ser submetidos: edição, ativação, substituição e fracionamento. Apenas cinco tipos de componentes são considerados, conforme apresentado na Tabela 2.1. A maioria dos componentes podem ser ativados ou desativados, mas o fracionamento (seleção de uma parte do componente) só é permitido para vídeo e áudio. Botões podem ser editados quanto às dimensões e etiquetas, mas não quanto à ação que eles realizam. O único componente que pode ser livremente editado é o texto.

Os componentes são utilizados em atividades que compõem um cenário de uso. Esses cenários podem ser relacionados a outros, um grupo de ações, e também podem ser compostos por outros cenários, formando uma estrutura hierárquica. Transições podem ser definidas entre os cenários, permitindo a execução de ações tal como a reconfiguração da interface

Tabela 2.1: Componentes de um objeto de aprendizagem personalizável e conjunto de operações permitidas.

Tipo	Edição	Ativação/Desativação	Substituição	Fracionamento
Vídeo	Não	Sim	Não	Sim
Áudio	Não	Sim	Não	Sim
Imagem animada	Não	Sim	Não	Não
Imagem	Não	Sim	Sim	Não
Texto	Sim	Sim	Não	Não
Botão	Parcial	Não	Não	Não

do usuário. Cada cenário é descrito por um *wireframe*, ou seja, uma estrutura que esboça a interface gráfica. O *wireframe* é criado com os componentes apresentados na Tabela 2.1, representados por um elemento de mídia para vídeos, áudio e imagens e por elementos de controle (botões, texto, caixas de listagem, caixas de verificação inclusivas e exclusivas). Todos esses elementos são representados em um metamodelo, estabelecendo uma DSL para a descrição arquitetural do objeto de aprendizagem, conforme apresentado na Figura 2.11.

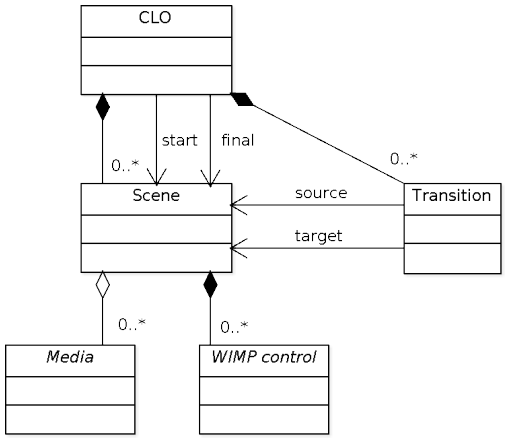


Figura 2.11: DSL para descrição arquitetura de um CLO. Adaptado de Souza *et al.* (2011, p. 27).

Ainda na linha de desenvolvimento baseado em modelos, Dodero *et al.* (2007) também adota MDD como uma solução adequada para representar os diferentes modelos, níveis de abstração, contextos e interesses do processo de engenharia de objetos de aprendizagem (DODERO; DIEZ, 2006). A abordagem idealizada denomina-se *Model-Driven Learning Design* (MDLD). Ela agrega modelos de projeto instrucional (tal como IMS LD), modelos de mapeamento (abstrações dos modelos instrucionais e definição de conjunto de regras de mapeamento desse modelo para o projeto instrucional), modelos de blocos de construção (referentes a elementos concretos que podem ser atribuídos a modelos de mapeamento) e modelos de moldes (*templates*), que servem de referência para a construção de partes de outros modelos. Com exceção dos modelos de mapeamento, todos os demais estão associados a um metamodelo.

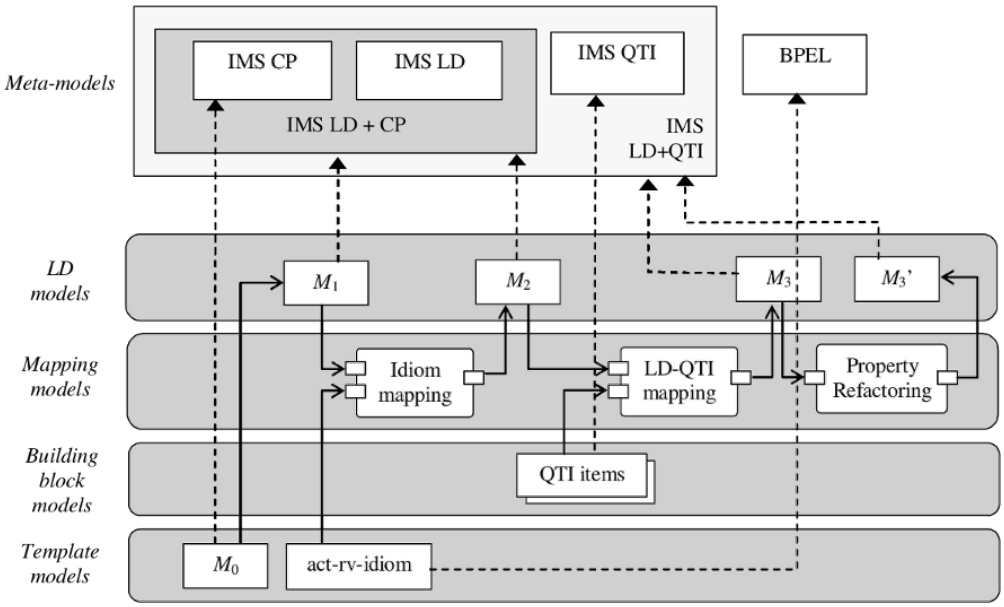


Figura 2.12: Exemplo com transformações de modelos conforme o método MDLD (DODERO *et al.*, 2007).

As transformações no MDLD são definidas como o encadeamento de processos de modelos de mapeamento, conforme ilustrado na Figura 2.12. A partir de um molde de um manifesto de um pacote *IMS Content Packaging* (IMS CP) M_0, instancia-se um modelo de projeto instrucional M_1. O modelo M_1 e os moldes para a definição de atividades $act-rv-idiom$ são utilizadas como entrada para as transformações com o modelo de mapeamento das construções de sequenciamento. Ao modelo M_2 resultante adicionam-se questões de avaliação $QTIitems$, acionando-se a transformação $LD-QTImapping$ e produzindo o manifesto M_3. Ajustes nas propriedades do modelo M_3 são realizados pela transformação $PropertyRefactoring$, que gera o objeto de aprendizagem M_3'.

2.3.2. Processos

O desenvolvimento de objetos de aprendizagem é uma atividade multidisciplinar, englobando pedagogos, comunicadores e engenheiros de software em uma organização distribuída. A aplicação de práticas sistemáticas é fundamental para a garantia da produtividade do processo de desenvolvimento e da qualidade dos objetos gerados (BARBOSA, 2004).

Apesar de sua importância, as iniciativas para a definição desses processos ainda são poucas. Identificou-se, na literatura, um trabalho que engloba todos os aspectos, inclusive organizacionais, e outros que relatam, superficialmente, os processos utilizados no desenvolvimento de objetos de aprendizagem. Esses trabalhos são descritos a seguir.

2.3.2.1. SP-DEM

O Processo Padrão para Desenvolvimento de Módulos Educacionais (SP-DEM) incorpora práticas de projeto instrucional, de modelagem de objetos de aprendizagem e de desenvolvimento cooperativo e distribuído (BARBOSA, 2004; BARBOSA; MALDONADO, 2006b). Ele foi estabelecido com base na norma ISO/IEC 12207 (ISO/IEC, 1995), organizando-se em três grupos de processos: fundamentais, organizacionais e de apoio.

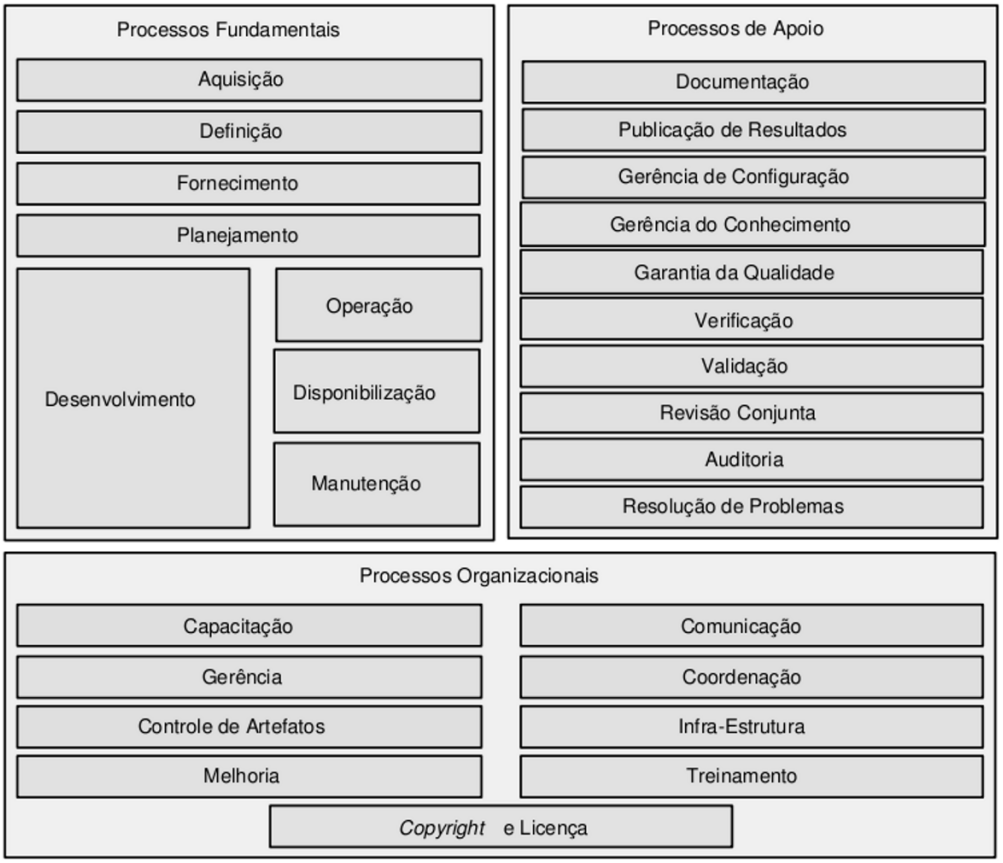


Figura 2.13: Estrutura geral do SP-DEM (BARBOSA, 2004, p. 127).

Observando-se a Figura 2.13 e comparando-a com a norma ISO/IEC 12207:1995, os processos fundamentais para objetos de aprendizagem diferem, sobretudo, com respeito à inclusão do *Processo de Disponibilização*, que trata das atividades e tarefas do instrutor do módulo. Já os processos de apoio distinguem-se pela inclusão do *Processo de Gerência do Conhecimento*, responsável por tratar os aspectos pertinentes ao gerenciamento e controle do conhecimento sob o qual o módulo está fundamentado. Quanto aos processos organizacionais, esses se diferenciam basicamente pela definição do *Processo de Copyright e Licença*, no qual são abordados aspectos pertinentes aos direitos autorais e aos termos de uso e distribuição do módulo.

Os Processos Fundamentais possuem uma forte correlação com o modelo ADDIE do processo instrucional, descrito no Seção 2.1, conforme apresentado na Tabela 2.2. Para

o desempenho dos processos do grupo de processos fundamentais do *Standard Process – Development of Educational Modules* (SP-DEM), a abordagem IMA-CID é utilizada.

Tabela 2.2: Correspondência entre o modelo ADDIE e os processos fundamentais do SP-DEM (BARBOSA, 2004).

Modelo ADDIE	**Processo fundamentais**
Análise	Definição e desenvolvimento
Projeto	Desenvolvimento
Desenvolvimento	Desenvolvimento
Implementação	Operação, Disponibilização
Avaliação	Manutenção

Cabe salientar que, embora os *Processos Fundamentais* sejam essenciais para o desenvolvimento de objetos de aprendizagem, os demais grupos de processo assumem papéis relevantes para a garantia da qualidade da execução e dos resultados do processo.

2.3.2.2. ISDMeLO

O processo ISDMeLO define uma abordagem para o desenvolvimento de objetos de aprendizagem que aplica princípios de várias teorias educacionais (BARUQUE; MELO, 2004). Ele integra duas visões, a educacional e a de objetos de aprendizagem, com uma estratégia *top-down*, ilustrada na Figura 2.14. Primeiro, define-se a teoria educacional (comportamentalista, cognitivista ou construtivista). A seguir, escolhem-se os princípios a serem seguidos daquela teoria: esses servem de subsídio para a seleção de um método instrucional e, conseguinte, as práticas, atividades, ferramentas e conteúdos requeridos.

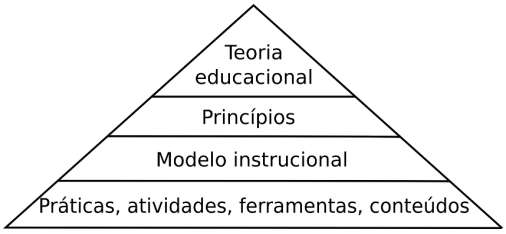


Figura 2.14: Estragégia *top-down* utilizada pelo método ISDMeLO (ALLERT *et al.*, 2001 apud BARUQUE; MELO, 2004).

Ele é baseado no modelo ADDIE, compreendendo as etapas de análise, projeto, desenvolvimento, implementação e avaliação. Na etapa de análise, definem-se o perfil do aprendiz, os principais objetivos de aprendizagem e o contexto em que as atividades serão realizadas. Com base nesses dados, inicia-se uma busca prévia por objetos de aprendizagem.

A etapa de projeto (*design*) consiste no refinamento dos objetivos de aprendizagem, analisando-se as tarefas e os conhecimentos necessários para realizá-las. As tarefas e conhecimentos são estruturados em uma árvore, a qual se designam objetos de aprendizagem.

A sequência de utilização dos objetos de aprendizagem é estabelecida conforme um método instrucional (escolhido de acordo com a teoria educacional mais adequada para o contexto em questão). Caso o objeto de aprendizagem não tenha seus metadados definidos, deve-se defini-los e atualizar os seus dados (para o caso do reúso em cenários distintos daquele originalmente previsto). O método prossegue com o projeto da interface humano-computador, compreendendo a implementação da árvore de tarefas, definição de metáforas, escolha de elementos gráficos, desenho da interface. Protótipos da interface devem ser gerados e avaliados, corrigindo os erros pela execução iterativa da etapa de projeto.

Depois de projetado o objeto de aprendizagem, deve-se buscar, reutilizar ou criar novos objetos de aprendizagem que atendam aos requisitos estabelecidos pelo projeto. Atividades de garantia de qualidade devem ser executadas e, após aprovação, os objetos devem ser disponibilizados em um repositório.

A implementação do objeto de aprendizagem dá-se pela integração dos objetos em um ambiente virtual de aprendizagem, a configuração do modo de entrega e o registro do uso do objeto de aprendizagem. Avaliando-se o progresso dos alunos em relação aos objetivos estabelecidos no início do método, é possível alterar os objetos de aprendizagem ou até mesmo substituí-los por outros mais adequados.

2.3.2.3. RIVED

O RIVED (Rede Internacional Virtual de Educação) é um projeto latino americano cujo objetivo é a produção de conteúdos didáticos, compostos de simulações e animações na forma de objetos de aprendizado e adotando novas abordagens pedagógicas (NASCIMENTO; MORGADO, 2003). No contexto desse projeto, diversos objetos de aprendizagem foram desenvolvidos e um processo, organizado em seis fases, foi elaborado (AMARAL *et al.*, 2006):

- Fase 1: Compreende o projeto instrucional e o projeto dos módulos. Nesta fase, ocorre a análise e descrição dos objetivos e das atividades dos objetos de aprendizagem. Essa descrição é organizada em um documento denominado *Projeto pedagógico* (também denominado de *General design*).
- Fase 2: Interação entre as diferentes equipes para a crítica do projeto pedagógico.
- Fase 3: Especialistas de conteúdo revisam o projeto pedagógico a partir das críticas e descrevem as especificações para cada objeto de aprendizagem, gerando o documento *Roteiro de atividades*.
- Fase 4: Produção dos objetos de aprendizagem.
- Fase 5: Criação do guia de utilização, denominado de *Guia do professor*, para cada objeto de aprendizagem.
- Fase 6: Validação o objeto de aprendizagem pelos usuários.
- Fase 7: Organização e disponibilização dos objetos na Web.

O processo é executado de forma iterativa, permitindo-se a interação entre etapas vizinhas, conforme ilustrado na Figura 2.15. Dessa forma, é possível assegurar a evolução do objeto de aprendizagem e a qualidade/aceitação pelos usuários. A primeira fase corresponde aos processos de projeto instrucional (*Instructional design*) e projeto do objeto de aprendizagem (*Design*). As demais fases são equivalentes, respectivamente, aos processos de *Team interaction*, *storyboard definition*, *development*, *documentation*, *validation* e *deployment*.

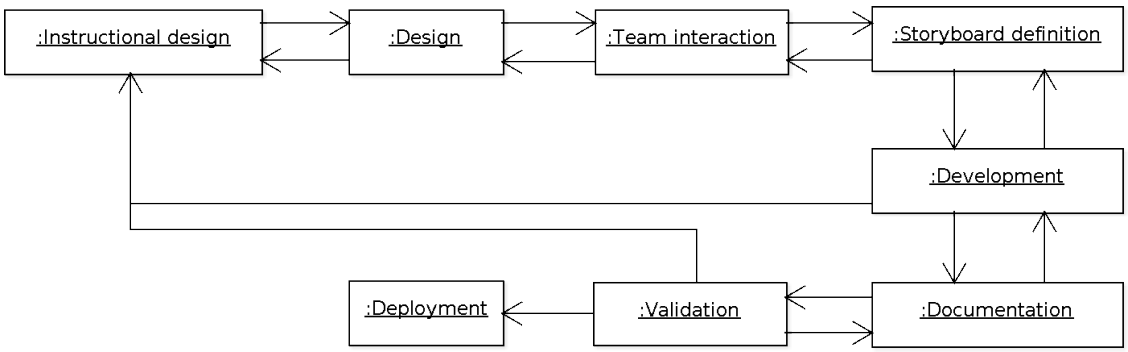


Figura 2.15: RIVED: Processo de desenvolvimento de objetos de aprendizagem. Adaptado de Amaral *et al.* (2006).

2.3.2.4. Sophia

O Processo Sophia define fases, atividades, artefatos e pessoas responsáveis pelo desenvolvimento de objetos de aprendizagem (PESSOA; BENITTI, 2008). Em uma visão macro, ele é organizado em três fases, executadas iterativamente: projeto, desenvolvimento e distribuição.

A fase de projeto estabelece que o coordenador, responsável pela criação do objeto de aprendizagem, define os objetivos educacionais e o plano pedagógico do objeto. O conteúdo necessário para atender aos objetivos, de acordo com o plano pedagógico, é preparado pelo professor. De posse do conteúdo, o coordenador define a estrutura básica do objeto de aprendizagem e aloca os recursos necessários para o seu desenvolvimento.

A fase de desenvolvimento consiste na engenharia do objeto de aprendizagem. Os principais atores envolvidos são o programador e o projetista gráfico (*designer*): o primeiro é responsável pela implementação dos elementos técnicos; o segundo pela elaboração dos elementos gráficos. O objeto de aprendizagem resultante é revisado pelo professor, que define ajustes, nos elementos técnicos e gráficos, a serem executados pelo programador e projetista.

A última fase do Processo Sophia é a distribuição. Ela consiste no empacotamento do objeto de aprendizagem em um formato padronizado (no caso, o SCORM) e na disponibilização do objeto no repositório. Compõe ainda essa fase a avaliação qualitativa do objeto de aprendizado pelos alunos e tutores. A partir dos resultados dessa avaliação, estabelece-se uma nova iteração do desenvolvimento do objeto de aprendizagem.

2.3.2.5. MISA

O processo *Méthode d'ingénierie de systèmes d'apprentissage* (MISA) é um processo de desenvolvimento de objeto de aprendizagem que implementa abordagem instrucional de resolução de problemas (PAQUETTE, 2001; PAQUETTE *et al.*, 2011). Ele se organiza em seis fases: (1) definição de problema, (2) definição de solução preliminar, (3) construção da arquitetura do sistema de aprendizagem, (4) projeto do material instrucional (objetos de aprendizagem), (5) desenvolvimento e validação dos objetos de aprendizagem e (6) implantação deles no sistema de aprendizagem.

Além da organização em fases, o processo define, perpendicularmente a elas, quatro aspectos: (1) conhecimento, (2) instrução, (3) objetos de aprendizagem e (4) implantação. O primeiro define a representação gráfica dos elementos do domínio de conhecimento, representando-os como fatos, conceitos, procedimentos e princípios inter-relacionados. Subconjuntos deste domínio são relacionados a objetivos de aprendizagem e pré-requisitos. O aspecto instrucional define uma rede de eventos e atividades associados aos objetivos e pré-requisitos provenientes do aspecto de conhecimento. Os recursos utilizados nas atividades são representados no aspecto de objetos de aprendizagem. A implantação consiste na configuração do ambiente para a execução da atividade de aprendizagem (PAQUETTE *et al.*, 2005).

Associado ao processo MISA, define-se uma linguagem de modelagem educacional denominada MOT+. Ela representa um modelo instrucional com uma rede de atividades de aprendizagem (*learning events* na nomenclatura adotada pelo método), recursos e regras ou papéis, todos interligados por relacionamentos tipificados (PAQUETTE, 2004). O principal elemento do modelo constitui-se em atividades de aprendizagem. Uma atividade de aprendizagem pode ser composta por unidades de aprendizagem: a composição é representada por um relacionamento do tipo `C` (Composição). Relações de precedência entre atividades são representadas pelo relacionamento do tipo `P` (precedência). Os recursos necessários para ou resultantes da execução de uma atividade são identificados pelo relacionamento do tipo `P/I`. As regras ou papéis são associadas às atividades por relacionamentos do tipo `R` (Regra).

O modelo definido pelo MOT+ é compatível com aquele definido no IMS LD: uma unidade de aprendizagem do IMS LD (UoD) corresponde a uma atividade de aprendizagem/ato ou método. Para as demais subdivisões do método, são criadas outras atividades, constituindo uma hierarquia a qual se associam outras atividades. Para as atividades que compõem um ato, são atribuídos papéis que conduzirão a atividade e descrição do ambiente (objetos de aprendizagem e demais recursos) necessários ou produzidos durante a sua execução. A sequência de atividades é definida pelas atividades estruturantes e relacionamentos de precedência entre as atividades associadas à estrutura.

2.4. Considerações finais

Neste capítulo, os principais conceitos referentes a objetos de aprendizagem foram abordados. Na perspectiva de educação, apresentaram-se as características da educação e do processo instrucional, com atenção ao projeto instrucional como instrumento para sistematizar o desenvolvimento de atividades de aprendizagem considerando desde a definição dos requisitos, passando pelo planejamento da instrução, desenvolvimento de material didático, a instrução e a avaliação.

Nesse contexto, objetos de aprendizagem são apresentados como elementos didáticos reutilizáveis no contexto de educação com informática. A definição de metadados e a de projetos instrucionais permitem a localização, recuperação e alteração dos objetos, adaptando-os de acordo com as especificidades de cada contexto educacional. Todavia, o desenvolvimento desses objetos não é uma atividade trivial, sendo necessária a aplicação de técnicas apropriadas para se obter os benefícios dos objetos de aprendizagem.

O primeiro conjunto de abordagens retrata um objeto de aprendizagem como um material didático definido como hipertextos. Essa visão é apropriada quando considerados os primórdios da Web, em que a preocupação era disponibilizar conteúdos para amplo acesso e com qualidade. Modelos de hipertexto baseados em modelos formais permitem assegurar a consistência dos documentos e a definição dos comportamentos de navegação entre os objetos interligados.

No entanto, a partir de um determinado momento, a grande quantidade de informações disponíveis torna inviável a recuperação com precisão de objetos apropriados e, portanto, o reúso. Metadados amenizam essa condição ao descrever os objetos de aprendizagem e os contextos em que eles foram aplicados, facilitando a indexação e a disponibilização de objetos de aprendizagem em mecanismos especializados/repositórios.

Paralelamente à definição de metadados, a necessidade de integrar objetos de aprendizagem a ambientes virtuais de aprendizagem na Web incorre na definição de dispositivos para o empacotamento dos objetos de aprendizagem e para a troca de informações com o ambiente virtual. Padrões para representação são definidos pela indústria, em cooperação com a academia, e amplamente adotados.

A contínua adoção de objetos de aprendizagem requer não apenas mecanismos para recuperação, mas para reúso e adaptação. O projeto instrucional contribui para essas metas ao estabelecer a forma com que os objetos de aprendizagem serão utilizados nas atividades de aprendizagem.

A representação do projeto instrucional, na forma do sequenciamento e adaptação dos objetos de aprendizagem, é tratada pelas abordagens posteriores àquelas de hipertexto. De fato, mantém-se algumas características, tal como a utilização de mapas conceituais, mas se adotam modelos que expressam com mais precisão ou de forma mais sucinta e escalável o

projeto instrucional. Em uma visão mais próxima da perspectiva educacional, linguagens de modelagem educacional permitem a especificação de atividades de aprendizagem (OUEML, IMS LD, PoEML). Em contrapartida ao nível conceitual mais abstrato, perde-se o rigor do formalismo existente na abordagem anterior (baseada em hipertextos). Abordagens que mantêm a utilização de técnicas de especificação formal não sofrem esse revés, permitindo a representação, com precisão, das sequências de utilização e interação dos objetos de aprendizagem (Redes de Petri, FSM). No entanto, elas se apresentam distantes da perspectiva educacional, aumentando o risco da especificação formal de um projeto instrucional que não corresponde aos requisitos educacionais. Os métodos de desenvolvimento baseado em modelos permitem a combinação de ambas perspectivas, definindo modelos apropriados para a representação de elementos educacionais e computacionais e regras de transformação entre esses modelos.

Além dos métodos, a definição de processos que conciliem o desenvolvimento da instrução com o dos objetos de aprendizagem, acomodando o resultado das avaliações do processo instrucional para o aperfeiçoamento dos objetos de aprendizagem, em um ciclo de vida iterativo, é desejável. Os processos descritos na literatura compartilham dessa característica, variando quanto às atividades exigidas e o alinhamento com métodos específicos de desenvolvimento de objetos de aprendizagem (tal como o SP-DEM e o MISA).

Observando-se as perspectivas educacionais e computacionais apresentadas neste capítulo, é possível definir um conjunto de requisitos desejável para abordagens de desenvolvimento de objetos de aprendizagem. No Capítulo 4 são definidos os requisitos dessas perspectivas, bem como quanto ao caráter multimídia e interativo de objetos de aprendizagem, tratados no próximo capítulo.

Capítulo 3

Objetos de aprendizagem multimídias e interativos para televisão digital

Um fato estabelecido em educação é a importância da interação entre aprendizes e professores e com o material didático, na forma de objetos de aprendizagem, e ambientes virtuais de aprendizagem para a realização das atividades de aprendizagem estabelecidas pelo projeto instrucional (CAO *et al.*, 2008). A interatividade dos objetos de aprendizagem deve contribuir para esse aprendizado, permitindo e facilitando as interações.

A disponibilização de objetos de aprendizagem interativos nos diferentes meios de comunicação e computação disponíveis contribuem para isso. Afinal, educação não precisa, necessariamente, ocorrer em um local específico: ela pode ser realizada em um ambiente formal, como é com a educação de ensino básico, médio e superior; no ambiente de trabalho (treinamentos); ou em situações informais, para fins recreativos ou para resolução de problemas cotidianos.

A transformação da Internet em meio educacional já é uma realidade, mas existem espaços que ainda não foram devidamente explorados, tal como a televisão digital. Atualmente, o alcance da televisão é superior ao da Internet e computadores como um todo, estabelecendo-se em aproximadamente 97,2% dos lares brasileiros (IBGE, 2009). Além disso, eles podem ser vistos como complementares, demonstrando uma oportunidade para o desenvolvimento de novas soluções para educação nesse contexto. Neste trabalho, escolheu-se como foco a televisão digital.

A televisão digital (TVD) é um serviço que agrega mídias de alta resolução (vídeo e áudio) em programas interativos. Em relação ao sistema tradicional de televisão (analógico), seu mais importante diferencial é interatividade (AARRENIEMI-JOKIPELTO, 2006), possibilitada pela plataforma para execução de aplicativos (*middleware*) e pelo canal de retorno. Além disso, a TVD não se restringe aos televisores tradicionais, compreendendo também dispositivos portáteis, tais como celulares e *tablets*. Atualmente 83,1% dos domicílios brasileiros possuem ao menos uma pessoa com celular (IBGE, 2009), atestando o potencial de uso deles tanto para acesso direto à TVD como para o estabelecimento de um canal de retorno. Aliado a esses fatores – alcance e capacidade multimídia e interativa – um dos motes da TVD é a disponibilização de programas educativos ou *t-learning* (BATES, 2003), provendo mecanismos para permitir o aprendizado em qualquer momento, integrando ensino formal e informal (DAMáSIO, 2003).

Diversos aspectos, aplicáveis à educação e ao desenvolvimento de objetos de aprendizagem, foram estudados nos Capítulo 2. Neste capítulo, discute-se a interatividade de objetos de aprendizagem e a televisão digital como plataforma para sua disponibilização e execução. Na Seção 3.1, descrevem-se as principais características da TVD e, em especial, das soluções adotadas para o sistema brasileiro. Seguinte à apresentação dessa plataforma, discute-se na Seção 3.2 a interatividade de objetos de aprendizagem como um todo e na Seção 3.3 no contexto da televisão digital.

3.1. Televisão digital

As principais características do sistema de TVD são a qualidade de som e imagem e a interatividade proporcionada pela execução de aplicações e a comunicação com outros dispositivos pelo canal de retorno. No entanto, embora pareçam recentes, esses atributos são alvos de pesquisas e empreendimentos da área de televisão desde os seus primórdios.

Uma das raízes da televisão digital está nos estudos desenvolvidos na década de 60 (século XX) pelo NHK, grupo de televisão do Japão, sobre as limitações do sistema analógico NTSC. Eles concluíram que o sistema em voga na época não aproveitava completa e efetivamente a capacidade visual humana (SGRIGNOLI, 2007). Desenvolveram então o primeiro sistema de televisão de alta resolução, com o quádruplo de resolução do sistema anterior, porém ainda analógico. A interatividade em televisão também é anterior ao sistema digital e até mesmo à alta definição. O primeiro programa interativo data de 1953: o *Winky Dink and You*. Com uma folha transparente especial (presa eletroestaticamente na tela da televisão) e lápis de cera, crianças eram incentivadas a desenhar sobre a imagem com o intuito de criar elementos que ajudassem os personagens do programa (CAREY, 1996). Logo após, iniciou-se a utilização de telefones, com a participação dos telespectadores, em uma primitiva demonstração de canal de retorno. Em 1970, começou o uso, na Inglaterra, do serviço de teletexto para enviar dados tais

como notícias, previsões climáticas, programações das emissoras e legendas. Posteriormente, começou a implantação de vídeos sob demanda e canais de compra, utilizando aparelhos específicos para este fim (STB) e com plataformas proprietárias de programação.

Em suma, a qualidade e interatividade não são exclusividades da TVD. O seu principal diferencial é que, com a digitalização e o estabelecimento de padrões abertos, viabilizou-se a criação de uma plataforma para conteúdos e programas interativos disponibilizáveis em larga escala. Três sistemas foram inicialmente definidos: o americano (ATSC), o europeu (DVB) e o japonês (ISDB). Recentemente, foram criados os padrões chinês (DTMB) e uma versão do padrão japonês aperfeiçoada pelo Brasil: o SBTVD, também conhecido internacionalmente como ISDB-Tb.

Embora esses padrões tenham suas particularidades, eles possuem uma arquitetura em comum, conforme exposto na Figura 3.1 e, inclusive, adotam as mesmas soluções para algumas das camadas, tais como sistema de transporte e multiplexação. Analisando a arquitetura sob o ponto de vista inicial do receptor, existe o meio de transmissão do sinal da televisão digital: terrestre, cabo ou satélite. Cada meio tem sua particularidade e restrições (faixa do espectro eletromagnético, tipo de erros aos quais a transmissão está sujeita, condições de recepção do sinal), requerendo sistemas de transmissão e codificação de sinais apropriados. Em função desse sistema de transmissão, define-se: (1) a robustez do sinal e o sucesso de sua recepção em dispositivos fixos e móveis, (2) a capacidade de transmissão de dados por canal, (3) custo da transmissão e (4) custo do receptor. No âmbito dos meios de transmissão por satélite e por cabo, essas questões são mais simples de equacionar. Entretanto, no caso da transmissão terrestre, equilibrar todos os fatores é uma tarefa delicada. Nesse caso, cada sistema possui sua particularidade: alguns privilegiam o baixo custo (*Advanced Television Systems Committee* (ATSC)) enquanto outros enfocam na robustez e capacidade (*Integrated Service Digital Broadcasting* (ISDB), SBTVD e *Digital Terrestrial Multimedia Broadcast* (DTMB)). Cabe destacar que estes foram os últimos padrões desenvolvidos, favorecendo-se da redução

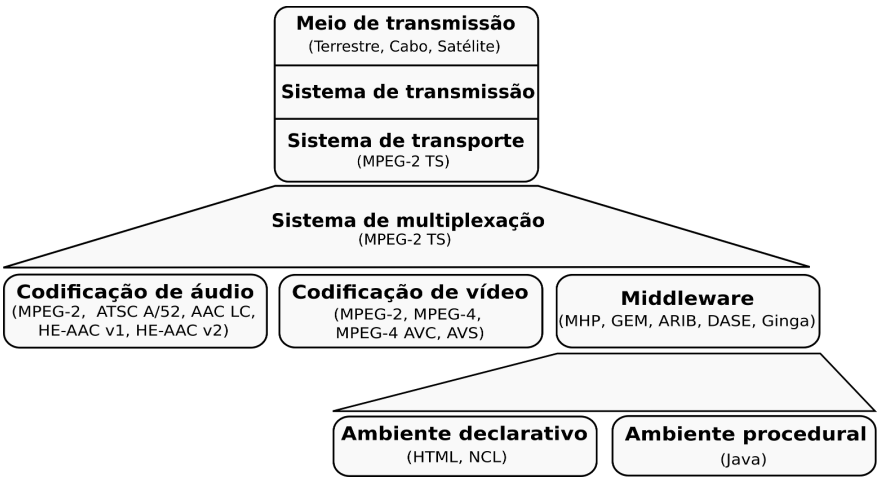


Figura 3.1: Camadas de um sistema de televisão digital.

do custo do *hardware* necessário para implementar soluções sofisticadas de codificação de sinal (CARVALHO *et al.*, 2007).

Ainda que o sistema de transmissão seja um componente essencial e complexo dos sistemas de TVD, do ponto de vista do desenvolvedor de programas e de objetos de aprendizagem, as informações mais importantes a serem lembradas dessa camada são os tipos de dispositivos que podem receber o sinal (fixos ou móveis), a quantidade de dados transmitida (largura de banda) e o modo de transmissão de dados em difusão.

Além da transmissão da emissora de televisão em direção aos usuários, também ocorre a comunicação deles com a emissora e com outros usuários pelo canal de retorno. Embora muito se tenha discutido sobre esse mecanismo no SBTVD, não existe uma solução consolidada. Assim, admite-se que três situações devem ser previstas: a ausência do canal do retorno ou a presença de um canal de retorno de forma intermitente ou plena. A intermitência é representada por uma conexão estabelecida sob demanda (por linha telefônica ou celular) e a plena por uma conexão permanente, ambas com a Internet.

Retomando a explicação do sistema digital a partir da camada de transmissão, existe a camada de transporte. Em todos os padrões, adota-se o sistema de transporte MPEG-2 (ISO/IEC, 2007). Ele define uma sequência de pacotes que formam fluxos elementares de vídeo, áudio ou dados, que são associados aos programas em transmissão conforme definido nas tabelas de dados PSI transmitidas.

Os principais elementos de um programa de televisão interativos são os fluxos de vídeo, áudio e os aplicativos. O primeiro, e mais importante em se tratando de televisão, é o vídeo. Técnicas de compressão são utilizadas para utilizar eficientemente a largura de banda disponível e permitir a transmissão de mídias em alta resolução ou, no caso de multiprogramação, vários vídeos de baixa resolução. A resolução máxima é de 1920x1080 (ou, nos termos televisivos, 1080 linhas) no modo entrelaçado e 1280x720 de forma progressiva. Para programas antigos (anteriores à digitalização), utiliza-se a resolução SDTV (720x480), que corresponde à resolução máxima dos sistemas analógicos (NTSC e PAL). A menor resolução, de 320x240, é utilizada na transmissão de vídeo para dispositivos móveis.

Além da resolução, existe a escolha do algoritmo de compactação do vídeo. Os sistemas mais antigos (ATSC e *Digital Video Broadcasting* (DVB)) utilizam os padrões MPEG-2/H.262 (ISO/IEC, 2000) e MPEG-4 (ISO/IEC, 2004). Sistemas mais recentes adotam métodos mais eficientes de compressão, em especial o MPEG-4 AVC/H.264 (ISO/IEC, 2010), escolhido pelo SBTVD, e o AVS, pelo chinês DTMB (YU *et al.*, 2009). Em termos práticos, para uma mesma qualidade de vídeo, o MPEG-4 AVC ou o AVS precisam da metade da quantidade de bytes necessários para o MPEG-4 (e ainda menos em relação ao MPEG-2).

Semelhantemente, a digitalização também permitiu a distribuição de áudio de alta qualidade, com frequências de amostragem de até 48 kHz, quantificação de 16 bits e formatos

estéreo ou 5.1 (cinco canais normais e um para sons de baixa frequência). Quanto à escolha dos algoritmos de compressão de som, a situação é semelhante àquela observada nos vídeos: os sistemas mais antigos utilizam o MPEG-2 AAC (ISO/IEC, 2006) para som estéreo ou o ATSC A/52 (ATSC, 2010) para áudio no formato 5.1. Os sistemas mais recentes utilizam o MPEG-4 AAC nos perfis de baixa complexidade AAC-LC (ISO/IEC, 1993) e de alta eficiência AAC-HEv1 e AAC-HEv2 (ISO/IEC, 2009), com taxas de compressão e qualidade superiores.

Embora a qualidade da imagem e áudio da televisão digital seja essencial para proporcionar aos usuários uma experiência diferenciada, ela não é suficiente. Por parte das produtoras e geradoras de programas de televisão, é necessário diminuir a distância entre a propaganda e a compra de produtos e serviços. Em termos de políticas públicas, a oferta de serviços à comunidade de forma ágil pela televisão também é vista como essencial. Por fim, para os usuários, apenas "assistir" televisão não basta: eles desejam controlar, compartilhar e criar novos conteúdos.

A interatividade dos sistemas digitais de televisão vai de encontro a esses ensejos, aproveitando-se do sistema embarcado indispensável à TVD para a definição de uma plataforma para a execução de aplicativos interativos. O último item da camada trata desse aspecto. Ele é representado pelo *middleware*, que é encarregado de oferecer um ambiente para a execução de aplicações interativas, isolando-as das particularidades do *hardware* e do sistema operacional utilizados pelo receptor. As principais funções do *middleware* são: controlar o ciclo de vida das aplicações; gerenciar a execução de múltiplas aplicações e a alocação de recursos; prover mecanismos para comunicação entre aplicações; permitir o acesso aos fluxos elementares e ao canal de retorno; decodificar e reproduzir mídias (vídeo, áudio e imagens); armazenar e recuperar dados (persistência); tratar eventos e elementos multimídia da interface com o usuário (LEITE *et al.*, 2005). Dentre suas funções, destacam-se o acesso às funcionalidades básicas do receptor e específicas de televisão digital (sintonizador de canal, decodificação e reprodução de áudio e vídeo), acesso a dispositivos de entrada (controle remoto) e bibliotecas para o desenho de interfaces gráficas.

O *middleware* nos sistemas de televisão digital permite a execução de dois tipos de aplicação: procedurais e declarativas. As aplicações procedurais correspondem a programas de computador formados por comandos imperativos (LEITE *et al.*, 2005), em geral aplicações em Java. As aplicações declarativas são hipertextos interpretados pelo receptor, permitindo a navegação, o acesso a elementos multimídia e a programação em função de eventos gerados pelo sistema ou pela interação do usuário. Os hipertextos podem ser documentos HTML (ou variações desse) ou, no caso do SBTVD, documentos *Nested Context Language* (NCL).

Expostas as características comuns dos sistemas de televisão digital, foca-se no padrão definido pelo Brasil, o SBTVD, alvo dos objetos de aprendizagem interativos desenvolvidos pela abordagem definida nesta tese. Cabe ressaltar que essa escolha não é arbitrária: o

sistema brasileiro adota tecnologias de vanguarda em todas as camadas, principalmente quanto ao *middleware*, que é compatível com o *Globally Extensible MHP* (GEM) (utilizado pelo DVB e ATSC), e oferece uma plataforma de programação declarativa diferenciada, que permite o sincronismo entre mídias e a interação do usuário.

3.1.1. Sistema Brasileiro de Televisão Digital

O Sistema Brasileiro de Televisão Digital (SBTVD) adota, para as camadas de transmissão e transporte terrestre, os mesmos padrões do padrão japonês ISDB: OFDM para transmissões terrestres (ABNT, 2008a) e MPEG-2 para transporte e multiplexação dos fluxos de vídeos e dados (ABNT, 2008b). As diferenças ocorrem nos métodos de compressão de vídeo e de áudio e no *middleware*.

A escolha brasileira foi pela codificação de vídeos com MPEG-4 AVC/H.264, permitindo o oferecimento de vídeos na resolução de 1280x760 a 60 quadros/s e de 1920x1080 a 30 quadros/s. A configuração do fluxo de áudio pode ser transmitida em estéreo (dois canais) para dispositivos fixos e móveis ou em 5.1 (cinco canais e um canal específico para som de baixa frequência) para dispositivos fixos. O método de compressão é o MPEG4 AAC-LC ou AAC-HEv1 para dispositivos fixos e AAC-HEv2 para dispositivos móveis (ABNT, 2011a). Em todos os casos, a taxa de amostragem é de 48 kHz.

O *middleware* do SBTVD é o principal diferencial do padrão brasileiro. Ele oferece dois ambientes de execução, tal como os demais padrões: um voltado para programas procedurais e outro para declarativos. Para programas procedurais, suporta-se a linguagem Java, constituindo o ambiente Ginga-J. Para programas declarativos, ao invés de adotar uma proposta similar aos demais padrões (baseada em HTML), adotou-se uma linguagem de especificação de hipertextos denominada NCL e pequenos programas (*scripts*) procedimentais especificados em Lua, definindo-se o ambiente Ginga-NCL. Juntos, eles constituem o Ginga, o *middleware* do SBTVD. Ele é organizado de acordo com a Figura 3.2: acima do sistema operacional do receptor, existe um núcleo comum ao Ginga-J e ao Ginga-NCL e um conjunto de serviços específicos do SBTVD. Os componentes correspondentes ao Ginga-NCL encontram-se

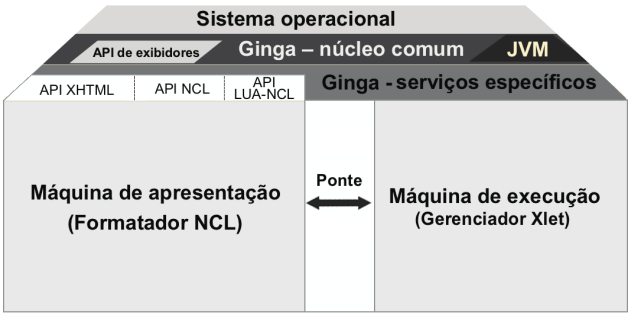


Figura 3.2: Arquitetura do Ginga (ABNT, 2011a, p. 10).

à esquerda e ao Ginga-J à direita. Uma aplicação interativa pode ser escrita exclusivamente em NCL, Java ou uma combinação de ambas, nesse caso utilizando a ponte existente entre os subsistemas Ginga-J e Ginga-NCL.

A arquitetura do Ginga implementa o modelo de referência estabelecido pela recomendação ITU J.200 (ITU, 2010a), que estabelece um núcleo comum de *middlewares* para aplicações declarativas e procedimentais. As funcionalidades implementadas compreendem a sintonia de canais, a recuperação de dados transportados nos fluxos elementares e do carrossel de dados, a decodificação e a apresentação de conteúdos multimídia (imagens, vídeos e sons), persistência de dados, controle de acesso, comunicação pelo canal de retorno, gerenciamento de aplicações, bibliotecas de interface gráfica, dentre outras, conforme ilustrado na Figura 3.3.

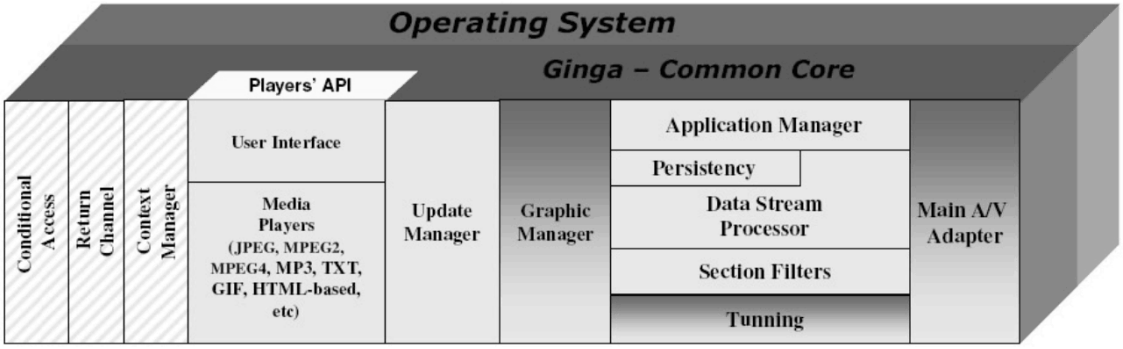


Figura 3.3: Núcleo do Ginga e partes comuns aos subsistemas Ginga-J e Ginga-NCL (FILHO *et al.*, 2007).

3.1.2. Plataforma procedimental e Ginga-J

A plataforma para programas procedimentais do SBTVD é definida pelo Ginga-J (ABNT, 2010a, 2011b), um *middleware* que define as *Application Program Interface* (API), formato de dados e protocolos para as aplicações escritas em Java (FILHO *et al.*, 2007). O Ginga-J foi definido para atender aos requisitos idealizados pelo governo brasileiro e, paralelamente, ser compatível com os padrões existentes definidos pela ITU (ITU, 2010b) e com o arcabouço de *middleware* adotado pelos demais sistemas de televisão digital, o GEM (DVB, 2011).

Um componente chave do Ginga-J (e de outros *middlewares* procedimentais) é a máquina virtual Java e as bibliotecas Java. Nesse aspecto, em particular, o sistema de televisão digital herda características do Java para dispositivos móveis (J2ME): máquina virtual para sistemas móveis e embarcados, compatível com subconjuntos das instruções das máquinas virtuais Java modernas (Java 1.3.x) e da biblioteca padrão do Java. Dessa forma, o Ginga-J segue a configuração *Java Connected Device Configuration* (CDC) (JSR, 2006c) e os perfis *Java Foundation Profile* (FP) (JSR, 2005), *Java Personal Basis Profile* (PBP) (JSR, 2006b) e *Java Personal Profile* (PP) (JSR, 2006a), que constituem a base dos componentes de sua

API, exposta na Figura 3.4.

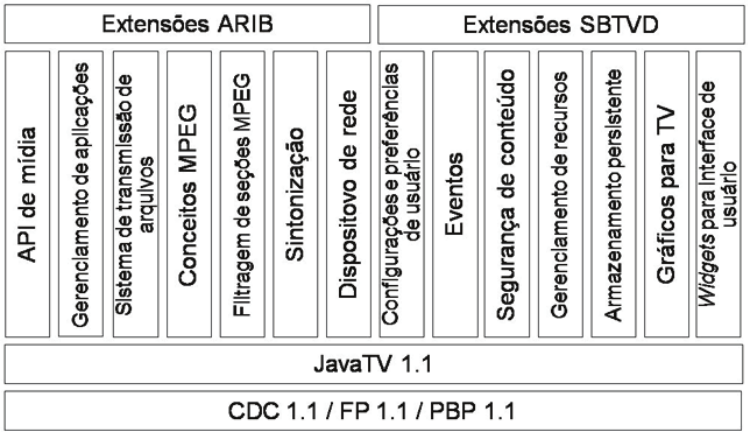


Figura 3.4: Componentes da API Java do Ginga-J.

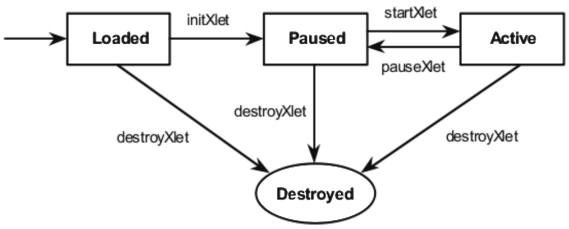
Além dos perfis FP, PBP e PP, o Ginga-J implementa um conjunto de API específico para o domínio de televisão digital. A proposta inicial do SBTVD previa a compatibilidade do Ginga-J com o GEM (DVB, 2011), o qual é utilizado nos demais sistemas de televisão digital. Posteriormente, tal decisão foi revogada, definindo-se, em cooperação com a empresa *Sun/Oracle* (criadora do Java), o padrão JavaDTV (ABNT, 2010b) em substituição ao GEM. Observando-se a Figura 3.4, identifica-se a organização final dos componentes do Ginga-J. Logo acima dos componentes referentes aos perfis da configuração CDC (FP, PBP e PP), encontra-se o JavaTV (JSR, 2008), uma API para a criação de aplicações de televisão digital interoperáveis (compatíveis com o GEM). Acima dele encontra-se um conjunto de interfaces e serviços para TVD: trata-se do JavaDTV. Ele substitui as API HAVi e DAVIC, utilizadas no GEM, por novas propostas de API e com a biblioteca *Java LightWeight User Interface Toolkit* (LWUIT). Finalmente, APIs específicas aos sistemas ISDB e SBTVD são definidas para complementar os serviços disponibilizados pelo JavaDTV.

Uma aplicação de televisão digital é definida como uma implementação da interface `javax.microedition.xlet.Xlet` (ou simplesmente `Xlet`) do pacote JavaTV, tal como exemplicado na Figura 3.5a. Um `Xlet` possui um ciclo de vida bem definido, conforme apresentado na Figura 3.5b. Ele inicia no estado `Loaded`. Desse estado, ele pode ser iniciado, momento em que aloca os recursos necessários para sua execução, mas sem iniciar qualquer atividade e interação. Apenas depois de autorizada a sua operação é que ele se torna ativo. Após realizar as operações necessárias ou de acordo com o determinado pelo gerenciador de aplicações, um *Xlet* pode ser interrompido. Além disso, ele pode ser destruído em qualquer estado (liberando recursos alocados e sendo descarregado da memória).

Além disso, cada aplicação para a plaforma Ginga-J deve atender às restrições e recomendações descritas nas normas ABNT 15606-4, 15606-6 e 15606-8 (ABNT, 2010a, 2010b, 2011b). As normas em questão estabelecem, além das restrições de operação, as

```
import javax.microedition.xlet.*;

public class Example implements Xlet
{
  public Example() {}

  public void initXlet(XletContext context)
    throws XletStateChangeException {}

  public void startXlet()
    throws XletStateChangeException {}

  public void pauseXlet() {}

  public void destroyXlet(boolean uncond)
    throws XletStateChangeException {}
}
```

(a) Exemplo mínimo de Xlet.



(b) Ciclo de vida de um Xlet (ABNT, 2010a).

interfaces para gerenciamento de ciclo de vida, comunicação com aplicações NCL, acesso e uso de planos gráficos, canal de interatividade, persistência, sintonização de canais, entre outras questões.

3.1.3. Plataforma declarativa e Ginga-NCL

A plataforma declarativa do SBTVD foi concebida para a execução de aplicações voltadas para o domínio de vídeos não-lineares, caracterizado pela sincronização das ações com o tempo e espaço de fluxos de áudio e vídeo (SOARES *et al.*, 2006). Enquanto os ambientes procedurais enfatizam os comandos dados pelo usuário para dirigir a interação, os ambientes declarativos utilizam os eventos de sincronização como base, sendo, nesse caso, as interações dos usuários apenas mais um tipo de sincronização a ser considerado.

O Ginga-NCL é o subsistema responsável pela execução de conteúdo declarativo, cujo principal componente é um formatador (componente de apresentação) para NCL (ABNT, 2011a). Conforme apresentado na Figura 3.5, ele é complementado por formatadores de HTML e CSS e por mecanismos de execução de programas em ECMAScript e Lua (utilizados nos conteúdos declarativos para a programação de comportamentos associados à sincronização).

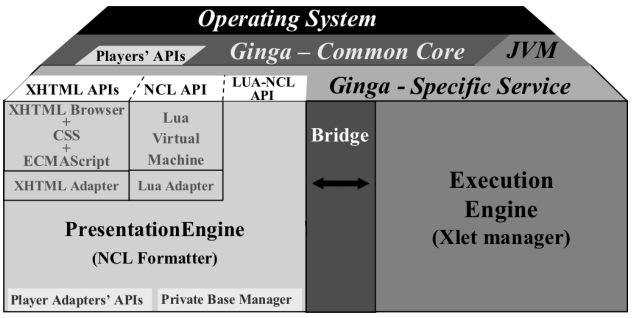


Figura 3.5: Componentes do subsistema Ginga-NCL (SOARES *et al.*, 2007).

O *middleware* declarativo brasileiro possui raízes no modelo *Nested Context Model* (NCM) (CASANOVA *et al.*, 1991). Esse modelo organiza o conteúdo declarativo em contextos sucessivamente aninhados, aos quais é possível o acesso por portas. A conexão entre os contextos dá-se por ligações (elos) entre as portas, às quais é possível definir condições e ações a serem executadas pelos contextos conectados.

A linguagem NCL utiliza o modelo NCM e permite, além da especificação dos contextos aninhados, a utilização de pequenos programas procedimentais (*scripts*) para realizar operações que não são suportadas no ambiente declarativo. Assim, documentos NCL definem uma aplicação XML com facilidades para a especificação de aspectos de interatividade, sincronismo espaço-temporal de objetos de mídia, adaptabilidade e suporte a múltiplos dispositivos (SOARES *et al.*, 2007).

O elemento principal de uma aplicação NCL é o contexto, representado na Figura 3.6 por um vértice (`Node`). Um contexto é definido por um vértice do tipo mídia (`Media node`) ou por um vértice de composição (`Composition`) que agrega outros vértices (sejam eles de mídia ou de composição). Cada contexto possui uma ou mais portas que permitem o acesso ao seu conteúdo. Um nó do tipo mídia possui, associado a ele, uma mídia qualquer (imagem, vídeo, som, texto, *scripts* em Lua ou ECMAScript). A apresentação de uma mídia relacionada a um vértice é definida por um descritor (`Descriptor`). O descritor é incumbido de acionar o reprodutor adequado da mídia (`Media player`) para apresentar o conteúdo em uma região (`Region`) de um dispositivo de saída (`InteractionDevice`).

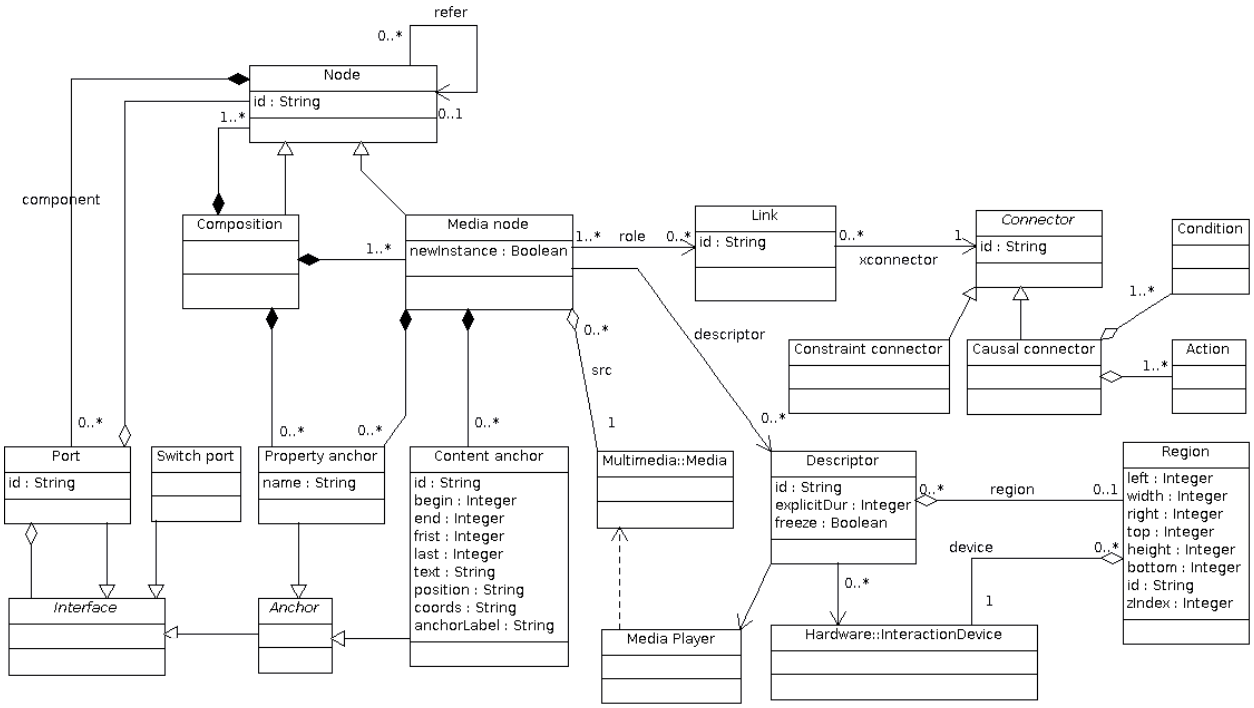


Figura 3.6: Modelo de aninhamento de contextos (NCM) implementado pela NCL.

Em NCL, a sincronização é baseada na estrutura e não na mídia. Isso significa que os vértices devem ser configurados com âncoras, as quais podem ser utilizadas para a especificações de interações na aplicação. Durante a apresentação/execução da mídia contida do vértice, os eventos gerados são comparados aos conectores (`Connector`) associados ao contexto e, se satisfeitas as condições do conector (`Condition`), as ações definidas no conector (`Action`) são executadas. As ações podem ser o acionamento de um contexto ou a alteração de propriedades (ou seja, a aplicação pode ser alterada em tempo de execução).

Observando a estrutura de um documento XML que representa uma aplicação NCL, exemplificado na Código-fonte 3.1, deve-se especificar quatro tipos de informação: o que, onde, como e quando. O primeiro tipo de informação (o que) é representado por nós (contexto e objetos de mídia), o segundo (onde) por regiões, o terceiro (como) por descritores e o último (quando) por elos e conectores. O documento NCL organiza-se em duas partes: o cabeçalho e o corpo. O cabeçalho define as bases para o documento: regiões, descritores e conectores, ou seja, "onde", "como" e parte do "quando". O corpo define os contextos e os objetos de mídia de cada contexto, as portas de acesso aos contextos e os relacionamentos, englobando "o que" e o restante do "quando".

```
<?xml version="1.0" encoding="ISO-8859-1"?>

<ncl id="saberDigital"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns="http://www.ncl.org.br/NCL3.0/EDTVProfile"
    xsi:schemaLocation="http://www.ncl.org.br/NCL3.0/EDTVProfile
␣␣␣␣http://www.ncl.org.br/NCL3.0/profiles/NCL30EDTV.xsd">

<head>
    <regionBase>
            <region id="rExample" left="0" top="0" width="720" height="576" />
    </regionBase>
    <descriptorBase>
            <descriptor id="dExample" region="rExample" />
    </descriptorBase>
</head>

<body>
    <port id="portExample" component="mExample" />
    <media id="mExample" src="example.png" type="image/png" descriptor="dExample" />
</body>
</ncl>
```

Código-fonte 3.1: Exemplo de documento NCL.

No exemplo mostrado na Código-fonte 3.1, é possível observar o elemento `port`. Ele é necessário aos vértices que representam um contexto, que é o caso do contexto inicial da aplicação NCL e que indica a porta de entrada para o vértice `mExample`. Em outras palavras, uma porta indica, para um determinado contexto, quais são os vértices que podem ser ativados ao acessar o contexto. No caso do acesso a um contexto com mais de uma porta

e sem especificar uma porta em específico, o acesso corresponde ao acesso não-determinístico ao vértice, com a escolha aleatória de uma das portas.

Os vértices de um documento NCL também podem ter âncoras. Elas permitem utilizar segmentos ou propriedades do vértice para definir os pontos de entrada no vértice de mídia ou de contexto. Dois tipos de âncora são suportados: de conteúdo (definido pelo elemento `area`) e de propriedade (definido por `property`). Um exemplo típico é a definição de capítulos em um filme, em que a âncora de conteúdo define um segmento da mídia (Código-fonte 3.2).

```
<media type="video" id="video1" src="media/video1.mpg" descriptor="dVideo1">
    <area id="aOpening" begin="0s" end="70s" />
    <area id="aEnding" begin="3600s" end="3700s" />
</media>
```

Código-fonte 3.2: Exemplo de âncora de conteúdo em um documento NCL.

Os elos de um documento NCL são representados por ligações (`link`), as quais estão associadas a um conector (`connector`). Os conectores definem mecanismos de causalidade e restrição e os associam a papéis (`role`). As ligações associam vértices aos papéis estabelecidos pelo conector. Os papéis, em NCL, correspondem a uma ação e uma condição. Os seguintes papéis estão disponíveis para um vértice: as ações `start`, `stop`, `pause`, `resume` e `set`; e as condições `onBegin`, `onEnd`, `onSelection`, `onMouseSelection` e `onKeySelection`. No exemplo da Código-fonte 3.3, define-se um conector causal `onBeginStart` e no elo `linkLectureChoiceStart` utiliza-se esse conector, associando vértices de mídia aos papéis `onBegin` e `start`.

```
<causalConnector id="onBeginStart">
    <simpleCondition role="onBegin" />
    <simpleAction role="start" />
</causalConnector>

<link id="linkLectureChoiceStart" xconnector="composer#onBeginStart">
    <bind role="onBegin" component="mLectureChoiceBackground" />
    <bind role="start" component="mLecture1PreviewPicture" />
</link>
```

Código-fonte 3.3: Exemplo de definição de um conector causal e de um elo deste tipo de conector.

Alguns conectores exigem parâmetros, além da associação de papéis, para o seu uso efetivo. Por exemplo, o conector da Código-fonte 3.4 define o parâmetro `keyCode`. Esse parâmetro é atribuído com o valor da tecla selecionada pelo usuário quando executada a aplicação.

```
<causalConnector id="onKeySelectionStart">
    <connectorParam name="keyCode" />
    <simpleCondition role="onSelection" key="$keyCode" />
```

```
    <simpleAction role="start" />
</causalConnector>
```

Código-fonte 3.4: Exemplo de definição de um conector causal e de um elo deste tipo de conector.

Uma aplicação que faça uso desse conector define, na associação ao papel `onSelection`, o valor da tecla desejada. Por exemplo, caso se deseje criar uma ligação que será executada quando o botão vermelho do controle remoto for selecionado, o código da Código-fonte 3.5 é uma possível solução.

```
<link id="iRedStart" xconnector="onKeySelectionStart">
    <bind component="video1" role="onSelection">
        <bindParam name="keyCode" value="RED" />
    </bind>
    <bind component="video2" role="start" />
</link>
```

Código-fonte 3.5: Exemplo de um elo que define um parâmetro para a associação com o conector.

Outras opções disponíveis para os conectores são a definição de ações e condições compostas e atrasos (*delay*) que podem ser estudadas na norma do NCL (ABNT, 2011a).

3.2. Interatividade em televisão digital

Em televisão digital, a interatividade está associada ao potencial da habilidade de uma mídia permitir que o usuário exerça uma influência no conteúdo ou forma da comunicação (JENSEN, 1998). Esse potencial pode ser definido em função da transmissão, da escolha, da produção e adaptação de conteúdo (JENSEN, 1998), associando-se com diversos fatores: tipo de atividade, dispositivos disponíveis, forma de conexão, fornecedor dos aplicativos de interatividade.

Os dispositivos para a interação com a televisão definem a forma com que os usuários recebem e transmitem informações para a aplicação interativa e, por intermédio dessa, a outros usuários ou consigo mesmo. Em seu nível mais essencial, existem o áudio e a imagem inerentes da própria televisão, constituindo-se no principal, e geralmente único, meio de transmissão de informação para o usuário. Na direção do usuário para a televisão, o principal dispositivo disponível é o controle remoto, responsável por repassar à televisão os eventos de acionamento de teclas. Outras propostas buscam sobrepujar as limitações do controle remoto pela adição de novas funcionalidades aos controles a partir de modelos já existentes, utilização de objetos do dia a dia ou de dispositivos pessoais como celulares e *tablets* (CESAR *et al.*, 2008). Embora vislumbra-se a utilização de outros meios e formas de interação (SILVA *et al.*,

2008; PEDROSA *et al.*, 2011), a maioria das aplicações ainda se restringirá aos controles remotos, apesar das suas limitações quanto à usabilidade e capacidade de representação de ações (CARMICHAEL *et al.*, 2006).

Os tipos de interação dependem das capacidades dos dispositivos disponíveis. Bulterman *et al.* (2006) classificam-as em renderização/apresentação, seleção/navegação e enriquecimento. No caso mais simples, de apresentação, a interação limita-se a ver o conteúdo, sem possibilidades de uma ação explícita com a aplicação. Esse é o caso em que o único dispositivo disponível é a tela da televisão. No caso da presença do controle remoto, a interação se restringe à seleção de opções e escrita de texto com teclados virtuais ou pelo mapeamento do teclado numérico para alfanumérico. A presença de outros dispositivos que oferecem mais formas de entrada de dados, tal como sistemas de captura de áudio (microfone) ou vídeo, permite interações mais ricas e possivelmente mais adequadas à intenção do usuário. Em especial, pode-se combinar diversos tipos de interação, permitindo a escolha da mais adequada de acordo com a situação de uso, condições ou preferências do usuário (VEGA-OLIVEROS *et al.*, 2010). Caso o dispositivo também permita a apresentação de conteúdo, ele se torna apropriado para o enriquecimento e o compartilhamento, colaborando para o aspecto social de TVD (CESAR; CHORIANOPOULOS, 2008).

Além das limitações impostas pelos dispositivos, deve-se considerar o tratamento dado às interações, em especial quanto às dependências temporais e, mais especificamente, a sincronização. Ela pode ser classificada como síncrona ou assíncrona e afeta diretamente a interpretação quando outros usuários participam da interação: a ausência de sincronização, com os efeitos de uma ação sendo percebidos pelas partes envolvidas em um momento diferente das ações realizadas, em uma situação em que tal ordenação seja essencial (tal como uma conversa em tempo real), prejudica a interação; já em uma situação em que existe informação contextual suficiente a cada ação de modo que seja possível inferir o significado independentemente da ordem original (tal como em um fórum de discussão), é razoável a assincronia.

Chorianopoulos (2007) considera a questão de sincronia e a presença dos usuários para a definição da classificação social de uma interação. Em geral, assistir televisão é uma atividade que envolve grupos de pessoas, sejam elas dispostas no mesmo local ou à distância, participando da mesma experiência simultaneamente ou a compartilhando posteriormente (NATHAN *et al.*, 2008). Obviamente, quando no mesmo ambiente, a interação é mais direta e não necessariamente é tratada pela aplicação interativa enquanto que, à distância, o tratamento deve ser diferenciado, permitindo a comunicação síncrona e assíncrona entre os usuários e o compartilhamento e enriquecimento dos conteúdos em questão.

O tipo de interação corresponde à complexidade da resposta do programa de televisão quando acionado algum comando pelo usuário ou evento externo. Uma forma de interação

simples corresponderia ao usuário selecionar um canal de áudio alternativo ou acionar as legendas, ou seja, apenas a habilitação ou não de alguma característica do programa em execução. Esse tipo de interação não requer a comunicação do dispositivo com alguma outra entidade (por exemplo, a emissora de TV). Uma interação complexa envolve a interação com outra entidade, de modo que seja possível alterar significativamente o programa a partir dessa interação, ou com a execução de um programa no dispositivo, localmente, e que esse programa altere o conteúdo apresentado na televisão. O tipo de interação também deve permitir a produção e o compartilhamento de conteúdo entre os usuários, de forma independente do produtor e transmissor do conteúdo original.

A fonte de conteúdos para televisão digital compreende a entidade responsável por prover a interatividade. Pode ser a própria emissora, com a transmissão dos dados no fluxo de sistemas MPEG, ou terceiros, utilizando meios como a Internet acessíveis pelo canal de retorno.

Por último, existe a questão de conectividade. A televisão é um meio de via única, com a transmissão de dados por difusão e geralmente desprovido de canal de retorno. Esse modelo requer que toda a interatividade dependa apenas do dispositivo local e dos conteúdos recebidos da emissora. A existência de um canal de retorno, ainda que simples (por exemplo, conexão à Internet por linha telefônica), permite o envio de dados para a emissora ou para outros televisores, ampliando as possibilidades de interação entre os usuários. Uma conexão de banda larga, por sua vez, permite a transmissão de conteúdos ricos, tais como áudio e vídeo, e o compartilhamento efetivo de material produzido pelos usuários.

Usualmente, classifica-se a interatividade de televisão digital em função dos principais serviços oferecidos. Por exemplo, Jensen (2005) classifica-a em melhorada, personalizada e completamente interativa. A primeira consiste no acréscimo de outros conteúdos, tais como textos e gráficos, ao vídeo transmitido. Guias eletrônicos de programação são um exemplo de serviço provido nessa categoria. A personalização contempla o controle sobre o fluxo do vídeo, permitindo-se interromper, continuar, repetir, avançar, retroceder, gravar, etc. A interação completa é possível quando existe um canal de retorno, permitindo, por exemplo, a comunicação com serviços de comércio eletrônico. Cesar *et al.* (2007) acrescentam à essa classificação o enriquecimento de conteúdos pelos usuários, o que requer dispositivos de interação mais sofisticados do que aqueles necessários para os serviços definidos por Jensen (2005).

De fato, considerando as características expostas nesta seção sobre a interatividade em televisão digital, atualmente é possível definir diversos tipos de serviço graças às plataformas de execução (*middleware*) e aos demais padrões associados a cada sistema de TVD. Na próxima seção, apresentam-se soluções sobre interatividade considerando-se objetos de aprendizagem.

3.3. Objetos de aprendizagem para televisão digital

O uso de televisão, em seu formato tradicional, possui valor educacional, permitindo a demonstração de procedimentos complexos no contexto real, a apresentação de conteúdos cuja representação visual é trivial em relação à textual, dentre outras possibilidades (ERTELT *et al.*, 2006). No entanto, esse poder de síntese também torna o processo superficial, levando o aprendiz a apenas reproduzir os conhecimentos e habilidades demonstrados, sem alcançar níveis adequados quanto aos objetivos cognitivos, afetivos ou psicomotores.

A partir desse panorama, alguns objetos de aprendizagem para televisão digital já foram desenvolvidos. Por exemplo, Araújo *et al.* (2009) definem uma aplicação sobre o sistema solar que emprega a apresentação de conteúdo multimídia, simulações e enquetes. Apesar de voltada para a televisão digital, ela não faz uso de vídeos ou se integra com os conteúdos provenientes pela rede de difusão terrestre. Nesse caso, a televisão digital é vista como uma plataforma acessível para execução de aplicativos em Java.

Monteiro *et al.* (2008) definem um objeto de aprendizagem sobre o efeito fotoelétrico na forma de uma aplicação para a plataforma Ginga NCL, combinando mapas conceituais e animações. Diferente do primeiro exemplo, este admite a integração e o uso simultâneo do programa televisivo (vídeo) (MONTEIRO *et al.*, 2009). Cabe ressaltar que o mapa é utilizado como parte do objeto de aprendizagem e não como um elemento do método de desenvolvimento: ele é utilizado para explicar os principais conceitos relacionados ao tema abordado enquanto as animações são utilizadas para ilustrar e simular situações. Existe a possibilidade de interação com o objeto, seja pela escolha dos conceitos do mapa e a apresentação de um texto descritivo ou pela alteração de parâmetros da animação, possibilitando a visualização dos efeitos das propriedades e conceitos estudados. Diretrizes de usabilidade para televisão digital definidas pela BBC foram utilizadas (BBC, 2002), mas nenhum outro método de desenvolvimento foi utilizado. Cabe destacar que as interações exigem apenas o uso do controle remoto e os botões direcionais, utilizando operações de escolha e seleção simples.

Uma limitação observada no contexto de TVD e confirmada nos trabalhos analisados, é a configuração dos sintonizadores (*set-top box*). Eles são sistemas embarcados, com pouco poder de processamento e memória disponível para aplicações gerais. Além disso, as capacidades de interatividade (em função dos dispositivos de entrada e saída por ele suportados e acessíveis) e de armazenamento de informações não são padronizadas, o que dificulta a criação de objetos de aprendizagem multimídia e interativos que sejam executados em todos os modelos de sintonizadores existentes.

Tal problema também é observado em dispositivos móveis (celulares). Comercialmente, os portais de aplicações são o principal meio de acesso às aplicações e, acredita-se, também será uma forma de acesso aos objetos de aprendizagem. No entanto, da mesma forma com

que as aplicações para celulares descrevem os requisitos para sua correta execução, também os objetos de aprendizagem o necessitarão. Tais informações devem ser definidas nos metadados, abrangendo desde as interfaces de programação (várias funcionalidades e API do SBTVD são opcionais) até os dispositivos de interação necessários.

Quanto à comunicação de dados nos sistemas de televisão, ela ocorre, predominantemente, por difusão. Essa forma de comunicação é a única que, garantidamente, está disponível e oferece uma largura de banda apropriada para a entrega dos dados. Por outro lado, ela ocorre apenas em uma direção, concentrando no canal de retorno a comunicação de dados do receptor para as transmissoras, para outros receptores ou para dispositivos diversos na Internet. O canal de retorno, por sua vez, oferece capacidades variáveis de banda e disponibilidade. As características de ambos os canais de comunicação – difusão e de retorno – devem ser consideradas no desenho de aplicações e de objetos de aprendizagem para TVD.

As características e dificuldades impostas pelos canais de comunicação e dos mecanismos de troca de dados entre as aplicações no receptor tornam o desenvolvimento de aplicações componentizadas e o acesso a serviços complexo. Cada aplicação é executada como um processo distinto, sem acesso direto aos dados e códigos de outras aplicações. A troca de dados precisa ser realizada por mecanismos de chamada remota, observando-se as limitações expostas no parágrafo anterior.

Uma abordagem para tratar esse problema é reduzir a necessidade de acesso a aplicações externas e oferecer interfaces que facilitem o acesso aos canais de comunicação (PAZOS-ARIAS *et al.*, 2008). López-Nores *et al.* (2010) implementam essa estratégia no arcabouço para *t-learning* ATLAS. Eles definem um objeto de aprendizagem como um conjunto de aplicações Java (*xlets*) para o *middleware Multimedia Home Plataform* (MHP). No escopo de cada *xlet* que compõe o objeto de aprendizagem, existe a troca de mensagens entre os objetos da aplicação. A comunicação entre as aplicações é realizada por chamadas remotas utilizando uma versão simplificada do RMI disponibilizado pela plataforma Java. Finalmente, para a comunicação entre objetos de aprendizagem em execução em outros receptores, utiliza-se uma rede de comunicação par-a-par (P2P) JXTA que oferece uma rede de comunicação em grupo (*multicast*). As classes Java implementam atos e cenas, formando uma peça, tal como definido no IMS LD. Associa-se, a cada objeto de aprendizagem, um gerenciador de cenário, encarregado da comunicação de dados e do sequenciamento. Com isso, é possível executar e coordenar atividades de aprendizagem localmente, mas não colaborativamente (com outras pessoas em locais distintos). Utilizando-se a rede *Peer-to-peer* (P2P), são estabelecidos grupos de usuários que podem realizar, então, atividades em conjunto.

Analisando-se a reusabilidade de objetos de aprendizagem, em geral observa-se que eles são vistos como aplicações educacionais, sem a composição por outros objetos de aprendizagem, utilização de metadados ou projeto instrucional. De fato, a denominação

objeto de aprendizagem ocorre pela simples utilização no contexto de educação, como instrumento para a instrução. Porém, um objeto de aprendizagem, conforme visto no capítulo anterior, possui características particulares, em especial a meta de reúso, a qual não se observa naqueles para televisão digital.

Um motivo para esse problema é a falta de padrões para representar um objeto de aprendizagem em TVD. Aqueles existentes foram definidos visando à plataforma Web, que possui conexão constante e bidirecional à Internet, disponibilidade de serviços Web, acesso a um ambiente de execução com linguagens como ECMAScript, interfaces de usuário com componentes gráficos WIMP, dentre outros. Em uma televisão digital, tais padrões não se aplicam tão diretamente: a conexão bidirecional não é garantida, o ambiente de execução não oferece, necessariamente, suporte a ECMAScript ou toda a gama de componentes gráficos tão habituais no ambiente *desktop*. Logo, são necessárias alterações nos padrões existentes para a representação e execução de objetos de aprendizagem para televisão digital (LóPEZ-NORES *et al.*, 2006).

De forma análoga às soluções propostas para o reúso de objetos de aprendizagem, encontram-se inicialmente na literatura trabalhos sobre a definição de metadados. Uma proposta é a descrição das características multimídia dos recursos educacionais com padrões já estabelecidos, tal como o MPEG-7 (HULSEN *et al.*, 2004) e o mapeamento dos metadados de objetos de aprendizagem (LOM) para formatos apropriados para televisão digital, tal como TV-Anytime (FRANTZI *et al.*, 2004).

A abordagem mais pesquisada consiste na extensão dos padrões atuais para atender alguns dos requisitos da TVD, definindo-se novos elementos de metadados (LóPEZ-NORES *et al.*, 2006; **??**). Por exemplo, uma solução é a descrição dos recursos de vídeo e segmentos de acordo com o padrão MPEG-7.

Em relação ao projeto instrucional, não foram encontrados na literatura trabalhos que definem métodos para a modelagem de objetos de aprendizagem para a TVD, ao menos não explicitamente. Por exemplo, Bellotti *et al.* (2008b) utilizam gabaritos (*templates*) que são preenchidos em tempo de desenvolvimento, adaptando a atividade modelada para a atividade de aprendizagem desejada. A limitação advém da pouca flexibilidade para definir a atividade, limitada aos elementos estabelecidos nos gabaritos.

Quanto ao processo de desenvolvimento, é necessário considerar as características inerentes daqueles dos programas televisivos. A usabilidade é um aspecto essencial da televisão (ERONEN, 2001), proporcionando experiências agradáveis e eficazes ao usuário quanto ao uso do produto em contextos específicos de seu dia a dia. Técnicas de projeto centradas no usuário permitem a compreensão das tarefas que o usuário precisa realizar e suas expectativas quanto à utilização de um produto, contribuindo para a qualidade de uso (HASSENZAHL; TRACTINSKY, 2006; MARTINS *et al.*, 2010).

Uma limitação associada a todos os objetos de aprendizagem é a inexistência de um ambiente virtual de aprendizagem para televisão digital. Em geral, as aplicações precisam prover tais funcionalidades ou definir adaptadores que estabelecem com os ambientes disponíveis em ambientes Web. Nesse caso, faz-se uso dos ambientes procedurais ou declarativos. Devido à natureza dos ambientes e suas interfaces de acesso (serviços Web), as soluções concentram-se nas linguagens procedurais, principalmente Java. A complexidade das aplicações varia desde simples sequenciadores, sem integração com ambientes (BELLOTTI *et al.*, 2008a, 2008b) até aqueles que proveem, além da integração, facilidades para comunicação em grupo (PAZOS-ARIAS *et al.*, 2008; LóPEZ-NORES *et al.*, 2010).

3.4. Considerações finais

Este capítulo descreveu os principais elementos sobre televisão digital e de questões de interatividade de objetos de aprendizagem para TVD. Da perspectiva tecnológica, observa-se que o SBTVD possui os mecanismos necessários para o desenvolvimento de objetos de aprendizagem interativos, limitado apenas pela falta de dispositivos adequados de interação. Existe uma dicotomia entre indústria e academia nesse quesito, dado que diversos trabalhos acadêmicos já discutem o uso de sistemas de reconhecimento de gestos e voz, integração de dispositivos pessoais e canais de retorno plenos enquanto que, na prática, a televisão digital privilegia a alta qualidade de imagem e som, mas limitada a controle remotos de teclas. Conciliar a interatividade necessária aos objetos de aprendizagem com essas condições reais é um desafio.

Quanto ao aspecto educacional dos objetos de aprendizagem, são poucos os relatos de desenvolvimento de objetos de aprendizagem com alinhamento pedagógico e utilização de mecanismos que contribuam com o reúso. Nos estudos encontrados, não foi detectada a utilização explícita de modelos para projetos instrucionais, focando-se mais na questão de usabilidade da interface gráfica, considerando-se a televisão digital um computador e não um meio de comunicação, associando-se os conteúdos normalmente transmitidos às atividades educacionais e o compartilhamento de novos conteúdos. As principais medidas adotadas para reúso são quanto a metadados, seja pela definição de metadados apropriados para as mídias de TVD ou pela adaptação dos metadados definidos em objetos de aprendizagem para aqueles utilizados em transmissões de TVD.

A partir das características de objetos de aprendizagem interativos sob a perspectiva educacional e computacional, apresentados no Capítulo 2, e da plataforma de televisão digital e interativa descritos neste capítulo, definem-se, no Capítulo 4, os requisitos para objetos de aprendizagem interativos e multimídias a serem considerados para a definição de uma abordagem para o desenvolvimento de objetos de aprendizagem multimídia e interativos para televisão digital.

Capítulo 4

Requisitos para desenvolvimento de objetos de aprendizagem multimídias e interativos

Recentemente, realizamos uma revisão sistemática sobre os fatores críticos (sucesso e fracasso) para educação em informática (*e-learning*) pela Internet (SILVA *et al.*, 2011e). Foram analisados estudos (recuperados das bibliotecas digitais da ACM, IEEE e Scopus) sobre cursos oferecidos pela Internet, a distância ou *blended*, de escolas e universidades, extraindo-se os fatores de sucesso e de fracasso relacionados à implantação deles. Ao todo, 91 estudos foram recuperados e, após a aplicação de critérios para a seleção de estudos, 31 trabalhos foram selecionados para análise.

Extraídos os fatores de sucesso e fracasso relatados nos artigos selecionados, procedeu-se a análise dos fatores críticos. Entre os fatores identificados, o item mais crítico para o sucesso foi o projeto de aprendizagem/instrucional. Treinamento dos professores, instrutores e tutores foi o segundo fator mais citado. A seguir, destacam-se usabilidade do ambiente virtual de aprendizagem e a interação entre os alunos, entre os alunos e o professor e do aluno com o conteúdo, todos esses relacionados com a usabilidade do ambiente, que é o meio onde as interações ocorrem.

A partir dos fatores identificados, observa-se que a tecnologia é um instrumento para educação: necessária, mas não suficiente. Projeto educacional, interação (atividades) e treinamento/qualificação técnica de docentes e funcionários despontam como fatores mais relevantes. A usabilidade do ambiente virtual de aprendizagem, o primeiro fator fortemente

relacionado com computação, tem uma estreita relação com interação: sem usabilidade, a interação é prejudicada. Obviamente, isso não significa que os aspectos computacionais não são importantes: eles apenas não figuram como críticos, atualmente, para *e-learning*. Neste capítulo, estabelecem-se requisitos quanto aos interesses relacionados a esses fatores: educação, computação e interação.

4.1. Requisitos educacionais para objetos de aprendizagem

Considerando os conceitos apresentados na Seção 2.1, pode-se definir um conjunto de requisitos que os materiais didáticos, realizados na forma de objetos de aprendizagem, devem satisfazer quanto às teorias educacionais e as demais características básicas de ensino.

O modelo apresentado na Figura 4.1 resume os elementos requeridos pelos objetos de aprendizagem (`Learning object`, em destaque na figura). Um objeto de aprendizagem é utilizado em atividades de aprendizagem (`Learning activity`). Uma atividade pode ser decomposta em subatividades que são sequenciadas de acordo com um projeto instrucional (`Instructional design`), visando à aplicação de métodos de instrução em conformidade com teorias de aprendizagem. Toda atividade é planejada para execução em um determinado contexto, definindo-se o tipo de acesso, a modalidade, a formalidade e a interação pelos quais uma ou mais pessoas participam visando alcançar um conjunto de objetivos educacionais (`Educational goal`) classificados de acordo com as taxonomias (`Educational objective`). A determinação do grau de satisfação de cada requisito é realizada pelo emprego de métodos de avaliação apropriados (`Evaluation method`) e da comparação dos resultados gerados (`Learning outcomes`) e esperados durante as atividades de aprendizagem.

Considerando essa visão de objetos de aprendizagem e seus aspectos educacionais, as próximas subseções tratam de requisitos gerais quanto à educação, o processo de ensino/instrucional e o projeto instrucional.

4.1.1. Requisitos quanto à educação

A educação é uma atividade que perdura por toda a vida de uma pessoa e que, consequentemente, causa mudanças nos seus conhecimentos e atitudes conforme o aprendizado ocorre. As diversas teorias educacionais e formas de aprendizagem são um reflexo disso, representando a evolução de nosso conhecimento sobre como aprendemos.

As teorias possuem alguns elementos em comum: o conhecimento e as atividades. No comportamentalismo, o conhecimento é passado ao aluno por intermédio de uma sequência

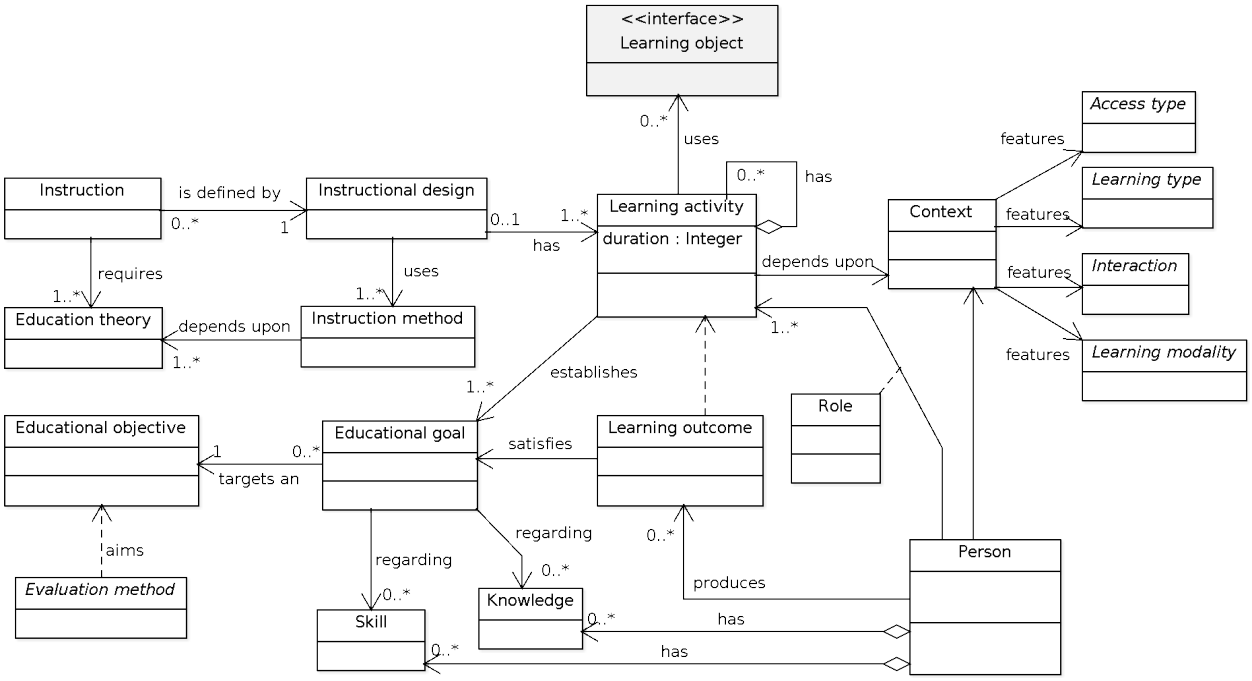


Figura 4.1: Elementos educacionais associados a um objeto de aprendizagem.

de atividades que talham e corrigem os conceitos que ele possui. Nesse caso, não é necessária uma estrutura complexa para representar o conhecimento a ser trabalhado. Enquanto isso, o cognitivismo necessita de um modelo que permita representar o conhecimento e a evolução dele. Por fim, o construtivismo requer um modelo que compreenda não apenas o indivíduo, mas o contexto em que ele se insere.

Quanto às atividades, o comportamentalismo preconiza uma forma direta de transferência de conhecimento: estimula-se o aluno, de quem se espera uma resposta (previamente estabelecida pelo professor), que, por sua vez, possibilita a criação de novos estímulos. Nesse modelo, o conhecimento previsto e o obtido são exatamente os mesmos. O cognitivismo não difere muito quanto a esse aspecto, mas, ao menos, estabelece ligações entre o conhecimento anterior e o novo. O construtivismo avança esse modelo ao não esperar que o modelo de conhecimento do aluno seja idêntico ao esperado (embora compatível), ou seja, ao invés de uma transferência absoluta de conhecimento, ocorre uma transformação que o torna único.

Continuando esta análise, com um viés educacional para tentar identificar semelhanças quanto aos principais elementos, a aprendizagem requer uma sequência de interações entre os seus diversos atores (alunos, professor, família, comunidade). No comportamentalismo, a interação restringe-se a um conjunto restrito de tipos de mensagens e símbolos, ignorando-se os processos internos de aprendizagem. A própria terminologia (estímulos e comportamentos) remete ao modelo computacional de máquinas de estados finitas: configura-se o estado inicial com conceitos que o aluno (supostamente) possui, fornecem-se estímulos (textos e questões) que levam a um novo estado (em que se assume que o aluno adquiriu um conhecimento bem definido), repetindo-se o processo com outros estados até que se alcance o estado final, em que

o aluno (provavelmente) aprendeu todo o conteúdo. Não é necessário ter uma representação de conhecimento além daquela presente no estado em que aluno ou o processo instrucional se encontra.

O cognitivismo não é passível de representação por uma única máquina de estado, dado que o estado inicial não é conhecido (conhecimentos prévios do aluno) e, principalmente, pela presença de vários níveis de memórias (os sensores, a memória de curto e de longo prazo). A representação do conhecimento não depende apenas daquele presente em um instante, mas do caminho percorrido e, conseguinte, da estrutura dos conceitos na mente do indivíduo.

O aprendizado segundo o construtivismo, no entanto, vai além: não é suficiente representar o conhecimento do indivíduo, mas também se deve caracterizar o contexto em que ele vive. A representação do contexto pode ser realizada apenas de forma limitada, o que restringe a capacidade de modelagem dessa situação. Além disso, não se trata apenas da transferência de conhecimento, mas também da criação de novos, por uma função (computação) desconhecida.

Quanto à execução do processo educacional e à elaboração de conteúdos didáticos, torna-se possível a elaboração antecipada dos mesmos por meio do comportamentalismo e, com algumas restrições, do cognitivismo. No comportamentalismo, é possível modelar os estímulos e respostas do sistema e projetar um sistema educacional adequado. Para o cognitivismo, é possível predizer um subconjunto de conhecimentos que os aprendizes possuem. No entanto, o conjunto de cada aluno é desconhecido, o que dificulta o estabelecimento de um modelo completo para todos os alunos. Uma alternativa é a personalização de conteúdos e de navegação de acordo com o perfil do usuário. O construtivismo, por outro lado, não suporta a preexistência de conteúdos: para cada aluno, um conjunto distinto de conhecimento é desenvolvido com base em experiências passadas e as interações com outros alunos e com o professor. Nesse caso, ao invés do material didático ser apenas um fim, ele também é um meio (FREIRE; GUIMARÃES, 2003, p. 109). A teoria construtivista requer um modelo de ciclo de vida iterativo, com participação dos atores na construção do conteúdo. Diferentemente do comportamentalismo e do cognitivismo, não é possível criar um modelo computacional centrado, inicialmente, em conteúdo, dado que este será um conjunto de artefatos resultantes do processo educacional, único para cada aprendiz.

Assim, quanto ao desenvolvimento de material didático (e de objetos de aprendizagem), definem-se os seguintes requisitos:

- Representação de conhecimento. Todos os modelos requerem, em maior ou menor grau, um esquema para representação de conceitos e relacionamentos. Em comum, todos estabelecem um modelo esperado de conhecimento e habilidades após o aprendizado (conforme representado pelos objetivos educacionais). Além disso, deve existir a representação do conhecimento do aluno (de forma simples, como no comportamentalismo, ou estruturada e temporal, como no cognitivismo) e flexível, de forma a possibilitar o

acréscimo de conceitos originalmente não previstos e, de certa forma, únicos para cada pessoa (tal como no construtivismo).

- Representação de estímulos (atividades). A mudança do conhecimento realiza-se apenas com a interação dos interessados (alunos e professores) entre si e com o material, a qual é estabelecida com atividades de aprendizagem.
- Representação de contexto. Compreende a descrição da situação em que o ensino ocorre e de mecanismos para registrar as contínuas alterações do contexto de aprendizagem em função das atividades de aprendizagem e do ambiente externo.

4.1.2. Requisitos quanto ao processo instrucional

As características do processo instrucional, apresentadas na Seção 2.1.2, são pontos importantes no desenvolvimento de objetos de aprendizagem. De fato, cada uma possui uma complexidade muito além da exposta nos curtos parágrafos que a descrevem. No entanto, a consciência da existência desses pontos permite definir o estabelecimento de limites quanto à definição de técnicas para criação de objetos de aprendizagem.

Quanto à formalidade, o ensino formal é respaldado por uma ampla base legal, a qual deve ser respeitada (obedecida). Em países como o Brasil, a legislação sobre o tema é demasiadamente complexa (as leis, em si, são simples e claras, mas os decretos e portarias que as regulam são diversos e ambíguos). Por exemplo, apenas em 2006 criaram-se cursos federais de graduação a distância pela Universidade Aberta do Brasil (Brasil, 2006), apesar da legislação permiti-los desde 1996. Acrescenta-se ainda que tais cursos não são totalmente a distância, exigindo-se algumas aulas presenciais. Enquanto isso, no âmbito informal, observa-se a ausência de organismos que promovam a autoregulamentação ou ações significativas e contínuas no âmbito governamental ou privado.

Uma segunda questão quanto à formalidade é a continuidade e integração delas. A disponibilidade de oportunidades de ensino que não tenham integração com a realidade das pessoas (principalmente na educação informal) e que não ofereçam atividades progressivamente mais ricas e complexas aos aprendizes não é desejável.

Em relação à modalidade, além do exposto anteriormente quanto à questão legal, é requisito que, quanto maior a distância entre os interessados, maior seja a interação planejada. Interação por si é necessária em qualquer atividade educacional, seja presencial ou a distância. No entanto, quando face a face, é possível perceber a (falta de) atenção e dúvidas, iniciar discussões, mudar a direção da atividade com base na intuição e experiência. Em educação a distância, essa capacidade de percepção e flexibilidade de mudança não existe nesse mesmo grau. Assim, devem-se planejar os conteúdos e atividades de modo que sejam oferecidas oportunidades para os interessados se externalizarem, apresentado comportamentos observáveis.

Cabe ressaltar que a distância em educação a distância compreende a distância física, temporal e transacional. As duas primeiras podem ser medidas de forma direta, mas a terceira representa os padrões de comportamento (tal como a fluidez do diálogo) entre os interessados em uma determinada situação (BOYD; APPS, 1980, p. 21), o que depende não apenas das duas primeiras distâncias, mas dos meios e ferramentas utilizados. Ela não pode ser medida quantitativamente e de forma absoluta, mas qualitativamente, detectando-se os desentendimentos entre os interessados em virtude da distância percebida entre as interações (MOORE, 1990). Intuitivamente, ela corresponde à distância percebida entre os interessados, refletindo aspectos afetivos e motivacionais da atividade educacional. Por exemplo, um professor expondo um conteúdo em aula, como se ignorasse a presença dos alunos, está mais distante transacionalmente de seus aprendizes do que um professor que coordena um fórum ao qual responde as perguntas prontamente.

Uma das propostas de educação a distância e, principalmente, do movimento de educação continuada é facilitar o acesso à educação. Não se trata apenas de oferecer educação para todos, com uma amplidão de cursos a distância, mas de uma postura aberta de educação, fomentada por instituições de ensino para a oferta de oportunidades de aprendizagem autônomas, gratuitas e sem restrições de acesso e materiais, e ferramentas didáticas com licenças de uso que concedam direitos de modificação e reúso (MEISZNER, 2011, p. 6). Para fazê-lo com qualidade, deve-se projetar unidades de ensino adequadas, personalizando-se o conteúdo para grupos de pessoas distintos, planejando-se cuidadosamente as interações e atividades e estabelecendo um mecanismo confiável de comunicação com os professores e entre os alunos. Nesse sentido, um modelo bem sucedido trata-se das universidades abertas (OU – *Open University*) (GIMENES *et al.*, 2012). Uma *Open University* (OU) prega a abertura quanto ao acesso (sem a imposição de requisitos para o ingresso do aprendiz) e forma de disponibilização (principalmente a distância), removendo-se restrições existentes no ensino tradicional e se adotando métodos e ideias de ensino adequadas para atrair os alunos e os educar com qualidade (PERRATON, 2006). A abertura observada aproxima-se daquela de recursos educacionais abertos (OER): material didático livre e gratuito para aprendizes e instrutores com fins de uso e reúso em atividades de ensino, aprendizagem e pesquisa (UNESCO, 2002; ATKINS *et al.*, 2007; BUTCHER, 2011; WILEY, 2010), em uma clara analogia com software livre e as comunidades de desenvolvimento. Logo, é importante explorar essa relação quanto ao processo de desenvolvimento de objetos de aprendizagem tal como recursos educacionais abertos para emprego em iniciativas tal como as universidades abertas.

Em relação à interação, é necessário representar os indivíduos que participam de cada atividade e os papéis que eles desempenham. Para cada atividade, é preciso especificar os recursos a serem utilizados e como eles serão utilizados.

Finalmente, existe a questão de avaliação. Ela é um instrumento essencial para

individualizar a educação (personalizando-a), guiar a execução das atividades, identificar dificuldades e atestar que os objetivos educacionais foram cumpridos. No ensino presencial, essa questão é mais simples: exercícios e provas permitem analisar os aspectos cognitivos enquanto que os afetivos e psicomotores podem ser avaliados pessoalmente pelos professores. No entanto, como fazê-lo na educação a distância? Embora seja viável utilizar exercícios e provas, a impossibilidade de observação direta do comportamento dos aprendizes não permite a correta aferição do progresso do aprendizem, principalmente nos domínios afetivo e psicomotor. Por outro lado, caso sejam utilizados sistemas computacionais para o acesso e execução das atividades, viabiliza-se a captura das interações dos aprendizes e, a partir desses dados, extrapolar algumas medidas que podem facilitar a execução das avaliações e a adaptação dos cursos de acordo com a reação e progresso dos aprendizes. A realização de várias atividades didáticas também reduz a distância transacional, permitindo uma avaliação qualitativa desses aspectos.

4.1.3. Requisitos quanto ao projeto instrucional

O projeto instrucional define processos comuns às atividades de instrução, conforme demonstrado no modelo ADDIE visto na Seção 2.1.3. Esse modelo assemelha-se aos processos típicos de Engenharia de Software, compreendendo a análise dos requisitos (objetivos de aprendizagem), projeto, desenvolvimento, implantação e avaliação. Além dos processos, o modelo também define um ciclo de vida iterativo, em que os resultados de cada iteração constituem uma entrada para o ciclo seguinte, configurando a melhoria contínua da instrução e das atividades e materiais didáticos associados.

No entanto, em Engenharia de Software, o modelo de ciclo de vida iterativo assume interações menores. No projeto instrucional, todos os objetivos de aprendizagem devem ser atendidos ao final da iteração enquanto que, em software, apenas um conjunto de requisitos – os mais relevantes para os interessados – é implementado. Dessa forma, os riscos inerentes a cada interação são menores e mais facilmente controlados.

Uma forma de contornar esse problema é estabelecer atividades de dimensões reduzidas (e poucos requisitos) para serem projetadas. A cada iteração, apenas os requisitos relevantes para as atividades de instrução de um período reduzido (*e.g.*, as aulas de uma semana) seriam considerados. Após a implantação e avaliação, os resultados obtidos seriam utilizados para projetar as próximas atividades, agora adaptadas de acordo com o aprendizado dos alunos. Ao final da macro-atividade (*e.g.*, uma disciplina), um produto completo se configuraria. Para um próximo período, ele poderia ser considerado como ponto de partida para uma atividade similar (*e.g.*, um novo oferecimento da mesma disciplina), mas ainda com as iterações para cada atividade que a compõe (certamente existirão requisitos diferentes, dado que o contexto e as características dos alunos mudam ao longo do tempo).

Uma divergência entre o modelo iterativo sugerido e o tradicionalmente utilização em educação é que, no primeiro, admite-se a alteração dos objetivos educacionais ao longo do desenvolvimento enquanto que, no segundo, os objetivos não são sujeitos a grau similar de alterações. No entanto, deve-se considerar que, embora do ponto de vista das instituições de ensino, os objetivos são fixos, pré-estabelecidos em ementas de cursos, para o aprendiz essa não é necessariamente a realidade, principalmente em cursos informais em que não se busca uma certificação, mas o desenvolvimento de conhecimento e competências que são um subconjunto dos requisitos estabelecidos do curso como um todo.

Assim, ao considerar o modelo iterativo, atende-se a todos os anseios. Caso opte-se pelo desenvolvimento de projetos instrucionais considerando todos os requisitos, realizando-se um processo em cascata ou incremental. Para casos em que se almeja flexibilidade, pode-se adotar um método mais ágil, com a realização de iterações que permitam o tratamento da variabilidade dos requisitos, promovendo uma educação mais personalizada.

4.2. Requisitos computacionais para objetos de aprendizagem

O desenvolvimento de objetos de aprendizagem pode ser definido de forma análoga à engenharia de software, envolvendo *processos*, *métodos* e *ferramentas* para o desenvolvimento de *produtos* de *qualidade* (BARBOSA, 2004). Nesta seção, definem-se, a partir dos elementos estudados na Seção 2.2, requisitos relacionados a esses elementos no contexto de objetos de aprendizagem em geral e que servem de instrumento para a definição de novas abordagens de desenvolvimento.

O produto "objeto de aprendizagem", conforme visto na Seção 2.2.2, pode ser apresentado em diferentes formas e complexidades. Embora esta tese seja sobre objetos de aprendizagem que contenham projeto instrucional (unidades de aprendizagem – UoL), todos os demais – recursos e objetos de informação (recursos com metadados educacionais) – são necessários para a efetiva construção desse. A Seção 4.2.1 estuda os requisitos para o desenvolvimento de objetos de aprendizagem sob essa faceta.

As ferramentas requeridas durante o ciclo de vida de um objeto de aprendizagem, desde sua concepção, passando pela construção e utilização, até a retirada de uso, são diversas: ferramentas de autoria, ambientes virtuais de aprendizagem, tutores inteligentes, jogos educacionais, etc. Atualmente, os esforços concentram-se em ferramentas de autoria e ambientes (na forma de gerenciadores de conteúdo de aprendizagem – LMS) que implementam padrões para a representação de metadados e projeto instrucional e para a comunicação entre o objeto de aprendizagem e o ambiente de aprendizagem, consoante com aqueles descritos na Seção 2.2.3. Nesse tocante, os requisitos quanto às ferramentas centram-se nos padrões

e abstrações para assegurar a interoperabilidade dos objetos de aprendizagem, tal como definidos na Seção 4.2.2.

Processos e métodos são elementos essenciais para a engenharia de produtos com qualidade sob a perspectiva do usuário final e da organização que realiza o desenvolvimento. Um processo estabelece um ciclo de vida, subprocessos, atividades e artefatos a serem utilizados e produzidos pelos desenvolvedores e demais interessados. A correta coordenação desses elementos, observando requisitos de qualidade, promovem segurança quanto à qualidade do objeto de aprendizagem e a satisfação do usuário. Os métodos permitem a realização de subprocessos e atividades e a obtenção de artefatos definidos pelo processo, aplicando técnicas específicas aos interesses e problemas envolvidos. A qualidade final do objeto de aprendizagem, resumida na satisfação dos requisitos e expectativas do usuário, pode ser sistematicamente estudada por modelos de qualidade, tanto de processo (meio) quanto de produto (fim). As abordagens examinadas na Seção 2.3 versam sobre esses tópicos e a análise delas permite o estabelecimento dos requisitos contidos na Seção 4.2.3.

4.2.1. Requisitos de taxonomia

Existe uma relação significativa entre o reúso e a granularidade do objeto de aprendizagem, indicando que objetos simples (recursos educacionais e objetos de informação) possuem mais reusabilidade (potencial de reúso). Diversos estudos tratam do reúso desses tipos de objetos de aprendizagem pela definição de metadados que possibilitem a recuperação desses objetos de forma eficiente.

No entanto, os objetos de aprendizagem que são efetivamente utilizados em atividades de aprendizagem são uma composição de muitos objetos simples e, do ponto de vista de professores e aprendizes, são esses objetos que deveriam ser reutilizados: não diretamente, mas pela adaptação ao contexto. Algumas técnicas, analisadas na Seção 2.3, estabelecem objetos de aprendizagem que não se restringem à simples composição, empregando estruturas que permitem a realização de operações de modularização, agregação, exclusão, substituição, reorganização e adaptação (MEYER *et al.*, 2011). Ao mesmo tempo, elas permitem a utilização de objetos simples.

Assim, uma abordagem para desenvolvimento de objetos de aprendizagem deve tratar de objetos com características bem distintas, porém complementares: de recursos simples a objetos de aprendizagem com projeto instrucional e outros modelos associados (Figura 4.2). Aos primeiros, para os quais não é possível definir um modelo, a utilização de metadados para descrevê-los é necessária para permitir a recuperação a partir de repositórios ou, no caso de recursos, mecanismos de busca tradicionais podem ser utilizados para recuperar itens relevantes. Quanto ao outro extremo, as abordagens devem considerar o uso de todo tipo

de objetos, desde recursos educacionais até outras unidades de aprendizagem, devidamente relacionados por um modelo instrucional e descritos por metadados.

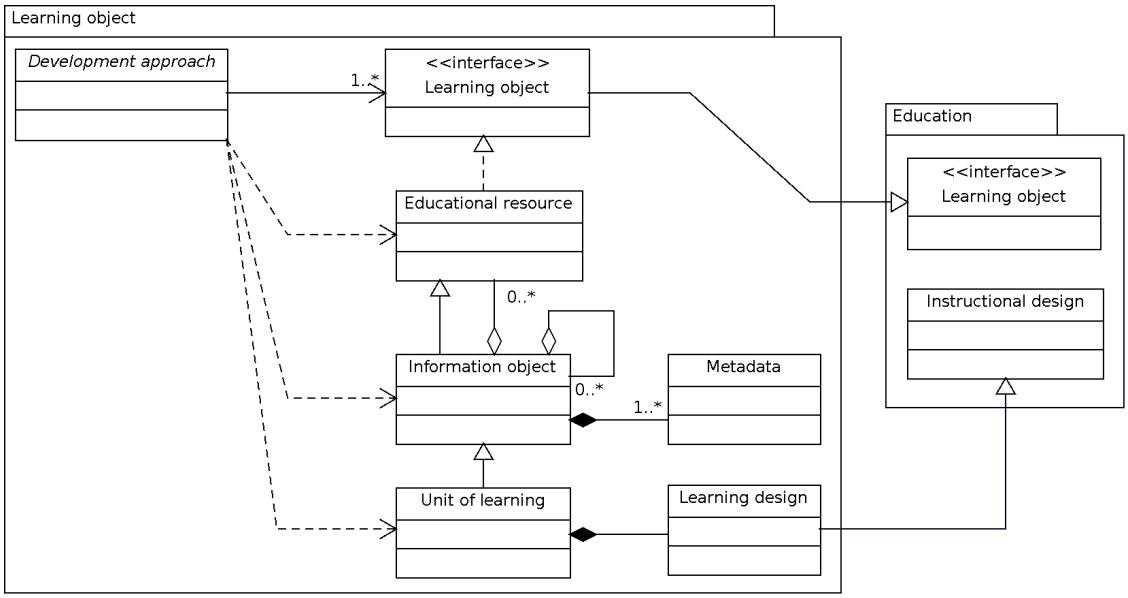


Figura 4.2: Tipos de objetos de aprendizagem que uma abordagem de desenvolvimento deve contemplar.

Na Figura 4.2 também é possível verificar a relação entre os elementos do modelo educacional (descritos na Seção 4.1) e estes do modelo de objetos de aprendizagem. Um objeto de aprendizagem, em educação, trata de qualquer recurso (concreto, *e.g.* um livro didático) ou evento (abstrato, *e.g.* o final da Segunda Guerra Mundial) que possa ser utilizado para fins educacionais. Neste capítulo, um objeto de aprendizagem é visto como um elemento representado computacionalmente e que possui um conjunto de características que buscam facilitar o reúso. Dentre esses elementos, têm-se os metadados, que não possuem um representante direto em educação (embora estejam relacionados com as diversas características da instrução e contexto educacional) e o projeto de aprendizagem, que é um modelo que, pelo emprego de linguagens de modelagem educacional, representa o projeto instrucional de uma atividade de aprendizagem.

4.2.2. Requisitos de representação

A representação de um objeto de aprendizagem engloba a definição de padrões adequados para a especificação dos metadados e do projeto instrucional e para a de execução e comunicação com outros objetos e aplicações educacionais, conforme apresentado na Figura 4.3. A princípio, uma aplicação educacional recebe os objetos de aprendizagem em um pacote contendo o conteúdo do objeto e os respectivos metadados e projetos instrucionais. O principal padrão para empacotamento é o IMS CP. Em geral, ele é suficiente para a distribuição dos objetos,

mas, com a emergência da utilização de serviços Web e computação nas nuvens, iniciativas para definir formatos alternativos para a disponibilização de objetos de aprendizagem encontram-se em desenvolvimento, tal como proposto no projeto TinCan (ADL; AICC, 2011).

Os objetos de aprendizagem, em uma aplicação educacional, não são apenas elementos passivos, utilizados apenas para apresentar informações. Existe a comunicação com a aplicação, realizada pelos mecanismos providos pelo ambiente de execução e utilizando o protocolo estabelecido por um modelo de dados. Os ambientes de execução, tais como o *SCORM Run-Time Environment* (SCORM RTE) e o ADL *HTTP AICC Communication Protocol* (HACP), definem uma API acessível pela Web para a troca de dados (*e.g.*, dados do aprendiz, resultados de avaliações). Os dados trocados seguem modelos de dados bem definidos, tais como o IEEE 1484.11.1 (para o caso do SCORM RTE) e o *CMI Data model* (para o HACP).

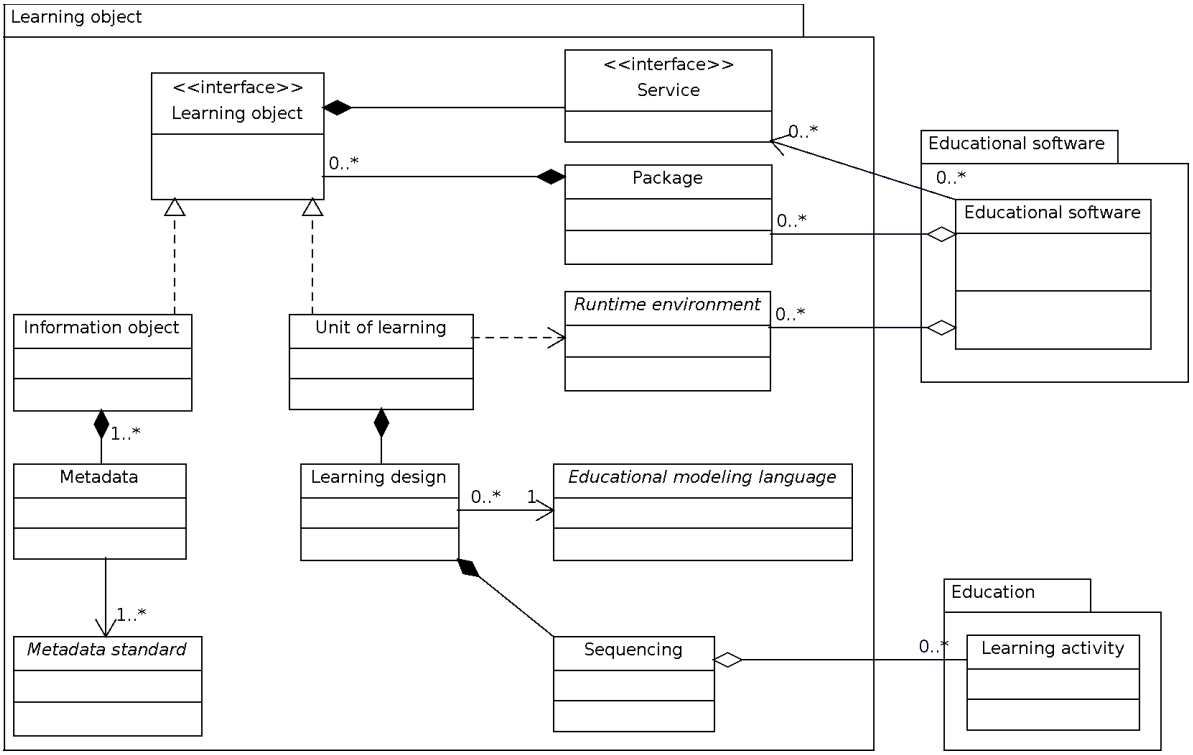


Figura 4.3: Elementos requeridos para a representação de um objeto de aprendizagem.

Os metadados devem representar as informações consideradas básicas para a descrição do conteúdo e do contexto de utilização, objetivando maximizar a sua recuperação com precisão em repositórios de objetos de aprendizagem. Além disso, como a caracterização dos atributos necessários para essa descrição varia de contexto para contexto, é preciso flexibilidade para definir outros atributos ou para o vocabulário utilizado como valor dos atributos. Por exemplo, o padrão IEEE LOM possui vários perfis de aplicação que estabelecem diretivas para a definição dos valores dos atributos, tal como o CanCore (MCGREAL *et al.*, 2004).

Além da especificação dos atributos, a forma com que eles são representados fisicamente também merece destaque. Não raramente, é necessário combinar metadados de diferentes interesses para descrever adequadamente um objeto de aprendizagem. Por exemplo, embora os padrões LOM ou DC sejam suficientes para descrever os propósitos educacionais, a descrição de detalhes de execução do objeto de aprendizagem e dos recursos por eles utilizados, tal como a resolução e demarcação de segmentos em um vídeo, não é contemplada. A solução é adotar formas de representação de metadados que permitam a combinação de diferentes padrões. A solução amplamente adotada é na forma de documentos e elementos XML: cada padrão adota um espaço de nomes distinto e, por consequência, é possível descrever as propriedades de cada objeto de aprendizagem utilizando o padrão de metadados mais apropriado (por exemplo, LOM para interesses educacionais e MPEG-7 para conteúdos multimídia).

O projeto instrucional, especificado em uma linguagem de modelagem educacional, determina a sequência com que as atividades de aprendizagem serão entregues ao aprendiz (sequenciamento). Os padrões IMS LD e IMS SS são os principais padrões. Os métodos que definem modelos para o projeto instrucional de um objeto de aprendizagem, ao invés de definir mecanismos próprios, optam por traduzir sua especificação em uma compatível com os padrões de mercado.

Como visto nos parágrafos anteriores, são diversas opções para a representação de cada elemento de um objeto de aprendizagem, opções essas nem sempre compatíveis. Diante da dificuldade encontrada na implementação de objetos de aprendizagem interoperáveis em função dessa diversidade de padrões, adota-se a utilização de perfis, ou seja, a combinação de subconjuntos dos padrões citados.

Os principais perfis são o SCORM (ADL, 2009), o *IMS Common Cartridge* (IMS CC) (IMS, 2011) e o AICC/CMI (AICC, 1998). Esses também são os perfis que os ambientes virtuais de aprendizagem implementam. No entanto, dada as diferenças entre os subconjuntos selecionados para cada perfil, prioriza-se a implementação de apenas um dos padrões ou dos elementos que lhes são comuns. Por exemplo, os metadados do SCORM e do IMS CC são os mesmos (LOM) e o padrão de empacotamento de ambos são compatíveis (IMS CP 1.1.4 e IMS CP 1.2, respectivamente). No entanto, quanto ao ambiente de execução, os três perfis suportam soluções diferentes: o SCORM utiliza o SCORM RTE, o IMS CC adota o *IMS Learning Tools Interoperability* (IMS LTI) e o AICC/CMI utiliza o HACP.

Embora existam esforços para definir um perfil e padrões extensíveis e, na medida do possível, compatíveis com os atuais para a representação e execução de objetos de aprendizagem – *e.g.*, o projeto Tin Can (ADL; AICC, 2011) –, o fato é que se deve focar em um dos perfis existentes. Caso contrário, não existirá uma aplicação educacional que possa utilizar o objeto de aprendizagem, prejudicando o uso e a avaliação das abordagens. Em geral, a combinação de vários tipos de metadados, mesmo que de forma redundante (por exemplo,

utilizar, em um mesmo objeto de aprendizagem, metadados no padrão IEEE LOM e DC), não é um obstáculo (preservada a consistência entre os valores declarados nos metadados). Quanto ao projeto instrucional, uma solução razoável é definir modelos que possam ser transformados em modelos equivalentes definidos em IMS LD. Futuramente, quando forem estabelecidos padrões mais adequados, basta alterar o mecanismo de transformação, preservando-se o modelo. Essa abordagem já é explorada em alguns trabalhos e também no contexto de personalização automática de objetos de aprendizagem, demonstrando o valor da técnica.

4.2.3. Requisitos para abordagens de desenvolvimento

As abordagens de desenvolvimento de objetos de aprendizagem compreendem os métodos, processos e modelos de qualidade. Os métodos tratam, em sua maioria, da modelagem da instrução, definindo as sequências de atividades e utilização de objetos de aprendizagem, estruturação de conhecimento, dentre outros requisitos referentes à modelagem e especificados na Seção 4.2.3.1. Os processos estabelecem os subprocessos e as atividades, adotando-se um ciclo de vida iterativo. A maioria dos processos é estabelecida para contextos específicos, tal como o processo RIVED ou MISA, mas é possível definir um modelo de processo, ou processo-padrão, que permite a representação dos processos, atividades e artefatos comuns a todos os processos, que, de fato, tornam-se uma especialização dele. Requisitos para o estabelecimento de modelos e de especialização dos processos são tratados na Seção 4.2.3.2. Finalmente, os modelos de qualidade permitem aferir a qualidade de um objeto de aprendizagem e do processo de desenvolvimento, nesse último caso exercendo papel na avaliação da maturidade de organizações e a definição de processos, a partir de um processo-padrão, que permitam a construção de objetos com qualidade.

4.2.3.1. Requisitos para métodos de modelagem de objetos de aprendizagem

Os métodos estudados na Seção 2.3 utilizam diversas técnicas para modelar um objeto de aprendizagem quanto à sua estrutura e comportamento, englobando desde os requisitos até o projeto instrucional. Tais técnicas proveem instrumentos para a identificação e definição de constructos (conceitos) e suas inter-relações, permitindo a delimitação do escopo e a organização desses com base em teorias de aprendizagem e métodos instrucionais previamente estabelecidos (BARBOSA, 2004, p. 63). Logo, a atividade de modelagem é essencial para a engenharia de objetos de aprendizagem, devendo satisfazer requisitos referentes à representação, estruturação e comportamento, conforme apresentado na Figura 4.4 e explicado nos parágrafos a seguir.

A representação consiste na identificação e definição dos principais constructos do modelo. Por exemplo, em um modelo de classe (do paradigma de orientação a objeto), o principal

conceito é a classe. Além desse, outros conceitos importantes seriam as propriedades e os métodos. No caso de uma técnica de especificação formal, tal como uma máquina finita de estados, os principais conceitos são o estado, seguido dos alfabetos de entrada e de saída, e as transições. No domínio de objetos de aprendizagem, pode-se afirmar que o constructo *conceito* (do domínio de conhecimento, tal como representado em um mapa conceitual) é relevante (dada sua presença em vários métodos descritos na literatura). Considerando o projeto instrucional, *atividade* é um conceito frequentemente empregado, definindo desde um curso (atividade de grande porte) até uma simples tarefa a ser realizada em uma aula.

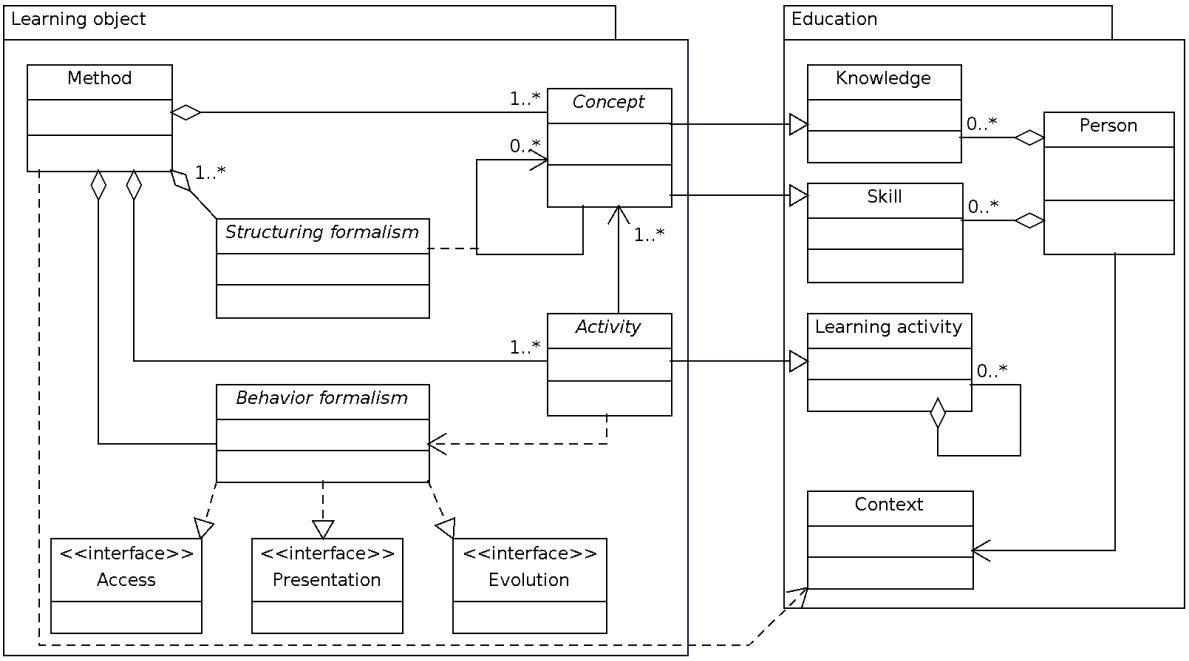


Figura 4.4: Elementos requeridos em um método de modelagem de objetos de aprendizagem.

Relacionada à questão de representação, existe a estruturação. Ela consiste da organização dos conceitos em uma estrutura útil, estabelecendo-se relações entre constructos distintos. Um mapa conceitual, por exemplo, é uma estrutura adequada para a representação de uma área de conhecimento. Da mesma forma, um *statechart* pode servir para a organização de estados.

O comportamento define, em função da representação e da estruturação, as ações que o modelo pode sofrer ou realizar durante a sua execução. Essas ações podem se restringir a mudar a configuração dos conceitos (tal como a ativação de um estado), a transferência de dados entre conceitos (mensagens) ou até mesmo a alteração da estrutura (por exemplo, pela definição de novos conceitos e a ligação desses com outros anteriormente estabelecidos).

Quanto ao comportamento, observa-se que diversos aspectos devem ser tratados em um objeto de aprendizagem: apresentação, acesso e evolução. O primeiro consiste na apresentação do conceito ou estrutura para o usuário ou o sistema. Para um sistema, a disponibilização

como serviços Web ou bibliotecas de sistema são exemplos de formas válidas. Para um usuário humano, no entanto, devem-se excitar os sentidos visual, gustativo, auditivo, olfativo e tátil. O acesso permite a navegação pelos conceitos, seja essa navegação direta ou então condicional (por exemplo, quando associada a um projeto instrucional e precondições de acesso a um determinado conceito). A evolução é uma consequência da apresentação e do acesso, as quais permitem o aprendizado por parte do aluno. Além de provocar a evolução dos conhecimentos e habilidades dos aprendizes, o próprio modelo sofre uma evolução, tendo sua representação, estruturação e comportamento alterados.

Em função desse modelo para métodos de modelagem para objetos de aprendizagem, os seguintes requisitos podem ser definidos. Barbosa (2004) estabeleceu um conjunto de requisitos abrangente relacionados ao conhecimento imbuído em objeto de aprendizagem:

- Diferenciação: Um conceito deve representar um conjunto interessante de instâncias do domínio de conhecimento e deve ser distinguível de outros conceitos de forma unívoca. Por exemplo, observando-se a teoria *Component Display* (MERRILL, 1983), definem-se os seguintes constructos: fato, conceito, procedimento e princípio; no Modelo de Michener (MICHENER, 1978), os constructos seriam *conceito*, *resultado* e *exemplo*.

Na perspectiva de estruturação, define-se ainda que os conceitos podem se relacionar pela herança (representando conceitos que compartilham semelhanças), pela composição de conceitos por outros conceitos (conceitos que são partes de outro conceito e cujas partes não possuem semelhança com o todo) e por grupos de conceitos e relacionamentos, e por relacionamentos específicos (relacionados a uma dependência funcional entre os conceitos):

- Taxonomia de conceitos: Representação da relação hierárquica entre os conceitos. Ela permite definir a estrutura dos conceitos em função de sua especialização (diferenciação progressiva) ou generalização (reconciliação integrativa) durante o processo de aprendizagem.
- Composição de conceitos: Representação da relação de composição entre os conceitos. Uma forma de diferenciação ou reconciliação é a percepção de que um conceito não é uma especialização ou generalização, mas sim parte de outro conceito.
- Referência ou uso de um conceito: Representação do uso de um ou mais conceitos. Tais relacionamentos não caracterizam uma relação hierárquica (dado que os tipos dos constructos são, em geral, distintos) ou composição (existe apenas a necessidade da referência a outro conceito, sem caracterizar uma dependência estrutural): o conceito relacionado é apenas utilizado ou referenciado para a definição de um comportamento específico do domínio, denotando uma dependência entre os conceitos.
- Decomposição hierárquica: Agrupamento de conceitos e relacionamentos em módulos que representam uma parte do domínio (mais precisamente um conceito).

Os requisitos acima citados tratam dos conceitos e da forma com que são estruturados. Em relação às atividades que utilizam os conceitos e o comportamento do modelo, deve-se observar:

- Ordenação ou sequenciamento: Estabelece a sequência em que as atividades permitirão a apresentação, o acesso e a evolução/alteração dos conceitos. Em geral, o sequencialmente depende das dependências dos conceitos, mas também pode ser definida pelo método instrucional. Por exemplo, em uma abordagem baseada em problemas, o conhecimento associado ao objetivo de aprendizagem está definido não no início, mas no final das atividades.
- História: Aquelas atividades que já foram executadas devem permitir a sua execução a partir do estado anterior ou a partir do estado original (por exemplo, em função do resultado de uma avaliação).
- Propagação de eventos: Os resultados de uma atividade não devem ser confinados a apenas àquela atividade e aos conceitos por ela utilizados, mas para todo o objeto de aprendizagem. Afinal, as alterações do conhecimento e habilidades do aprendiz, durante o uso do objeto de aprendizagem, possuem um efeito global.
- Contexto de aprendizado: As interações com as atividades não ocorrem apenas por parte do aprendiz, mas por influência do próprio ambiente em que a atividade é executada (tal como se ela escutasse e reagisse ao contexto).

Os métodos apresentados na Seção 2.3 atendem parcialmente a esses requisitos. Por exemplo, a abordagem IMA-CID satisfaz plenamente os requisitos quanto aos conceitos (estruturação), mas falha parcialmente quanto ao comportamento ao defini-lo plenamente em função dos conceitos. Métodos que utilizam mapas conceituais satisfazem a taxonomia, composição e referência/uso de conceitos, mas não permitem a definição de aspectos quanto ao comportamento e diferenciação de conceitos. Enquanto isso, os métodos baseados em Redes de Petri primam pela especificação do comportamento, mas não atendem aos requisitos expostos para conhecimento.

4.2.3.2. Requisitos para processos de desenvolvimento de objetos de aprendizagem

O processo de desenvolvimento de objeto de aprendizagem requer uma equipe multidisciplinar, com participação de atores da área educacional e computacional para a bem sucedida criação de objetos (SILVA *et al.*, 2010). Diversos trabalhos reconhecem isso e estabelecem mecanismos para facilitar a comunicação sem ambiguidades entre as partes – utilizando linguagens específicas de domínio e a transformação dos modelos com elas gerados – ou organizando os processos para permitir a seleção e adaptação daqueles relevantes para cada projeto de

desenvolvimento, estabelecendo-se arcabouços ou modelos de referência de processos.

Observando o processo de desenvolvimento de objetos de aprendizagem na execução de atividades de aprendizagem (Figura 4.5), são fatores primordiais a organização (para habilitar o desenvolvimento de processos de instrução bem resolvidos e acreditados pela sociedade), planejamento (para definir o processo e planejar a execução das atividades associadas sem comprometer o custo e a qualidade), a capacitação dos recursos humanos (para a correta execução das atividades dos processos), a disponibilidade de infraestrutura adequada e a execução das atividades propriamente ditas, destacando-se a construção dos objetos de aprendizagem e o suporte à utilização deles (SILVA *et al.*, 2010).

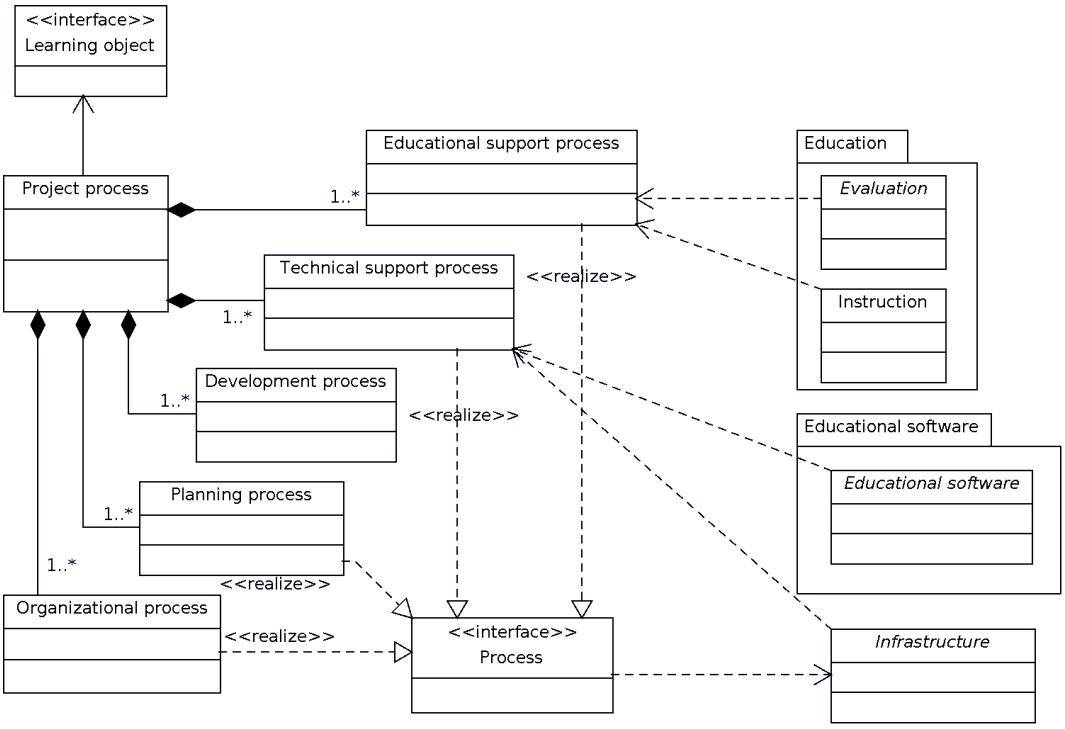


Figura 4.5: Elementos requeridos em um processo de desenvolvimento de objetos de aprendizagem.

Requisitos para tais elementos – processos organizacionais, de apoio e técnicos – podem ser definidos em modelos de referência ou processos-padrão. Modelos de referência são de suma importância para o desenvolvimento de produtos, definindo processos e atividades mandatórios e recomendados para o ciclo de vida de um produto, desde sua concepção até a obsolescência e retirada de uso. Aliados com os modelos de qualidade de processo, eles permitem o estabelecimento de processos concretos que oferecem um compromisso de qualidade em função da maturidade da organização e o tipo de produto em questão. O modelo de referência para processo de desenvolvimento de objetos de aprendizagem SP-DEM segue essa linha. Ele define uma extensão do modelo definido pela norma ISO 12207:1995 (ISO/IEC, 1995), com novos processos para tratar do desenvolvimento distribuído e aberto encontrado

na engenharia de software livre (e, consequentemente, de recursos educacionais abertos ou objetos educacionais livres). Processos para o desenvolvimento de objetos de aprendizagem devem considerar tais modelos em sua definição, completando-os com novas tarefas, atividades ou processos, conforme necessário.

Instâncias de modelos de processo de referência devem definir, além de processos concretos que satisfaçam aqueles definidos no modelo padrão, os atores que desempenharão os papéis de cada atividade e que serão responsáveis pela execução de cada processo e os recursos necessários a cada atividade.

Em relação ao modelo de ciclo de vida, vê-se a adoção de modelos iterativos, permitindo a avaliação contínua da qualidade dos artefatos produzidos e a evolução gradual do objeto de aprendizagem. Essa configuração é análoga àquela observada nas técnicas de processo instrucional e, de certa forma, era esperada. Objetos de aprendizagem devem ser adaptados conforme as especificidades de cada contexto e uma forma racional de alcançar esse objetivo é pelo desenho/projeto participativo junto com os potenciais usuários. Além de essa estratégia permitir a detecção e correção prematura de erros (principalmente quanto à usabilidade), ela promove um sentimento de apropriação pelos seus atores, o que é importante para fomentar o reúso de objetos de aprendizagem.

4.3. Requisitos para objetos de aprendizagem multimídias e interativos

O processo educacional requer a interação de seus atores com o meio, sendo este meio caracterizado por outros atores e pelos objetos de aprendizagem por eles manipulados. Tais objetos observam padrões de metadados, de formatos e de comunicação desses entre si, com os ambientes virtuais de aprendizagem e com os usuários.

Dados e metadados são necessários para a efetiva realização da interação entre os atores, mas não são suficientes. Não pode ser omitida a representação da interação dos atores com os objetos e dos atores com os demais atores. No cenário educacional, as interações necessariamente envolvem um ou mais atores, os quais podem ser reais (alunos, professores) ou virtuais (*e.g.*, ambientes virtuais de aprendizagem, aplicativos educacionais, agentes). Para alguns desses, foram definidos padrões para reger as interações. Do ponto de vista computacional, eles definem como os objetos interagem com os sistemas computacionais (tal como o SCORM). Do ponto de vista pedagógico, definem-se linguagens de modelagem educacional para a especificação das atividades (*e.g.*, recursos necessários e sequência de execução) de uma situação de ensino (*e.g.*, como um determinado tópico a ser trabalhado em uma aula).

Entretanto, ainda existe uma distância considerável entre as teorias educacionais/métodos

e processos instrucionais e as técnicas de desenvolvimento e implantação de objetos de aprendizagem. O desenvolvimento de objetos de aprendizagem nesse cenário é, na maioria das situações, restrito às competências do desenvolvedor e seu conhecimento tácito. Como consequência, a produção de objetos de qualidade e com custos e prazos bem definidos é prejudicada.

Com o objetivo de identificar e caracterizar as formas de interação encontradas em objetos de aprendizagem e as técnicas de desenvolvimento utilizadas, realizou-se uma revisão sistemática (KITCHENHAM, 2004), buscando a identificação, a avaliação crítica e a síntese de evidências de diversos estudos e o fortalecimento (ou a descrença) de teorias sobre o desenvolvimento de objetos de aprendizagem interativos (DYBA *et al.*, 2007). Por essa síntese, objetivou-se a captura das características de interatividade – tipo de interação, sentidos exercitados, atores, grau de computação, forma de condução da interação) –, bem como as formas de avaliação da efetividade das interações e dos objetos de aprendizagem no processo educacional. Com esses elementos, é possível identificar quais são as características mais importantes e que devem ser tratadas em métodos de desenvolvimento de objetos de aprendizagem interativos. A seguir são apresentados, brevemente, o protocolo da revisão sistemática e os resultados obtidos. Detalhes sobre o protocolo, seleção de fontes, seleção de estudos, extração e análise de dados podem ser consultados no relatório técnico (SILVA *et al.*, 2011c).

4.3.1. Protocolo da revisão sistemática

O objetivo da revisão sistemática é a identificação e a caracterização das formas de interação utilizadas em objetos de aprendizagem. A partir desse objetivo, delineiam-se as seguintes questões primárias e secundárias:

- **Q1:** Quais as características das interações que ocorrem ou que são possíveis com objetos de aprendizagem?
  - **Q1.1:** Qual o domínio de aplicação do objeto de aprendizagem?
  - **Q1.2:** Quais são os atores envolvidos na interação?
  - **Q1.3:** Quais são as formas de interação dos atores com os objetos de aprendizagem?
  - **Q1.4:** Quais são as formas de interação entre os atores?
  - **Q1.5:** Quais são as formas de interação entre os objetos de aprendizagem?
  - **Q1.6:** Quais são os dispositivos de interação que os objetos de aprendizagem utilizam?
  - **Q1.7:** Como são identificadas as formas de interação necessárias para um objeto de aprendizagem?
  - **Q1.8:** Como são projetadas as formas de interação a serem implementadas por um objeto de aprendizagem?

  - **Q1.9:** Como são implementadas as formas de interação em um objeto de aprendizagem?
- **Q2:** Quão efetivas são as formas de interação (adotadas em objetos de aprendizagem) quanto ao processo educacional?
  1. **Q2.1:** Como é avaliada a qualidade de um objeto de aprendizagem?
  2. **Q2.2:** Como é avaliada a qualidade da interatividade de um objeto de aprendizagem?

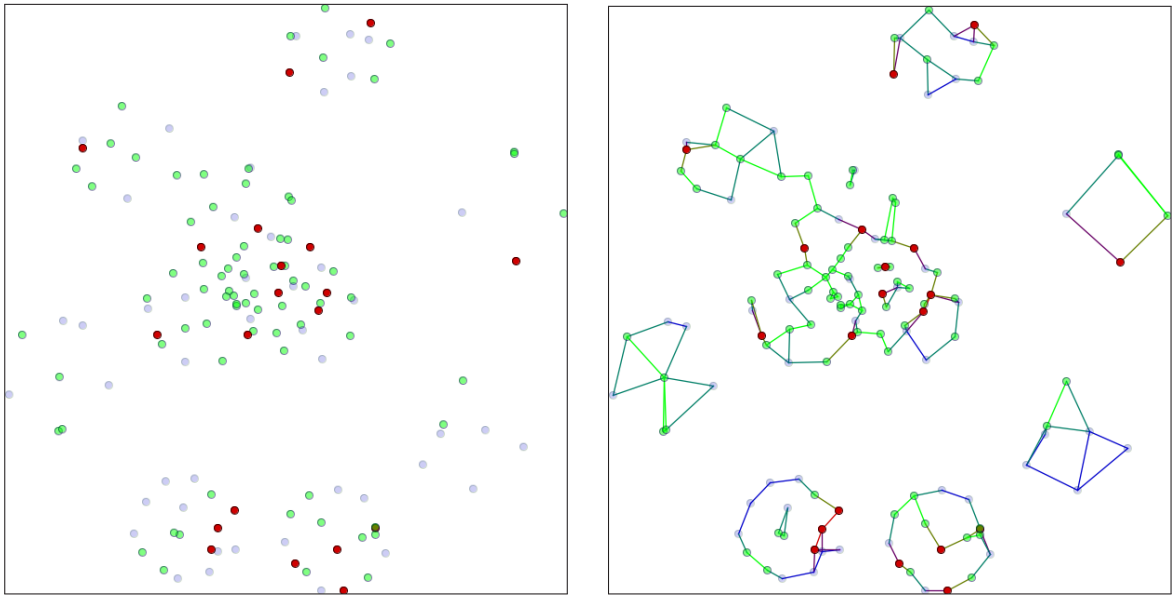
Considerando as questões acima estabelecidas, definiram-se os atributos e a amplitude da revisão sistemática com a técnica PICO (SANTOS *et al.*, 2007) e identificaram-se termos a serem utilizados na expressão de busca:

- **Quanto à população:**
  - objetos de aprendizagem,
  - modelagem de objetos de aprendizagem,
  - desenho de objetos de aprendizagem,
  - projeto instrucional.
- **Quanto à intervenção:**
  - interatividade,
  - colaboratividade.
- **Quanto aos resultados esperados:**
  - formas de interação,
  - técnicas de desenvolvimento de objetos de aprendizagem.

A partir dos termos identificados, define-se a expressão de busca para a recuperação de estudos. Todos os termos devem ser traduzidos de acordo com o idioma dos artigos que se deseja recuperar (inglês) e associados com sinônimos, conforme sugestões de especialistas. Com essa expressão básica, deve-se realizar a busca por estudos em fontes tais como bibliotecas digitais. Para cada estudo retornado, os seguintes critérios de seleção devem ser observados: idioma do artigo (inglês), tipo de estudo (apenas artigos completos), disponibilidade (integral e gratuitamente disponíveis para o pesquisador), conteúdo (apresentar um objeto de aprendizagem interativo, descrever as formas de interatividade empregadas e as técnicas para sua definição e implementação). Os critérios devem ser aplicados pela leitura do resumo e, posteriormente, do texto completo. Para complementar a seleção, devem ser utilizadas técnicas de visualização gráfica para aferir a qualidade da seleção em relação à proximidade dos estudos quanto aos termos utilizados e por gráficos de coautoria. Para cada pergunta secundária definida no protocolo, devem ser extraídos dados de cada estudo selecionado. Com esses dados, procede-se à identificação dos tipos de interação, os dispositivos utilizados, os atores envolvidos e as formas de avaliação empregadas pelos estudos.

4.3.2. Resultados

As fontes de estudos utilizadas durante a revisão sistemática foram as bibliotecas digitais da *Association for Computing Machinery* (ACM), *Institute of Electrical and Electronics Engineers* (IEEE) e ScienceDirect. Foram recuperados 215 estudos da ACM, 1.564 da IEEE e 17.483 da ScienceDirect. Para a seleção, todos os resumos dos estudos recuperados das bases da ACM e da IEEE foram lidos. Os critérios de seleção foram aplicados para os artigos recuperados, excluindo-se aqueles não atendiam ao menos um dos critérios. Dos artigos incluídos pelo resumo, aqueles que utilizavam, explicitamente, os termos *learning object*, *learning design*, *instructional design* e *educational module* foram lidos na íntegra e selecionados para a extração de dados. Ao final, foram selecionados 20 estudos da ACM, 18 da IEEE e 2 da ScienceDirect.



(a) Estudos selecionados. (b) Estudos selecionados (ligados por arestas).

Figura 4.6: Projeção dos estudos recuperados da ACM e respectiva classificação (vermelho para selecionados, verde para selecionados pelo resumo, azul para excluídos).

Em relação aos estudos da ACM, a Figura 4.6 identifica os grupos de estudos recuperados e classificados em excluídos, selecionados pelo resumo e selecionados pelo texto completo, tal como projetados com a técnica *Projection Clustering* (PAULOVICH; MINGHIM, 2006). Em poucas palavras, essa técnica permite a visualização de agrupamentos de documentos em uma coleção com base na similaridade dos termos contidos nos resumos.

Considerando a projeção dos artigos selecionados, identificaram-se estudos que, embora não definissem explicitamente um objeto de aprendizagem (ou seja, não foram escolhidos para leitura completa, ainda que selecionados com base no resumo – representados pelos vértices verdes), foram suficientemente similares a um estudo selecionado considerando o

texto completo (vértices vermelhos). Por exemplo, os estudos de Lavonen *et al.* (2001), Pickman e Liu (2009) (selecionados apenas pelo resumo, mas que utilizam uma nomenclatura pouco usual para objetos de aprendizagem), vizinhos do estudo de Ting e Jensen (1974) (selecionado), poderiam ser relevantes dada a similaridade dos termos utilizados nos resumos de ambos. Nesse caso, procedeu-se com a leitura do texto completo dos estudos de Lavonen *et al.* (2001), Pickman e Liu (2009) e, após isso, eles foram excluídos, dado que eles realmente não definiam um objeto de aprendizagem (e, portanto, não atenderam a um dos critérios de seleção). O mesmo procedimento foi realizado com os estudos recuperados da IEEE e da ScienceDirect, diferenciando-se apenas pela utilização de outras técnicas de projeção (mais apropriadas para o volume de dados a ser tratado).

Um total de 40 estudos foram selecionados para extração de dados a partir das bases da ACM, IEEE e ScienceDirect. Na Figura 4.7 são apresentados os trabalhos selecionados, identificando-se os grupos com base nos termos extraídos do título, resumo e evidências (obtidas na atividade de extração de dados).

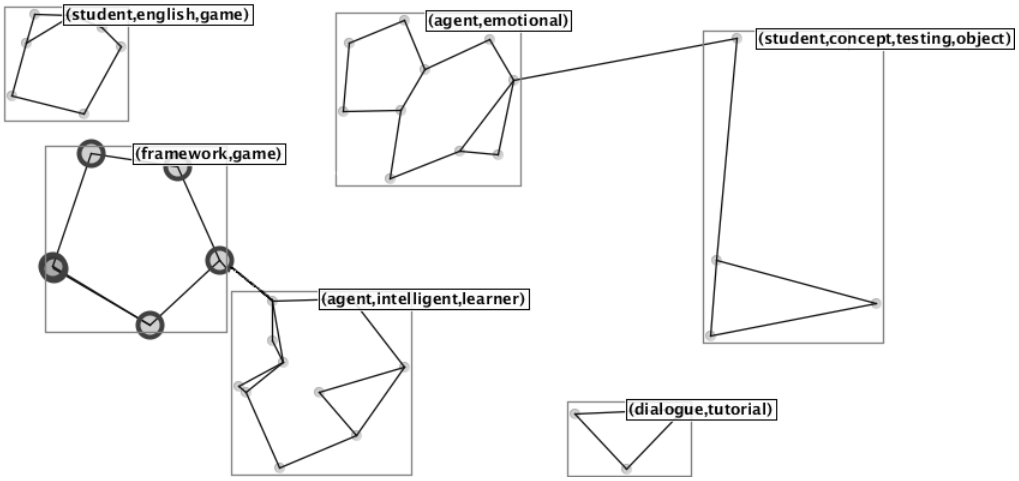


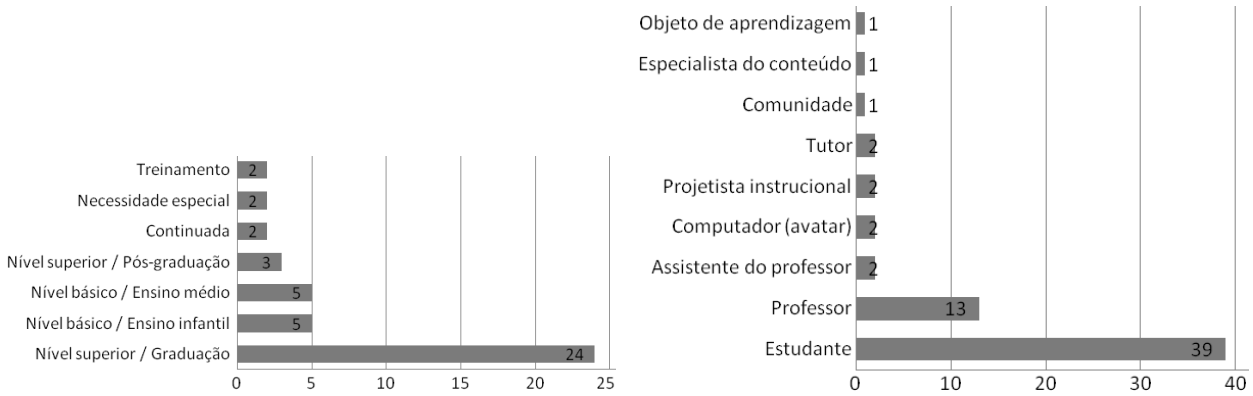
Figura 4.7: Agrupamento dos estudos selecionados de acordo com os termos extraídos do título, resumo e evidências.

4.3.3. Extração e análise de dados

Os quarenta estudos selecionados na etapa anterior foram analisados, extraindo-se dados relevantes quanto à interação, de acordo com as questões estabelecidas no protocolo da revisão sistemática. Os dados extraídos foram: contexto, atores, formas de interação, tipos de interação, dispositivos de interação, métodos para elicitação de requisitos, métodos para projeto e desenvolvimento, tipo de objeto gerado, forma utilizada para avaliar a qualidade do objeto e sua interação.

Os objetos relatados na literatura foram utilizados, em sua maioria, no contexto de ensino superior (graduação e pós-graduação): vinte e sete (27) de um total de 40 foram

utilizados nesse contexto. Uma quantidade significativa dos objetos (10) envolveu o nível básico (ensino de crianças e adolescentes). Poucos foram utilizados em educação continuada (2) e treinamento (2). Os resultados apresentados na Figura 4.8a podem ser explicados pelo contexto em que estão inseridos os autores dos estudos. Sendo professores universitários, a tendência é que desenvolvam objetos de aprendizagem voltados para suas áreas de atuação. Em relação aos atores que utilizam os objetos de aprendizagem, a maioria dos estudos envolve apenas o aluno (Figura 4.8b). Embora uma quantidade significativa também envolva o professor, essa relação ocorre majoritariamente na autoria dos objetos e não no uso. Em outras palavras, não existe interação entre aluno e professor.



(a) Contextos em que os objetos de aprendizagem foram utilizados.

(b) Atores envolvidos em um objeto de aprendizagem.

Figura 4.8: Contextos em que os objetos de aprendizagem são utilizados e atores que os utilizam.

Quanto à modalidade, observa-se, na Figura 4.9, a utilização de muitos objetos de aprendizagem no modo presencial e a disparidade quanto ao grau de aprendizagem. Embora a interação possa ocorrer em qualquer modalidade, espera-se que, no ensino a distância, enfatize-se a interação intermediada pelos objetos. No entanto, não foi o observado. Dos cinco objetos mais interativos, apenas um é a distância (os outro quatro são presenciais).

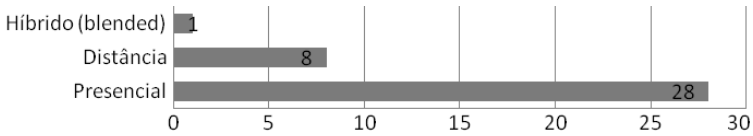


Figura 4.9: Modalidade de ensino em que os objetos de aprendizagem foram utilizados.

Mesmo entre os objetos de aprendizagem utilizados no modo presencial, existe uma disparidade quanto ao grau de interação provido: alguns com forte apelo interativo e cooperativo, enquanto outros com diversos mecanismos de interação, mas que não exercitam a interação entre os atores (principalmente alunos). Por exemplo, todos os jogos educacionais encontrados operam no modo de um único jogador: embora um jogo exercite vários sentidos

e tenha potencial de interação elevado, a ausência de jogos no modo cooperativo torna a interação mais pobre (principalmente no âmbito educacional).

Quanto à interação, as principais formas utilizadas nos objetos de aprendizagem são hipertextos, fóruns de discussão e enquetes (Figura 4.10). Uma explicação é que a implementação dessas formas de interação é simples de realizar. No entanto, elas exercem apenas um sentido (visual) com textos. Ambientes de realidade virtual também se destacam, presentes em 6 dos 40 estudos analisados. Apesar disso, na maioria dos estudos, os ambientes virtuais não são utilizados para interagir com outros atores, restringindo-se a apresentar de forma mais interessante os conteúdos a serem trabalhos educacionalmente.

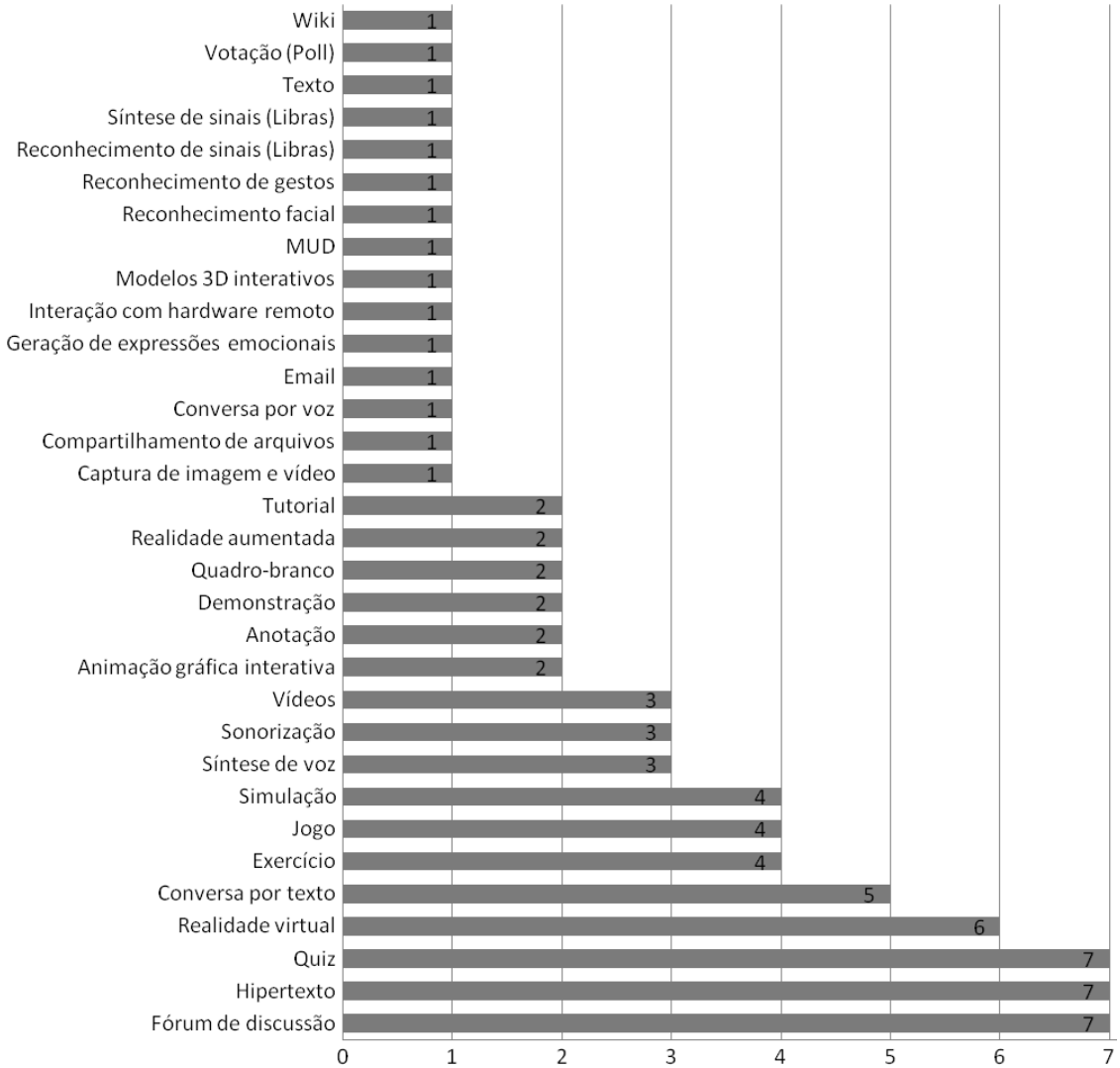
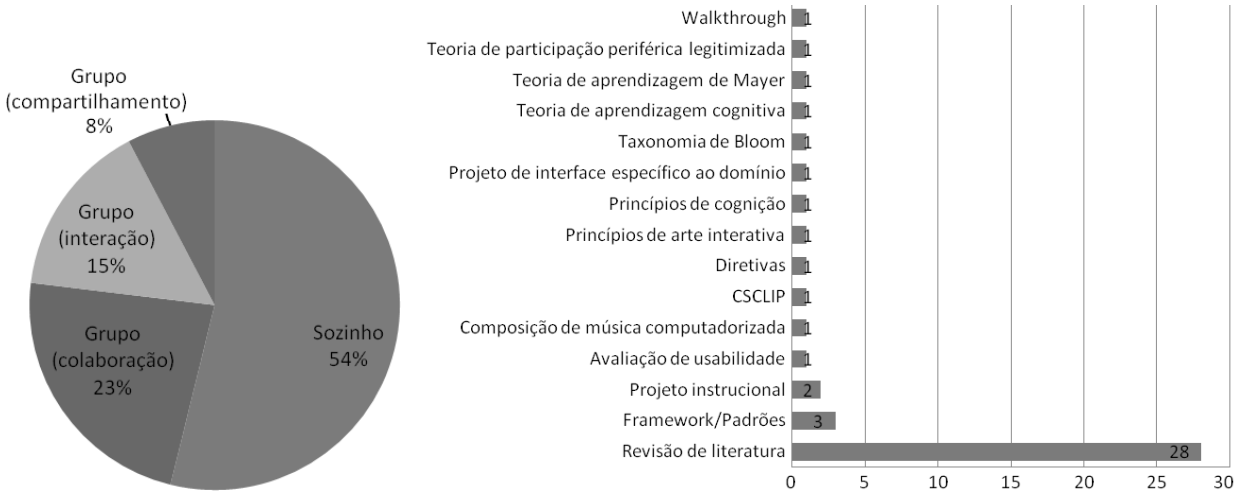


Figura 4.10: Formas de interação utilizadas em objetos de aprendizagem.

Em relação aos tipo de interação, observa-se na Figura 4.11a que a maioria dos objetos de aprendizagem considera sua utilização por apenas um ator, sem possibilitar a interação de outros atores no mesmo contexto.

A maioria dos estudos emprega apenas revisão da literatura (Figura 4.11b) para

especificar os requisitos educacionais e de interação que foram considerados para os objetos de aprendizagem. Embora os dois estudos mais interativos utilizem mais de uma técnica para elicitação de requisitos, não se observa uma relação entre o uso de técnicas específicas e o índice de interatividade.



(a) Tipos de interação utilizadas em objetos de aprendizagem.

(b) Técnicas para elicitação de requisitos de interação.

Figura 4.11: Tipos de interação e técnicas de elicitação de requisitos de interação

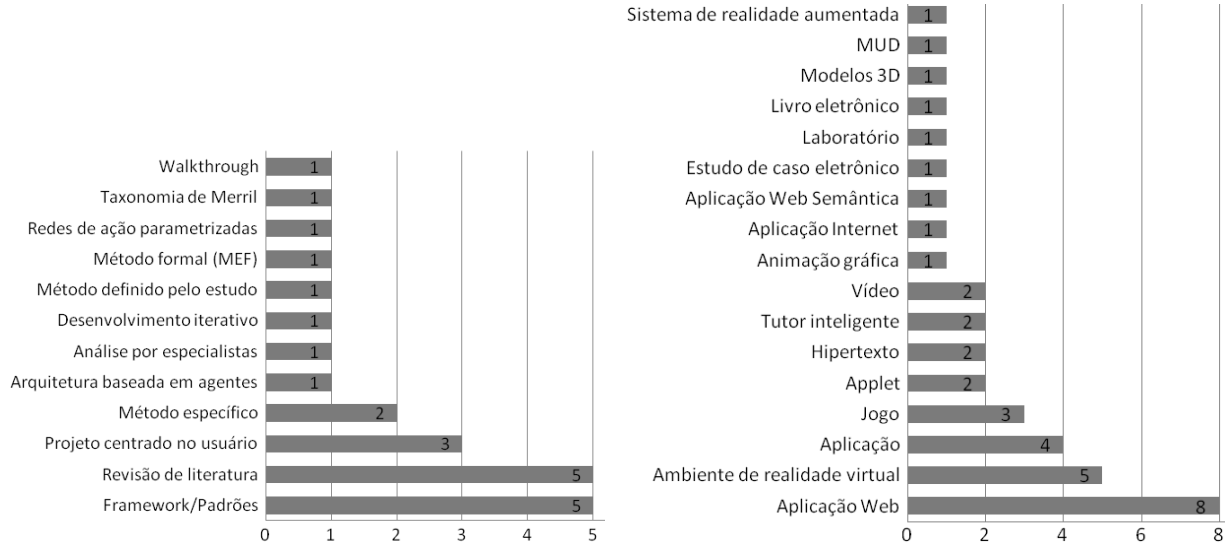
.

Quanto às técnicas de projeto, a maioria dos estudos não emprega técnicas para projeto do objeto de aprendizagem. Daqueles que as utilizam, observa-se a predominância do uso de padrões e projeto centrado em usuário (Figura 4.12a). Também não se observa relação entre o uso de técnicas de projeto e a interação dos objetos.

Observa-se na Figura 4.12b que diversas formas de implementação foram utilizadas para concretizar ou entregar o objeto de aprendizagem. A maioria dos estudos relata a disponibilização como aplicações Web, seguido de ambiente de realidade virtual e aplicações como um todo.

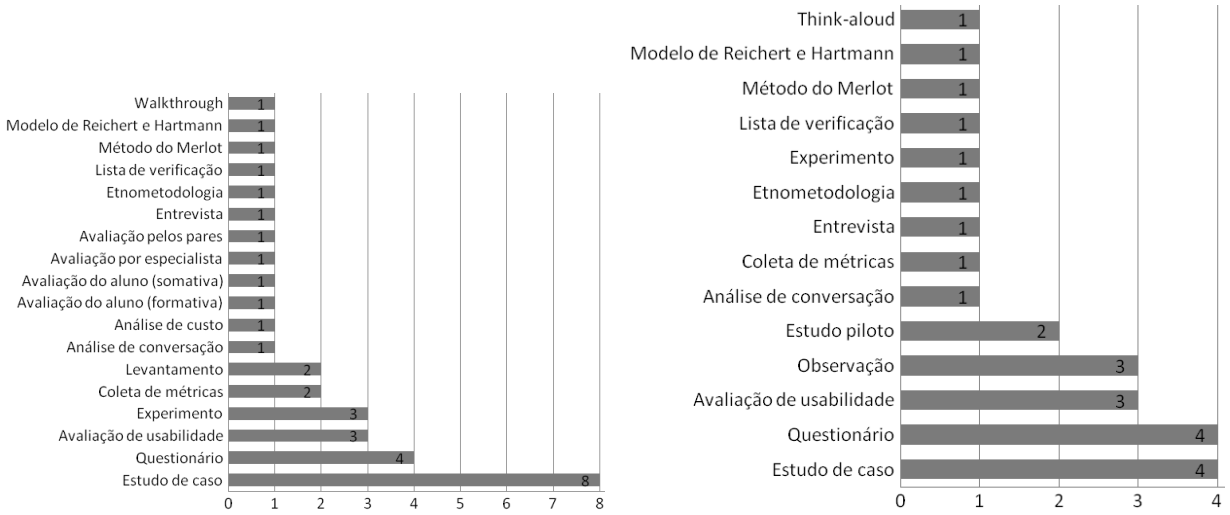
Finalmente, quanto à qualidade, a principal forma de avaliação de qualidade dos objetos de aprendizagem é estudo de casos (Figura 4.13b). Nesses estudos, questionários são frequentemente utilizados para coletar as opiniões dos atores (alunos) quanto ao objeto de aprendizagem.

A principal forma de avaliação de qualidade da interação constitui-se em estudos de casos e questionários. Avaliações de usabilidade e estudos observacionais também são utilizados. No entanto, observa-se, na Figura 4.13a, a baixa utilização de técnicas específicas para avaliação de interação homem-computador (especificamente, apenas duas técnicas – *Think aloud* e listas de verificação – foram utilizadas).



(a) Técnicas para projeto do objeto de aprendizagem. (b) Implementação dos objetos de aprendizagem.

Figura 4.12: Técnicas de projeto e formas de implementação de objetos de aprendizagem.



(a) Avaliação da qualidade dos objetos de aprendizagem. (b) Avaliação da qualidade da interação dos objetos de aprendizagem.

Figura 4.13: Formas de avaliação da qualidade dos objetos de aprendizagem e de sua interação.

4.3.4. Consolidação da análise

Considerando todos os fatores analisados, é possível sintetizar os dados conforme apresentados na Tabela 4.1. A ausência de métricas consolidadas para avaliar os objetos de aprendizagem torna a comparação nada trivial, principalmente quando os estudos não relatam adequadamente seus resultados.

Em linhas gerais, observa-se que os objetos mais interativos são aqueles utilizados no contexto de educação de nível superior (graduação e pós-graduação – valor de contexto igual a 4). Uma hipótese para explicar esse fenômeno é que adultos possuem maior facilidade em

Tabela 4.1: Relação entre a interatividade e os fatores analisados.

	Interatividade	Contexto	Modalidade	Atores	Elicitação	Projeto	Qualidade	QualidadeInteração
lucca-etal:2004	16	4	4	1	2	1	1	0
birchfield-etal:2006	15	3	1	1	3	0	0	0
fisher-etal:2003	14	4	1	2	1	0	0	0
ertelt-etal:2006	13	4	1	1	1	0	1	1
xu-wang:2006	13	4	1	2	1	1	1	1
dafoulas:2005	12	4	4	1	1	0	1	1
lao-dalida:2010	12	6	4	1	2	1	1	1
arbaugh-benbunanfich:2007	11	8	4	1	1	0	0	0
cuthbert-etal:2005	11	4	1	3	1	0	2	2
leblanc-etal:2005	11	4	0	2	1	2	0	0
lin-etal:2003	11	3	5	1	1	0	3	3
ting-jensen:1974	11	4	5	1	0	0	0	0
gard-mcauliffe:2009	10	8	1	3	1	0	2	0
adamovillani-wright:2007	9	1	1	2	2	2	3	5
cakir-etal:2010	9	2	1	1	1	1	3	3
li-etal:2003	9	4	1	3	0	1	0	0
ammar-neji:2006	8	0	1	2	1	1	0	0
boyer-etal:2008:b	8	4	1	2	0	0	1	1
bacim-etal:2010	7	4	1	1	3	1	1	0
dorneich-jones:1998	7	4	1	2	3	0	0	0
gonzalez-etal:2010	7	8	4	1	1	0	0	0
mcbride-mcmullen:1996	7	4	1	2	0	0	1	1
sawatpanit-etal:2004	7	4	1	2	2	1	2	1
adamchik-gunawardena:2005	6	4	1	2	1	0	1	0
anane-etal:2008	6	4	1	1	1	1	1	1
howard-etal:2010	6	16	0	2	1	1	0	1
mili-etal:2008	6	32	1	1	1	0	0	0
abad:2008	5	4	1	1	0	0	1	0
aleven-etal:2010	5	1	1	1	3	1	0	0
arvanitis-etal:2009	5	2	1	1	1	1	2	1
branovic-etal:2002	5	4	5	1	1	0	0	0
kerner-freedman:1990	5	32	2	3	0	0	0	0
kotzinos-etal:2005	5	4	1	2	1	0	0	0
murray-woolf:1991	5	4	0	2	1	3	1	0
rumetshofer-woss:2003	5	0	0	1	0	0	0	0
valentin-etal:2008	5	4	1	2	1	1	2	1
wang-etal:2004a	5	4	1	1	1	0	0	0
yang-etal:2008	5	1	0	1	1	1	2	1
hagen-etal:2006	4	4	0	2	1	1	1	0
assis-etal:2005	0	16	1	1	1	1	0	0

interagir com as ferramentas mais comuns (baseadas em texto, tal como hipertexto, fóruns e questões) e usá-las efetivamente. Entretanto, isso também implica que a interação é fruto da iniciativa dos atores e não algo induzido pelo objeto de aprendizagem.

Quanto ao uso de técnicas de elicitação e projeto, não se observa o uso consistente delas. Após a leitura dos objetos, chega-se à conclusão de que a experiência dos autores em atividades de ensino é a principal fonte de requisitos e que os projetos são realizados de forma *ad hoc* para solucionar uma situação de ensino que os autores (professores) enfrentam. Embora seja importante que os autores/professores sejam envolvidos na criação do objeto, a adoção de técnicas apropriadas de engenharia, tanto para elicitação, projeto, implementação e avaliação são necessárias. Caso contrário, a efetividade dos objetos de aprendizagem, de forma análoga a software, fica comprometida.

4.4. Requisitos para objetos de aprendizagem para TVD

Objetos de aprendizagem multimídia e interativos para televisão digital, possuem características particulares quanto à interação. A partir da análise dos trabalhos apresentados no Capítulo 3 e da revisão sistemática realizadas, o seguinte conjunto de requisitos é estabelecido.

Quanto aos dispositivos de televisão digital e o tipo de sinal de vídeo transmitido, é necessário considerar as *características do hardware*. O vídeo/imagem da televisão digital pode ser reproduzido no modo entrelaçado (linhas pares e ímpares intercaladas) ou progressivo. No modo entrelaçado, itens gráficos horizontais com tamanho inferior a 2 pixels (duas linhas) produzem um efeito de tremulação (*flicker*), o que causa cansaço visual (além de não ser agradável). Deve-se planejar os gráficos de modo que detalhes horizontais tenham, ao menos, de 3 a 4 pixels de espessura, evitando-se esse inconveniente (FLóREZ *et al.*, 2004). Outra limitação das televisões é a reprodução de cores. Não é recomendável utilizar cores saturadas (*e.g.*, acima de 80%), cujo brilho pode ser visualmente desagradável ao usuário (FLóREZ *et al.*, 2004).

Quanto à usabilidade, são vários os requisitos. O primeiro é um tanto óbvio: multimídia. A expectativa do usuário, ao acionar a televisão, é ser estimulado visualmente e auditivamente. Um erro comum é utilizar apenas textos (sem narração) e simples apresentações gráficas. Em TVD, é necessário que várias mídias sejam utilizadas, se possível estimulando mais de um sentido simultaneamente (multimodal) (MAYER; MORENO, 2003).

Em se tratando de multimídia e televisão digital, deve-se atentar ao fato da concorrência (diversas mídias apresentadas simultaneamente) e da sincronização (a alteração do estado de uma mídia em função de eventos relacionados a outras mídias). Por exemplo, o início da execução de um objeto de aprendizagem pode ser configurado pelo envio de um evento junto à transmissão de um programa de televisão e a apresentação dos elementos de um documento NCL devem ser coordenadas pelos eventos recebidos pelo ambiente externo (resultantes da interação com o controle remoto ou de sinais enviados pelo fluxo MPEG) ou de ações associadas às transições entre os contextos que contém os recursos apresentados.

O acionamento de um objeto de aprendizagem, transmitido com um programa de televisão, pode ser automático ou condicionado a um comando explícito do usuário, sendo esta última opção a mais desejada. Nesse segundo caso, é necessário que sejam fornecidas dicas visuais de que existe uma atividade passível de realização, representada por alguma dica visual (MARTINS *et al.*, 2010) e de forma não intrusiva, ou seja, não deve interferir ou interromper a atividade principal do usuário (SANTOS *et al.*, 2006; MARTINS *et al.*, 2010).

Questões estéticas não são uma exclusividade de televisão digital, mas é necessário considerar as condições típicas em que seu uso acontece. Os usuários geralmente assistem

em grupo e a uma distância significativa da tela (na ordem de metros). Dessa forma, deve ser considerada a utilização de fontes de estilos e tamanhos apropriados, parágrafos sem hifenização (AARRENIEMI-JOKIPELTO, 2004), textos devidamente espaçados, figuras proporcionais e cujos detalhes sejam nítidos a certa distância, bem como fazer uso de recursos esperados de televisão: vídeo e som (SANTOS *et al.*, 2006). Além de respeitar essas restrições, é preciso utilizar componentes gráficos multimodais para a interface, sem se ater à consistência e à (falta de) dinâmica das aplicações típicas de computador (baseadas no estilo WIMP) (CESAR; CHORIANOPOULOS, 2008). Efeitos de transição devem ser utilizados para as mudanças significativas de cena, tal como ocorre atualmente em televisão (*dissolve*, *fade out*, etc) (CESAR; CHORIANOPOULOS, 2008).

A interface das aplicações interativas deve utilizar um modelo de navegação e de interação consistente (CARVALHO *et al.*, 2007) e que seja fácil de aprender, ou seja, que não exija mecanismos de ajuda para explicar como funciona a interação. A interação deve ser natural e óbvia, exigindo modelos mentais simples (MARTINS *et al.*, 2010). Se for necessário explicar a forma de interação, ela deve ser fornecida na forma de vídeos ilustrativos, por requisição explícita do usuário ou pelo próprio objeto de aprendizagem ao detectar que o usuário está tentando interagir (*e.g.*, ao pressionar vários botões do controle remoto em sequência) sem ativar uma tarefa específica na atividade de aprendizagem.

Quanto às atividades interativas a serem contempladas no projeto instrucional, elas devem ser fortemente contextualizadas com o vídeo sendo transmitido. As atividades devem ser anunciadas para o usuário, utilizando dicas visuais e sonoras, e, quando acionadas, deve-se evitar o deslocamento total da atenção do usuário para a atividade, de modo que ele não perceba o que está acontecendo no vídeo. Isso implica em manter o vídeo sempre visível na tela, utilizando apenas parte dela para a exibição da atividade. Durante a execução da atividade, o progresso do usuário e o tempo gasto devem ser visíveis. Toda atividade também deve ter um limite de tempo para execução (e esse limite – e o tempo utilizado – devem ser mostrados para o usuário) (BELLOTTI *et al.*, 2011). As atividades devem contemplar não apenas a simples escolha de opções, mas também a criação e o compartilhamento de conteúdos pelo usuário (CESAR; CHORIANOPOULOS, 2008). Esses requisitos estabelecidos para a interação devem ser contemplados pelos métodos de desenvolvimento de objetos de aprendizagem para televisão digital. Além disso, para a representação dos objetos, deve-se adaptar os mecanismos apresentados no Capítulo 2.

Quanto aos metadados, os objetos de aprendizagem, em suas diferentes formas, precisam ser devidamente descritos para aumentar a possibilidade de reúso e permitir a adaptação quando necessária. No âmbito de televisão digital, devem-se descrever os recursos esperados dos dispositivos de interação e as características das mídias utilizadas (HULSEN *et al.*, 2004; BORGES *et al.*, 2008). Os metadados apresentados no Capítulo 2 são suficientes para tratar dos aspectos educacionais dos metadados de um objeto de aprendizagem. No

entanto, no contexto de televisão digital e objetos de aprendizagem interativos, torna-se necessária a especificação de requisitos quanto aos recursos requeridos para a execução das aplicações interativas: características dos receptores; programas de televisão para os quais foram desenhados; eventos que devem ser transmitidos durante esses programas; e o escalonamento dos recursos utilizados pelo objeto de aprendizagem (LóPEZ-NORES *et al.*, 2006).

Em relação ao ambiente de execução, deve-se estabelecer mecanismos de comunicação entre os objetos e o ambiente que sejam apropriados para as formas de comunicação disponíveis em televisão digital. A migração de um modelo de execução dependente de navegadores Web para um baseado em serviços Web, uma tendência já observada para os novos ambientes virtuais de aprendizagem, permite a implementação de objetos interativos quando o dispositivo de TVD está sempre conectado. Ainda resta estabelecer uma forma de comunicação satisfatória para a comunicação assíncrona, tal como sistemas de envio de mensagens ou serviços Web assíncronos.

4.5. Considerações finais

O ser humano é, por natureza, interativo. Os dispositivos computacionais hoje disponíveis – *tablets*, celulares, televisores – atendem a esse ensejo e convergem para tornar a computação ubíqua e natural. Um exemplo disso são os celulares, dispositivos nascidos para interação – telefonemas e envio de mensagens – e que incorporaram tecnologias de *tablets* – tela sensível ao toque, interfaces comandadas por gestos – e de televisores – reprodução de vídeos obtidos pela rede ou até mesmo o próprio sinal de televisão aberta por difusão. Para a nova geração, essa é a nova forma de computação esperada nas atividades de aprendizagem.

Sobre essas tendências educacionais e o suporte computacional, é importante lembrar que não são as instituições de ensino e pesquisa que determinam os dispositivos e tecnologias a serem utilizados, mas sim elas que se adaptam àqueles utilizados pelos alunos (LITTO, 2009). Embora possa se impor uma seleção de ferramentas para utilizar em um espaço controlado (a escola), o restante do tempo é vivido em um ambiente livre, no qual o uso de qualquer dispositivo é permitido. Se a educação continuada deve conciliar o ensino formal e o informal, é necessário adequar-se a essa realidade: não pela alteração do ambiente controlado, mas com a integração das duas realidades.

Todavia, não importa a presença de dispositivos com tantas capacidades se eles não forem bem utilizados. Necessita-se do desenho de objetos de aprendizagem que usufruam da alta fidelidade de áudio e imagem e que forneçam amplas oportunidades para interação, planejados para facilitar e motivar o aprendizado. Conforme constatado na revisão sistemática, existem várias aplicações e objetos de aprendizagem que são interativos e que alcançam bons

resultados, combinando princípios de educação (ciência) e ambientes imersivos, equipados com estímulos visuais e sonoros emitidos em resposta às ações dos usuários. Ao mesmo tempo, não se observam evidências de que o desenvolvimento foi conduzido de forma sistemática ou, ao menos, integrada a um projeto instrucional e teorias educacionais. Embora o projeto de sistemas interativos seja uma combinação de arte e ciência(DIX *et al.*, 1998, p. 4-5), deve-se enfatizar que ambas são necessárias.

Os requisitos estabelecidos neste capítulo prestam-se ao papel de definir aquilo que é necessário em objetos de aprendizagem multimídia, interativos e educacionalmente relevantes. A abordagem LOD, definida no capítulo seguinte, permite a criação de objetos de aprendizagem que atendam a esses requisitos, estabelecendo um método, um processo e ferramentas computacionais para o desenvolvimento dos objetos.

Capítulo 5

LOD: Abordagem para desenvolvimento de objetos de aprendizagem

A partir dos requisitos definidos no Capítulo 4, foi estabelecida uma abordagem para o desenvolvimento de objetos de aprendizagem multimídia e interativos para a televisão digital. Tal abordagem compreende o método *Learning Object Development Method* (LODM), construído a partir da abordagem IMA-CID e estendido para representar projetos instrucionais com interações dos usuários e entre as informações associadas ao objeto de aprendizagem. Transformações são definidas e parcialmente implementadas entre cada modelo, permitindo a geração sistemática de objetos de aprendizagem tanto para plataformas tradicionais como para sistema de televisão digital.

O método LODM foi criado visando à utilização em um processo mais amplo de desenvolvimento de objetos de aprendizagem, envolvendo equipes multidisciplinares trabalhando de forma aberta para a criação de recursos educacionais livres. O processo *Learning Object Development Process* (LODP) descreve esse processo a partir do modelo de processo *Standard Process for Learning Objects Development* (SPLOD). O SPLOD segue o mesmo conceito do SP-DEM, descrito na Seção 2.3.2.1, porém revisado para a versão mais recente da norma ISO 12207 (2008) e considerando também a norma ISO 15288 (2008).

Por fim, considerando o método e o processo definidos, o protótipo de um ambiente para desenvolvimento de objetos de aprendizagem, denominado *Learning Object Development Environment* (LODE), foi especificado e parcialmente implementado, contemplando as

principais transformações definidas pelo LODM e possibilitando a integração de ferramentas específicas para modelagem, tais como editores de mapas conceituais e de modelos UML.

5.1. LODM: Learning Object Development Method

LODM é um método para o desenvolvimento de objetos de aprendizagem baseado na abordagem IMA-CID. As alterações em relação ao método original consistem na completa adoção de uma abordagem de desenvolvimento orientada a modelos, contemplando-se a transformação de modelos e a geração automática de objetos de aprendizagem. Além disso, os modelos definidos pelo LODM servem ao propósito da modelagem de interações alinhadas com o projeto instrucional e à geração de objetos conforme as plataformas de saída desejadas (mídias tradicionais, tais como conjuntos de transparências e aplicativos para televisão digital), atendendo aos requisitos propostos para os métodos de desenvolvimento de objetos de aprendizagem estabelecidos no capítulo anterior.

Os modelos definidos no LODM são três: conceitual, instrucional e de interação. O modelo conceitual representa os objetivos educacionais na forma de um modelo de conhecimento estruturado em conceitos e proposições. O modelo instrucional representa a parte do projeto instrucional relacionado com os conceitos do modelo conceitual, definindo-se a organização das informações necessárias para as execuções das atividades de aprendizagem. O modelo de interação complementa o modelo instrucional, definindo elementos instrucionais e reorganizando os elementos do modelo instrucional, definindo as atividades de aprendizagem em função da sincronização entre os elementos do modelo.

Os modelos são definidos de forma iterativa, conforme apresentado na Figura 5.1. Inicia-se pelo modelo conceitual, construído a partir de outros materiais didáticos ou modelos conceituais. Os conceitos e proposições são organizados de acordo com métodos instrucionais e representados no modelo instrucional. Caso identifiquem-se erros no modelo conceitual, é possível voltar ao modelo anterior para alterá-lo e gerar um novo modelo instrucional. O mesmo raciocínio se aplica para o modelo de interação quanto ao modelo instrucional. Ao término da especificação das interações, escolhe-se a plataforma de execução (*e.g.*, televisão digital), e geram-se objetos de aprendizagem a partir do modelo de interação. Caso satisfatórios, eles são utilizados na implementação da instrução e, após sua execução, inicia-se uma nova iteração de desenvolvimento, revisando-se e alterando-se os modelos em função dos erros observados e das avaliações dos aprendizes.

As próximas seções especificam os modelos utilizados no LODM e as diretivas para sua construção e transformação entre modelos.

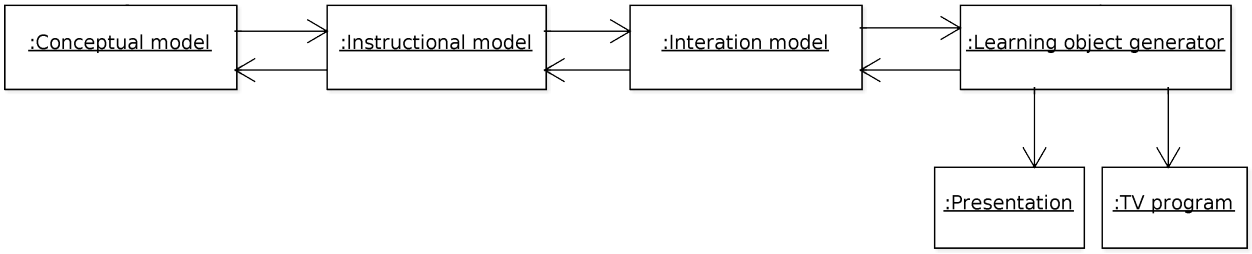


Figura 5.1: LODM: Visão geral do método.

5.1.1. Modelo conceitual

O modelo conceitual tem o propósito de representar os objetivos educacionais do objeto de aprendizagem. Ele é construído a partir da técnica de mapeamento de conceitos para a identificação dos conceitos relevantes para a compreensão do domínio de conhecimento e a especificação da forma pela qual eles se relacionam (NOVAK, 1977). Considerando um objeto de aprendizagem, este modelo estabelece os principais requisitos da perspectiva educacional.

Observando o princípio de independência de abordagens educacionais, mapas conceituais satisfazem as principais teorias. No comportamentalismo, os mapas representam o estado final que se deseja alcançar na forma de um modelo de conhecimento (COFFEY *et al.*, 2002), não existindo preocupação quanto aos processos cognitivos e às progressivas alterações do entendimento do aprendiz. Quanto ao cognitivismo e construtivismo, o mapeamento conceitual é especialmente apropriado em virtude do embasamento na Teoria de Aprendizagem Significativa (AUSUBEL, 1968), regida pelos princípios da diferenciação progressiva e reconciliação integrativa. Esses princípios estabelecem, respectivamente, que os conceitos são progressivamente diferenciados em conceitos mais específicos (estabelecendo uma hierarquia) e que as semelhanças e diferenças entre os conceitos são representadas por proposições que permitem a exploração explícita de semelhanças e diferenças significativas entre ideias relacionadas e a reconciliação de inconsistências reais ou aparentes dessas ideias (NOVAK, 1977; FARIA, 1989). Dessa forma, o mapa conceitual pode ser utilizado não apenas para a construção do objeto de aprendizagem, mas também para visualizar o significado educacional do objeto de aprendizagem e possibilitar o aprendizado significativo (NOVAK; GOWIN, 1984, p. 1).

Conforme dito anteriormente, um mapa conceitual é composto por conceitos e proposições. Um conceito representa um objeto ou evento cuja ocorrência não seja acidental, mas percebida ou observada de forma regular ou de acordo com algum padrão (NOVAK, 1998, p. 21). As proposições são as relações percebidas quanto a dois ou mais conceitos.

A representação de um mapa é feita graficamente, identificando-se os conceitos como vértices rotulados e as proposições como arestas rotuladas. Os rótulos dos conceitos são, em geral, substantivos e únicos em um mapa conceitual, correspondendo à noção de que um conceito, em um domínio específico, é único. Os rótulos das arestas são representados

por verbos e preposições, formando uma oração. Toda proposição possui ao menos um verbo ligando dois conceitos e, opcionalmente, uma preposição ligando outros conceitos complementares. Ambos, verbos e preposições, podem ser repetidos, sendo que o verbo classifica o relacionamento entre os conceitos associados (CANAS; NOVAK, 2008).

O mapa conceitual utilizado no modelo conceitual da LODM, tal como na abordagem IMA-CID, é estendido com primitivas para a representação de relacionamentos dos tipos de taxonomia e de composição, que representam duas operações típicas da construção de um mapa conceitual. Os relacionamentos de taxonomia são representados pelo rótulo `is a` e os de composição por `is part of` ou `has` (dependendo da direção do relacionamento). Os demais tipos de proposição denotam, conforme o verbo apresentado, uma forma de uso ou dependência funcional entre os conceitos. Esse tipo de relacionamento é classificado como funcional.

O diagrama contido na Figura 5.2 representa os elementos do modelo conceitual, considerando os tipos de proposições apresentados até então. Formalmente, ele pode ser definido como uma tupla $M_C =< V, T, \delta, C, P >$ em que:

- V é o conjunto de símbolos (palavras) que podem ser utilizadas como rótulos nos elementos do mapa conceitual;
- T é conjunto de tipos de proposições permitidas no mapa conceitual, que no LODM é definido pelo conjunto $T = classification, composition, use$;
- δ é uma função parcial sobrejetora $\delta = V \twoheadrightarrow T$, ou seja, para cada tipo de proposição, existe ao menos um nome no vocabulário do mapa conceitual.
- C é o conjunto de conceitos do mapa conceitual, tal que cada conceito é composto por um rótulo l que é uma palavra tal que $l \in V \wedge \delta(l) = \varnothing$, ou seja, uma palavra que não seja utilizada para representar algum tipo de proposição;
- P é o conjunto de proposições do mapa conceitual, sendo que uma proposição p é uma sequência de tuplas $< t, l, c_S, c_T >$ em que:
  - t é o tipo da proposição, tal que $t \in T$;
  - l é o rótulo da proposição, que é um verbo ou uma preposição que respeita as seguintes regras:
    - Se $t = classification \vee t = composition$, então $l = \delta(t)$,
    - Senão, $l \in V \wedge l \notin C \wedge \delta(l) = \varnothing$.
  - $c_S \in C \wedge c_T in C$.

Graficamente, o modelo conceitual é representado por um grafo direcionado cujos vértices rotulados correspondem a conceitos e cujas arestas rotuladas definem as proposições entre dois ou mais conceitos. Os conceitos são preferencialmente organizados de forma hierárquica, tal como exemplificado na Figura 5.3, que possui o vértice `Software testing` como raiz do modelo e que denota o principal conceito sobre o tema tratado pelo modelo conceitual

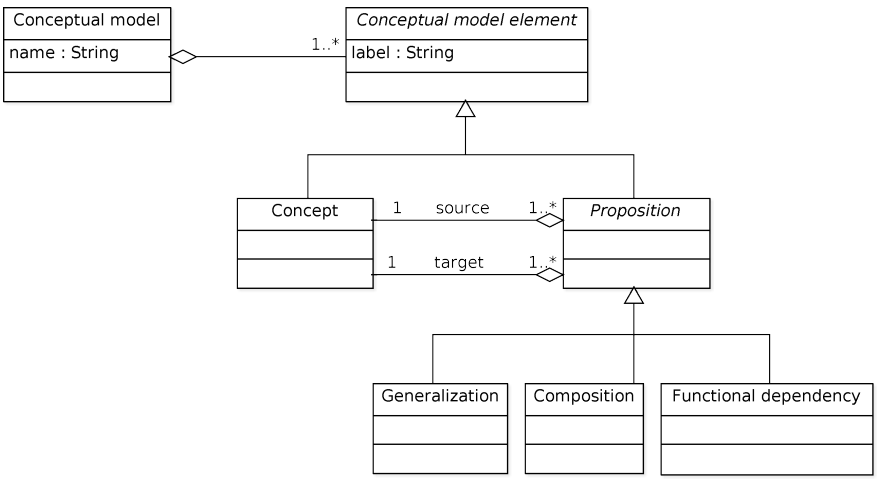


Figura 5.2: LODM: Diagrama de classe dos elementos do modelo conceitual

(destinado a um curso introdutório sobre teste de software). A leitura do modelo é realizada tal como a leitura textual ocidental: de cima para baixo, da esquerda para a direita. A direção das arestas, por padrão, segue essa lógica, não apresentando explicitamente a direção quando ela aponta para um conceito hierarquicamente abaixo ou para um conceito no mesmo nível, porém à direita. Nos demais casos, utiliza-se uma seta para indicar a direção da aresta e a forma de leitura da proposição. Opcionalmente, pode-se indicar a direção de todas as arestas, mas isso não é indicado: a intenção é que, de fato, a leitura do modelo conceitual seja o tão natural quanto possível.

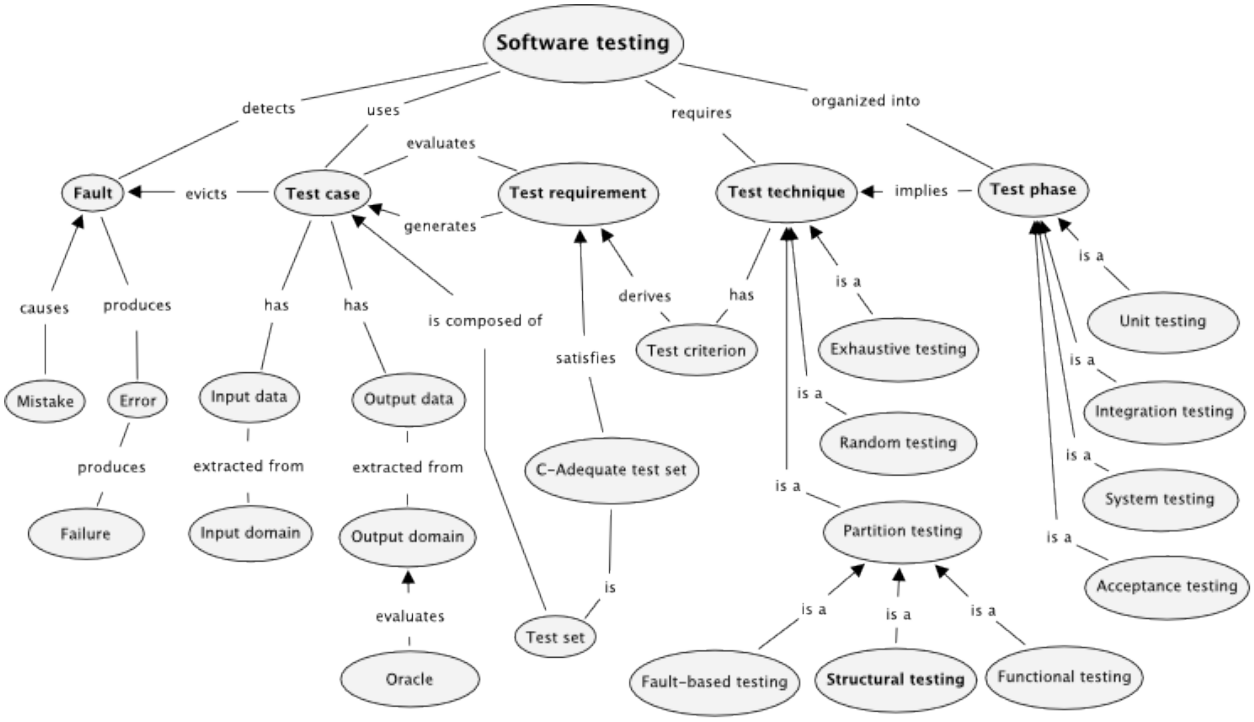


Figura 5.3: Modelo conceitual sobre teste de software para um curso introdutório sobre o tema.

O vocabulário V é o conjunto de todas as palavras definidas no modelo conceitual: `Software testing`, `requires`, `Test technique`, etc. Os três tipos de proposições permitidos pelo LODM foram empregados: taxonomia, composição e uso. No exemplo, observa-se a presença de proposições de taxonomia representadas pelas arestas rotuladas com `is` e `is a` (*e.g.*, entre `C-Adequate test set` e `Test set` e entre `Unit testing` e `Test phase`) e de proposições de composição representadas por arestas rotuladas com `has` e `is composed of` (*e.g.*, entre `Test case` e `Input data` e entre `Test set` e `Test case`). As arestas que não representam proposições de composição ou de taxonomia são classificadas como de uso (*e.g.*, a aresta rotulada com `evicts` entre `Test case` e `Fault`).

A construção de um modelo conceitual é uma atividade desenvolvida pelo especialista do domínio em conjunto com o instrutor e projetista instrucional. No caso geral, todos esses papéis são realizados pela mesma pessoa, o professor. Ela se inicia com a definição dos principais objetivos de aprendizagem a serem atingidos, representados por uma questão que dirige a construção do modelo conceitual. Por exemplo, considerando um curso sobre garantia de qualidade de software, em cuja primeira aula foram apresentados os conceitos sobre erros e uma visão geral das atividades disponíveis para detectá-los, a aula seguinte seria sobre teste de software e como essa atividade pode contribuir na garantia de qualidade. A partir dessa questão, inicia-se o estabelecimento do vocabulário específico do domínio (e desses os conceitos e proposições), expandindo-o até que se estabeleça toda a lógica responsável pela eficiência da atividade de teste.

O especialista de domínio, certamente, conhece todos os conceitos e proposições que definem o conceito central do modelo. No entanto, a especificação do modelo com base exclusivamente no conhecimento tácito apresenta riscos significativos. Um deles é a omissão: conceitos que são óbvios para o especialista não o são necessariamente para os aprendizes (que utilizarão o material produzido a partir do modelo conceitual). Um segundo motivo é a definição errônea de um conceito ou proposição, que pode ocorrer mesmo em se tratando de um especialista.

Da mesma forma que existe material didático aos quais se espera que o aprendiz utilize, também deve ser utilizado, na construção do modelo conceitual, material já existente. Não se trata de substituir o papel do especialista, mas de assegurar a qualidade do modelo gerado. Além disso, o conteúdo disponível em um material didático representa um esforço realizado por terceiros para consolidar o conhecimento de um domínio para fins educacionais. O reúso desse consenso permite que o especialista concentre-se em aperfeiçoar o modelo, ao invés de despender tempo em elementos já feitos por outras pessoas.

No contexto da técnica IMA-CID, a principal abordagem para extrair os elementos relevantes consiste na leitura e anotação de material existente, anotação essa realizada de acordo com algum modelo de informação específico para educação. Como a IMA-CID

pressupõe o uso das categorias de conhecimento da técnica CDT (MERRILL, 1983) em modelos posteriores, tais mecanismos de classificação eram adotados para a anotação. No entanto, essa estratégia possui duas deficiências. A primeira é o tempo necessário para sua execução, decorrente da execução essencialmente manual da leitura e, principalmente, da identificação e classificação dos termos. A segunda deficiência é que a extração com base em um modelo de informação específico (voltado aos aspectos instrucionais), além de ser complexa (dada as várias classes a serem consideradas), insere um viés instrucional já no modelo conceitual. Em tese, o modelo conceitual busca a neutralidade quanto à teoria educacional ou ao método instrucional utilizado.

Uma alternativa é realizar a leitura com o único propósito de identificar os conceitos e seus relacionamentos. De fato, conceitos geralmente correspondem a substantivos e as proposições com as frases que relacionam os conceitos por verbos (CANAS; NOVAK, 2008). No entanto, para esse caso é possível oferecer algum nível de automatização, identificando-se potenciais conceitos e proposições a partir do texto extraído do material. Dessa forma, evita-se o viés da abordagem anterior e se permite a realização da atividade de forma mais rápida, cabendo aos atores dessa atividade a identificação dos conceitos e proposições interessantes e acrescentar aquelas omitidas do material didático, mas de conhecimento do especialista do domínio.

O método LODM recomenda a utilização de métodos de mineração de dados textuais e de análise de texto natural, de forma similar a outros trabalhos sobre a construção automática de mapas conceituais (GAINES; SHAW, 1995; PéREZ-MARíN *et al.*, 2007; PASCUAL-NIETO *et al.*, 2008; PéREZ-MARíN; PASCUAL-NIETO, 2010). Dado um material didático, extrai-se o texto, identificando-se os termos utilizados com mais frequência. Um termo consiste de uma sequência de palavras, o que permite identificar palavras-compostas relevantes a um domínio de conhecimento (*e.g.*, *software testing*). Dessa lista de termos, são removidos aqueles de pouco interesse, tais como artigos, adjetivos e pronomes. Para cada termo, armazena-se também a frequência com que ele é utilizado. A Tabela 5.1 apresenta os resultados da aplicação desse procedimento para um texto utilizado em um curso sobre teste de software (VINCENZI *et al.*, 2010a).

Existe uma clara correspondência entre os termos identificados na Tabela 5.1 e os conceitos apresentados no modelo conceitual apresentado na Figura 5.3. Deve-se ainda destacar que esse exemplo foi o resultado obtido da extração de texto de apenas um material didático, que trata de algumas técnicas de teste de software com mais profundidade (técnicas estruturais e de mutação). A utilização de várias fontes permite a determinação dos termos de forma mais idônea, possibilitando a escolha de algoritmos mais eficientes para avaliar a importância de cada termo (*e.g.*, o TF/IDF).

Os termos identificados constituem o vocabulário (inicial) do modelo conceitual. Aqueles mais frequentes são os principais candidatos a conceitos. No entanto, ainda resta identificar as

Tabela 5.1: Termos extraídos de um texto didático sobre teste de software e suas respectivas frequências.

(a) Termos com uma palavra (1-gram).

Palavra	**Frequência**
testing	314
test	244
criterion	199
mutation	136
suite	125
case	102
mutant	92
node	63
use	63
program	59
use	63
program	59
product	56
path	53
fault	52
variable	52
...	< 52

(b) Termos com duas palavras (2-gram).

Palavra	**Frequência**
test suite	122
test case	87
testing criterion	70
mutation testing	59
mutation operator	29
structural testing	28
testing requirement	25
data flow	24
equivalence partition	23
mutation score	23
equivalent mutants	22
control data	21
testing technique	18
testing tool	16
...	< 16

proposições que relacionam esses termos. Para esse fim, utilizam-se técnicas de processamento de linguagem natural para identificar substantivos e verbos das frases extraídas dos textos, em particular das frases que contenham os termos que correspondam a conceitos. Resumidamente, o procedimento extrai as frases do texto e as palavras de cada frase (*tokenization*), identifica a provável função léxica de cada palavra (*tagging*) e a função sintática de sequências de palavras na frase (*chunking*). Ao final desse processo, as frases que apresentam como sujeito ou objeto algum dos termos identificados como conceitos possuem o verbo extraído, formando uma proposição. O tipo de proposição é identificado pelo verbo utilizado (algum daqueles que compõem o codomínio da função δ do modelo conceitual) e o rótulo é o próprio verbo. O conceito C_S é o sujeito da frase e o conceito C_T é o objeto.

Após a extração dos conceitos e proposições, todos os elementos necessários para a definição de um modelo conceitual são obtidos. No entanto, ainda falta a representação visual do modelo conceitual. Para isto, foi escolhido o formato CXL (CANAS *et al.*, 2006), que consiste em um esquema XML para documentos que definem mapas conceituais para a ferramenta CmapTools (CANAS *et al.*, 2004). A primeira transformação de modelo para modelos do método LODM consiste na transformação do modelo conceitual em um documento *Concept Mapping Extensible Language* (CXL). A transformação é relativamente simples. Cada mapa conceitual é definido por um `map`. Os conceitos são transformados em elementos `concept` filhos do elemento `concept-list`. Os rótulos das proposições são equivalentes

a elementos `linking-phrase` definidos como filhos do elemento `linking-phrase-list` e por uma sequência de elementos do tipo `connection` que definem os conceitos origem e destino. A configuração dos aspectos gráficos (fonte do texto, tamanho, posição no plano, cor, dentre outros atributos) é realizada por elementos `concept-appearance` para os conceitos e `linking-phrase-appearance` para os rótulos das proposições.

Evidentemente, a transformação de modelos mais complexos, não apresenta um resultado tão simples: a quantidade de conceitos e proposições automaticamente extraídos pode ser elevada e um mesmo conceito pode ser representado por rótulos diferentes (por exemplo, `test set` e `test suite`). Para amenizar esse problema, no procedimento de mineração de texto, pode-se empregar um dicionário de sinônimos para identificar a equivalência entre termos e permitir uma justa priorização de termos e proposições identificados. Quanto à quantidade elevada de conceitos, pode-se reduzir o percentil utilizado e, consequentemente, a quantidade de termos relevantes.

Ainda que os resultados da geração automática sejam promissores, a definição e revisão dos modelos conceituais é imperativa. No entanto, a experiência de uso da técnica torna evidente que algumas diretivas podem ser utilizadas para construir e revisar modelos conceituais (BARBOSA; SILVA, 2009; MACHADO *et al.*, 2011). Sugere-se a aplicação iterativa do seguinte conjunto de diretivas:

1. A primeira etapa da iteração é a seleção dos conceitos que serão analisados. Na primeira iteração, deve-se identificar um único conceito que represente o principal tópico do objeto de aprendizagem. Nas demais iterações, deve-se considerar todos os conceitos presentes no modelo para a análise e os conceitos mais frequentes ou relevantes que não foram incluídos durante a iteração anterior.
2. Assume-se, para o modelo conceitual, uma estrutura hierárquica, organizada em níveis (a começar em 0, correspondente à raiz da hierarquia). Os conceitos selecionados para análise e que não ainda foram incluídos no modelo conceitual devem ser dispostos no primeiro nível livre disponível.
3. Identificar as proposições que contêm dois ou mais conceitos do conjunto sob análise e classificá-las. Para a obtenção das proposições, pode-se especificá-las manualmente ou identificá-las com o auxílio da ferramenta de mineração de texto.
   3.1. Identificar relações de taxonomia (classificação). Verbos e expressões tais como "é classificado em" e "é um tipo de" podem auxiliar na determinação das relações da taxonomia.
   3.2. Identificar relações de composição. Verbos e expressões tais como "é formado por", "é composto de" e "pertence a", que associam dois ou mais conceitos, podem auxiliar na identificação de tais relações.

3.3. As proposições restantes definem algum tipo de dependência funcional entre os conceitos.

Para as proposições identificadas e classificadas, inseri-las no modelo, rotulando-as com o verbo adequado.

4. Identificar as proposições que contêm apenas um dos conceitos do conjunto e observar os conceitos relacionados. Caso eles sejam relevantes, eles devem ser inseridos no conjunto de conceitos a serem analisados na próxima iteração.

Como critério de parada das iterações, pode-se considerar que o modelo está completo quando não existem novos conceitos a serem analisados no início de uma nova iteração ou escolher um conjunto de métricas (por exemplo, quantidade de conceitos ou proposições especificadas no modelo) e como indicador a quantidade de tempo a ser alocada para a atividade de aprendizagem em que o objeto será utilizado.

Para a validação do modelo conceitual, propõe-se um conjunto de diretivas a partir das propriedades dos conceitos, a hierarquia por eles estabelecida e as proposições que os relacionam. Nessas diretivas, denomina-se um superconceito aquele que é especializado por outro conceito, conforme definido por uma proposição de taxonomia; um subconceito é todo conceito que especializa outro conceito; um conceito é composto quando ele é o sujeito de uma ou mais proposições do tipo de composição; um conceito depende de outro conceito quando na existência de proposições de função (ou seja, qualquer proposição que não seja de taxonomia ou composição).

- Para todo conceito, o conhecimento do superconceito (caso este exista) é necessário para determinar se o conceito foi atendido.
- Para todo conceito, o conhecimento do subconceito (caso este exista) não deve ser necessário para determinar se o conceito foi atendido.
- Para todo conceito, o conhecimento do subconceito (caso este exista) é suficiente para determinar se o conceito foi atendido.
- Um conceito é atendido se e somente se todos os conceitos dos quais ele é composto também são atendidos.
- Um conceito é atendido se e somente se todos os conceitos dos quais ele dependa funcionalmente também são atendidos.

5.1.2. Modelo instrucional

O modelo instrucional representa as atividades de aprendizagem e os conteúdos associados, planejados de acordo com os objetivos educacionais identificados no modelo conceitual. As atividades possuem dois aspectos: o teórico e o prático. A ênfase em cada aspecto depende do método instrucional e das teorias empregadas. Por exemplo, a IMA-CID busca uma

relação de equilíbrio entre teoria e prática, aliando: (1) os conhecimentos modelados de acordo com a teoria CDT (conceitos, princípios, fatos e procedimentos) na forma de itens de informação e (2) elementos facilitadores, de acesso e de avaliação, que demonstram e exercitam os conhecimentos (representados como elementos instrucionais).

O LODM flexibiliza a definição de elementos referentes à teoria e prática, permitindo o uso de diferentes métodos instrucionais e, ao mesmo tempo, preservando a compatibilidade com a IMA-CID. A primeira alteração realizada foi quanto à redefinição do item de informação e do elemento instrucional conforme apresentados no modelo instrucional da IMA-CID. Um item de informação associa elementos adicionais de conhecimento aos conceitos identificados no modelo conceitual e um elemento instrucional visa complementar os itens de informação, proporcionando uma melhor compreensão e assimilação do domínio de conhecimento em questão (BARBOSA, 2004, p. 82). Ocorre, dessa forma, uma classificação clara entre teoria (itens de informação) e prática (elementos instrucionais). Considerando a proposta original de utilização de conceitos, princípios, fatos e procedimentos para os itens de informação, essa divisão é adequada: os itens de informação realmente desempenham um papel de refinar o modelo de conhecimento definido no modelo conceitual, atuando no aspecto teórico do objeto de aprendizagem. No entanto, ao utilizar outros tipos de item de informação, tal como um mapa conceitual, observa-se que ele, além de desempenhar o papel de item de informação, também pode atuar como elemento instrucional, permitindo a explanação (apresentação de exemplos), exploração (acesso a novos conceitos) e avaliação (a partir da construção de mapas conceituais pelos aprendizes). Na concepção atual da técnica, não é possível representar adequadamente elementos que possuem papéis tanto teóricos quanto práticos.

O modelo definido pelo LODM reutiliza os itens de informação da abordagem IMA-CID, porém não restritos às categorias do CDT. Por outro lado, a definição de elementos instrucionais é distinta daquela da IMA-CID. No LODM, qualquer elemento utilizado no modelo instrucional é denominado de elemento instrucional. Um item de informação é, portanto, uma especialização de um elemento instrucional. A classificação em explanação, exploração e avaliação agora é denominada de finalidade instrucional (*Instructional function*). Todo elemento instrucional pode ser associado a uma ou mais finalidades instrucionais, logo também o podem os itens de informação. No entanto, alguns itens de informação possuem uma função mais definida e podem ser pré-associados a determinadas finalidades instrucionais.

De forma similar à abordagem IMA-CID, o modelo instrucional é definido como um *statechart*, desconsiderando as transições e os elementos por ela estritamente associados, conforme apresentado na Figura 5.4. A técnica de *statecharts* atende os requisitos para a modelagem de objetos de aprendizagem definidos quanto à concorrência e sincronização (referentes à interação) e hierarquia (estrutura). Além disso, *statecharts* combinam o rigor formal, herança das máquinas finitas de estados, à facilidade de representação proporcionada pela hierarquização de estados, difusão (*broadcasting*) de eventos e ortogonalidade.

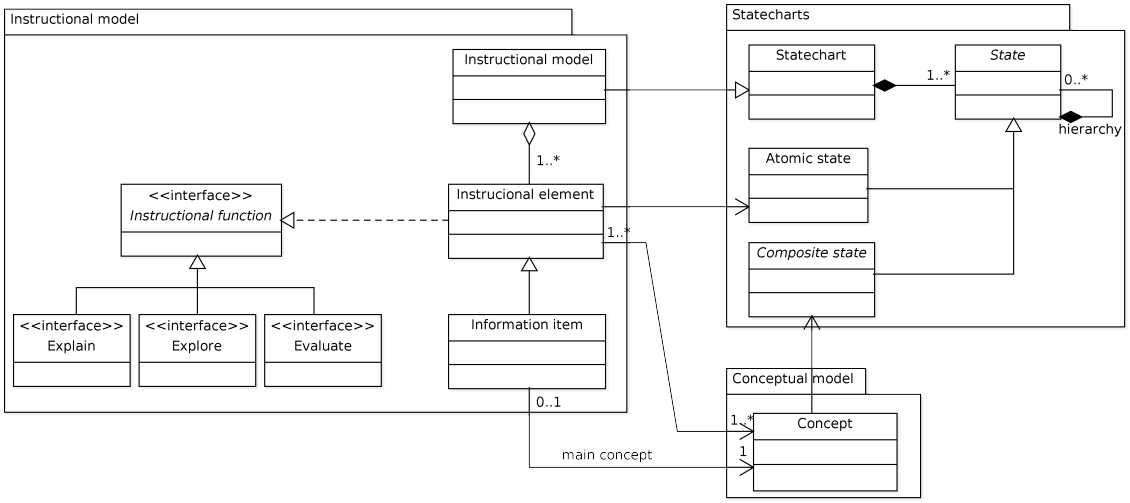


Figura 5.4: LODM: Diagrama de classe contendo os elementos do modelo instrucional.

Os elementos instrucionais são associados com estados aninhados. Em linhas gerais, define-se um estado para cada conceito definido no modelo conceitual e, para cada nível daquele modelo, aninham-se um ou mais estados. O principal (primeiro) estado do modelo instrucional (que contém todos os demais estados) corresponde ao conceito do modelo conceitual a partir do qual será conduzida a atividade de aprendizagem. Uma escolha usual é o conceito de nível 0, ou seja, o conceito raiz da hierarquia. No entanto, qualquer outro conceito pode ser utilizado. Por exemplo, considerando um curso sobre teste de software e o seu respectivo modelo conceitual (Figura 5.3), pode-se optar por iniciar o modelo por um erro observável pelo usuário de um software (`Failure`) ao invés da escolha usual de começar a atividade trabalhando o conceito de teste de software (neste caso, o elemento raiz do modelo conceitual). Essa decisão deve ser tomada de acordo com as atividades anteriores de aprendizagem e o método instrucional adotado pelo professor.

Formalmente, o modelo instrucional, considerando as definições de elemento instrucional, item de informação e finalidade instrucional acima mencionadas, é especificado como $M_I =< S, \alpha, \beta, C_I, \lambda, R, \gamma, IF, \eta, \delta, E, T >$, em que:

- S é o conjunto finito não vazio de estados;
- $\alpha : S \hookrightarrow basic, xor, and$, ou seja, α determina, para todo estado em S, o tipo dele entre básico, composto do tipo XOR (apenas um subestado ativo) ou composto do tipo AND (todos os subestados ativos).

  Nas definições a seguir, será utilizada a definição do conjunto de estados do tipo básico S_B e compostos S_C:
  - $S_B = s \in S | \alpha(s) \in basic$,
  - $S_C = s \in S | \alpha(s) \in xor, and$.
- $\beta : S_C \rightarrow \mathcal{P}_1(S)$, ou seja, β define a hierarquia de estados do *statechart*. Cada estado composto deve conter um conjunto não-vazio de estados e um estado pode ser subestado

de apenas um estado ($forall s_1, s_2 \in S_C, \beta(s_1) \cap \beta(s_2) = \emptyset$). Além disso, existe um estado composto que não é contido por nenhum outro estado, estado esse denominado raiz $root$ ($root \in S_C \wedge \nexists s \in S_C | root \in \beta(s)$);

- C_I é um subconjunto dos conceitos em C definidos no modelo conceitual;
- $\lambda : C_I \hookrightarrow S_C$, ou seja, cada conceito tratado no modelo instrucional é associado a um estado composto;
- R é o conjunto de elementos instrucionais do modelo;
- $\gamma : S_B \hookrightarrow R$, ou seja, cada estado básico está relacionado com um elemento instrucional;
- IF é conjunto de funções instrucionais (inicialmente definido como $explain, explore, evaluate$);
- $\eta : S_B \hookrightarrow IF$, ou seja, todo estado básico e, consequentemente, o elemento instrucional relacionado no contexto daquele estado, está associado a uma função instrucional.
- δ é a função que define o estado inicial de cada estado composto do tipo OR, tal que $\delta : s \in S_C | \alpha(s) = or \hookrightarrow S$, ou seja, o estado inicial de um estado composto do tipo OR pode ser qualquer estado (básico ou composto) e todo composto do tipo OR deve ter um estado inicial;
- E é o conjunto de eventos. Para um modelo instrucional, a entrada e a saída de um estado marcado com o estereótipo de elemento instrucional causam eventos cujo valores são os nomes dos conceitos relacionados ao conceito.
- T é o conjunto de transições entre os estados. Uma transição realiza a transferência de um conjunto de estados para outro conjunto de estados, sendo habilitada pela ocorrência de um evento e disparada quando a condição a ela associada é satisfeita. Após finalizado o disparo (transição), executa-se a ação associada com a transição. No modelo instrucional, não são definidas condições e ações. Ela é definida por uma tupla $< T_S S, T_T S, T_E, T_C, T_A >$, em que:
  - $T_S S = \mathcal{P}_1 S$ é o conjunto não vazio de estados do qual a transição é originada,
  - $T_T S = \mathcal{P}_1 S$ é o conjunto não vazio de estados para o qual a transição destina-se,
  - $T_E = \mathcal{P} E$ é o conjunto de eventos que habilitam a transição.

Esse modelo é uma simplificação do *statechart* definido por Harel (1987) e, por conseguinte, daquele adotado para a máquina de estados da *Unified Modeling Language* (UML). Por esse motivo, adota-se a UML para a representação gráfica do modelo. Para a associação de elementos instrucionais e a designação de funções instrucionais aos estados, empregam-se estereótipos. Os estereótipos são definidos em um perfil UML denominado *LODM Instrucional Profile*, conforme apresentado na Figura 5.5: *InstructionalElement*, *InformationItem* e *InstructionalFunction*. O primeiro é aplicável a estados básicos e define os valores (*tagged values*) para o identificador e o tipo de mídia do recurso associado ao elemento (SILVA *et al.*, 2011a). O estereótipo `InstructionalFunction` define a função O estereótipo `InformationItem` estabelece que o estado define um item de informação.

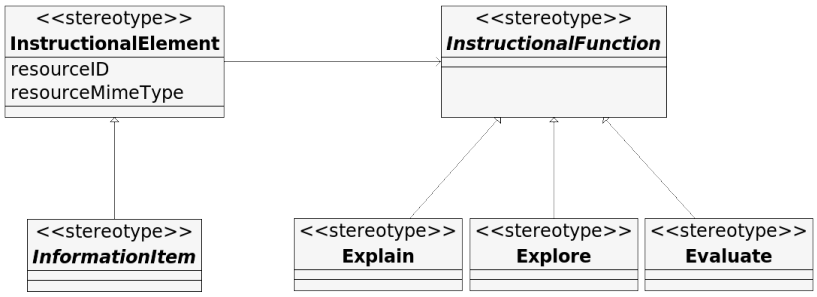
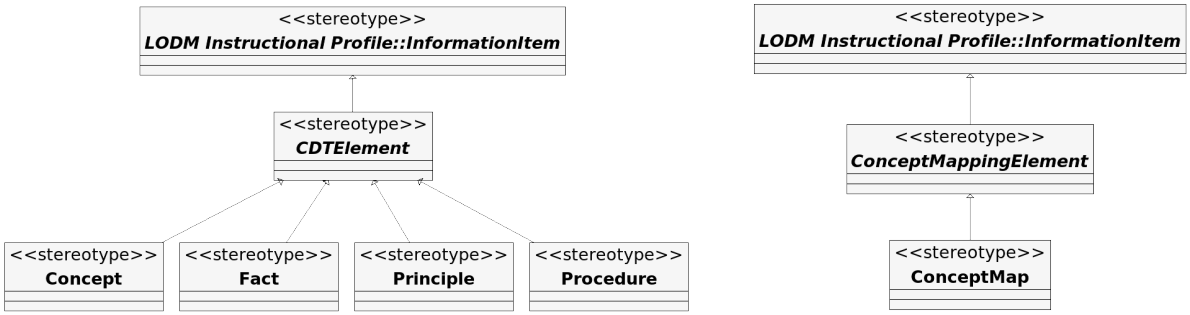


Figura 5.5: LODM: Diagrama dos elementos do perfil instrucional.

O estereótipo *InstructionalFunction* é abstrato, não podendo ser utilizado diretamente em modelos instrucionais. As especializações (concretas) atualmente disponíveis dele são aquelas que correspondem ao elemento instrucional da IMA-CID: explanatório (`Explain`), exploratório (`Explore`) e de avaliação (`Evaluate`). O estereótipo `InformationItem` também é abstrato, dado que, quando um elemento instrucional tem o propósito de refinar o modelo de conhecimento sobre um determinado conceito, o meta-modelo utilizado deve ser especificado. Por exemplo, para o LODM, são utilizados dois: o CDT (tal como a IMA-CID) e o mapa conceitual. Para cada modelo, define-se um novo perfil, tal como apresentado na Figura 5.6: um estereótipo abstrato rotulado com o nome da técnica deve estender `InformationItem` (do perfil `LODM Instructional`) e dele devem ser definidos estereótipos (não mais abstratos) para cada constructo permitido pela técnica.



(a) Perfil da técnica CDT para itens de informação.

(b) Perfil da técnica de mapeamento conceitual para itens de informação.

Figura 5.6: LODM: Perfis de itens de informação aplicáveis a um modelo instrucional.

Considerando o modelo conceitual definido em Figura 5.3, um possível modelo instrucional, definido com os perfis definidos pelo método LODM, é aquele apresentado na Figura 5.7. Neste caso em específico, definiram-se apenas os elementos instrucionais referentes a erros (`Fault`), omitindo-se os demais em `Test case`, `Test technique` e `Test phase`. Para o engano (`Mistake`), definem-se dois elementos instrucionais, sendo um deles um item de informação que representa um conceito, conforme definido no perfil CDT, e o outro um elemento instrucional que será utilizado para fins explanativos. Por ser o estado que os contém (`Mistake`) do tipo XOR, ambos os elementos instrucionais serão ativos e apresentados

simultaneamente. Para `Fault` e `Error`, apresentem-se apenas os conceitos. Em `Failure`, além do conceito, utiliza-se o caso (real) do Therac-25 (LEVESON; TURNER, 1993) para demonstrar os efeitos que um erro pode causar, atribuindo-se o estereótipo de elemento instrucional a esse estado.

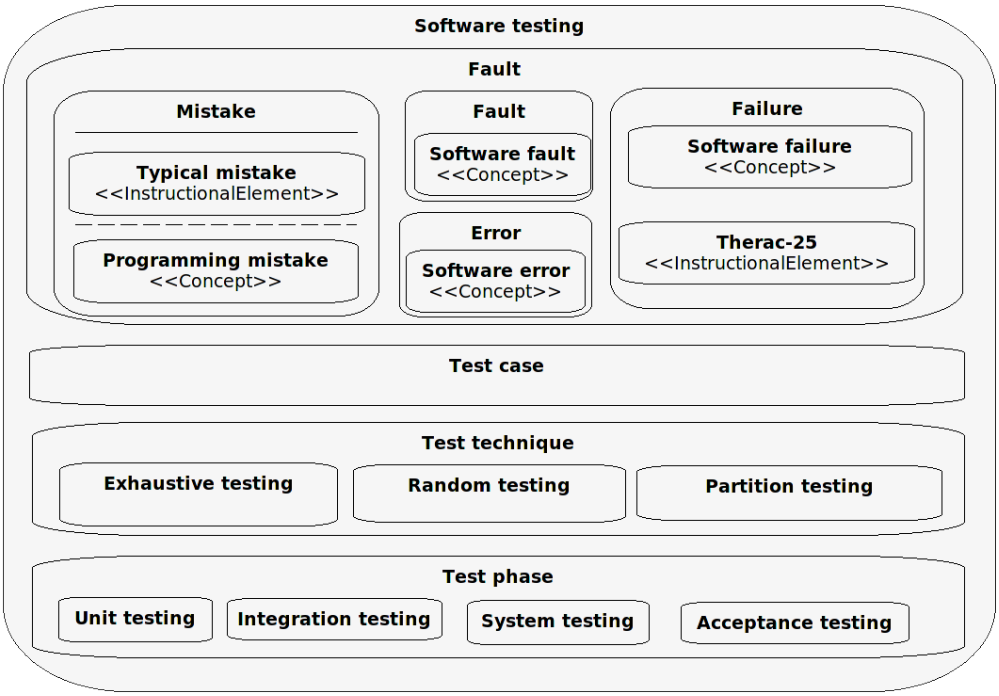


Figura 5.7: Modelo instrucional para um objeto de aprendizagem sobre teste de software.

Os valores (*tagged values*) definidos pelos estereótipos são definidos com um identificador e tipo de recurso associado. Por exemplo, pode-se utilizar endereços (URL) de uma página ou serviço Web como identificador de um item de informação, tal como apresentado na Figura 5.8 para o estado `Software failure`.

De forma similar, é possível a geração automática de modelos instrucionais a partir de um modelo conceitual. No entanto, para este modelo não se recorre a técnicas de mineração de textos, mas em diretivas estabelecidas após a criação de diversos objetos de aprendizagem (e respectivos modelos). Os procedimentos definidos nas diretivas devem ser executados de forma iterativa até que todos os conceitos do modelo conceitual sejam utilizados e que nenhum outro elemento instrucional seja alterado na iteração.

1. Defina o conjunto de conceitos e de elementos instrucionais que serão analisados.
   Na primeira iteração, escolha o conceito com que se iniciará o objeto de aprendizagem. Conforme dito anteriormente, em uma abordagem clássica e com ênfase teórica, pode-se assumir que esse conceito será o do primeiro nível do modelo conceitual. Em abordagens com tendências construtivistas (por exemplo, que adotam uma abordagem dirigida a problemas), geralmente se opta por conceitos em níveis mais altos.

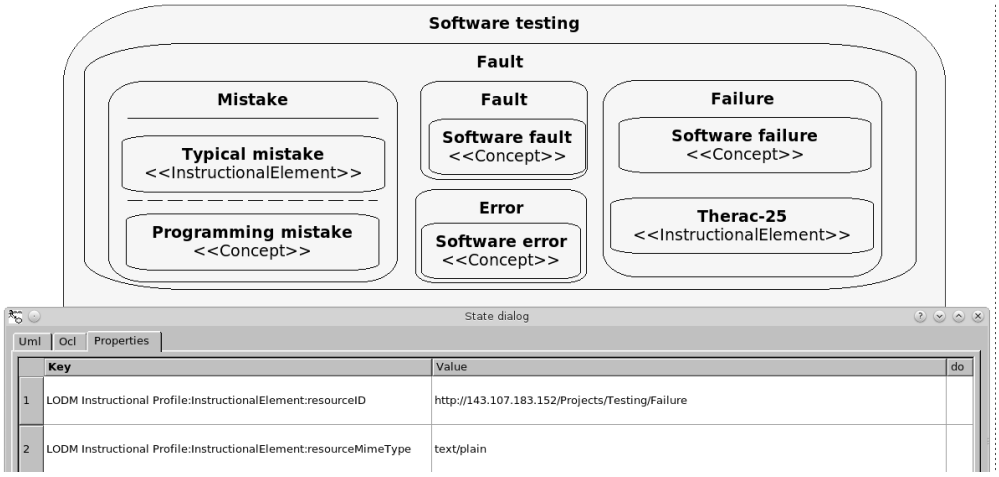


Figura 5.8: Valores definidos para os *tagged values* associados ao item de informação `Software failure` do modelo instrucional da Figura 5.7.

Nas demais iterações, escolha os conceitos que foram inseridos na iteração anterior (ou seja, os conceitos relacionados a estados anônimos).

2. Para cada conceito C sob análise e que não tenha um estado que trate do conceito, crie um estado S_C composto do tipo AND no superestado SS_C (ou, caso não exista um superestado, simplesmente um estado composto do tipo AND).
3. Para cada estado S_C (os que foram no passo anterior ou na iteração anterior), especifique:
   3.1. Caso não existam conceitos C' relacionados com o conceito C tratado pelo estado S_C, defina-o como um estado composto do tipo AND.
   3.2. Caso existam conceitos C relacionados, crie um estado composto S'_C do tipo XOR com o nome do conceito. Esse estado deve ser identificado como o estado inicial (*default*) de S_C.
   Ainda quanto ao estado S'_C:
      3.2.1. defina uma nova região em S'_C para cada conceito C' relacionado com o conceito C por proposições de uso ou de composição. Para cada conceito C':
         3.2.1.1. defina um estado básico $S'_{C'}$ com um elemento instrucional ou um estado composto do tipo XOR com vários estados básicos (e elementos instrucionais) na nova região em S'_C, com a restrição de que todos os elementos instrucionais estejam relacionados, simultaneamente, com o conceito C e C';
         3.2.1.2. Caso já exista no modelo um estado S'_C que represente o conceito C', defina uma transição de $S'_{C'}$ para $S_{C'}$.
         3.2.1.3. Caso não exista, crie um estado $S_{C'}$ aninhado no estado S_C e defina uma transição entre o estado $S'_{C'}$ para o estado $S_{C'}$ e habilitada por um evento de valor C'.
      3.2.2. caso exista algum conceito C' relacionado com C por uma proposição de

taxonomia, defina uma nova região em S'_C e, nessa região, defina um estado $S'_C children$ composto do tipo XOR. Para cada conceito C':

3.2.2.1. defina um estado básico $S'_{CchildrenC'}$ com um elemento instrucional que relacione C e C'=;

3.2.2.2. defina um estado $S_{C'}$ composto do tipo AND em S_C;

3.2.2.3. defina uma transição entre $S_{CchildrenC'}$ e $S_{C'}$, que é habilitada por um evento de valor C'.

Tal como no modelo conceitual, as diretivas são um conjunto de boas práticas para a construção de modelos instrucionais, mas formas distintas para a organização dos elementos instrucionais podem ser utilizadas.

5.1.3. Modelo de interação

O modelo de interação define a interação com o objeto de aprendizagem a partir de sua representação no modelo instrucional. De acordo com a plataforma de execução, o conjunto de eventos possíveis é ampliado, compreendendo aqueles inerentes ao ambiente de execução, aos dispositivos de interação utilizados pelos usuários e às próprias mídias utilizadas. Por exemplo, o acionamento de botões em um controle remoto é repassado para o objeto de aprendizagem na forma de eventos; eventos podem ser transmitidos junto ao fluxo de vídeo de um programa de televisão digital; ações associadas a transições executadas no modelo também podem emitir eventos.

O modelo de interação atua de forma semelhante ao modelo didático da IMA-CID. Acompanhando a Figura 5.9, observa-se que ele também define o estado composto XOR DD, que permite que a ativação de um estado dentre aqueles definidos em um estado composto XOR seja definido pelo usuário. No entanto, uma importante adequação foi realizada com o propósito de adequá-lo para o cenário de uso típico de televisão: um usuário que aciona o programa e espera que exista uma execução do conteúdo sem sua intervenção direta. Para atender esse requisito, são estabelecidas as seguintes regras para o modelo de interação:

- Sempre deve existir uma sequência de eventos, emitidos internamente pelo modelo, que permita a ativação de um dado estado do modelo.
- Todo estado composto XOR, DD ou não, deve indicar o estado inicial (*default*).

Um estado XOR DD, quando existente, vai ter prioridade quanto à definição do subestado a ser ativado. No entanto, a todo estado XOR DD estará associado um temporizador que, ao expirar, vai recorrer ao uso da função de história para determinar o estado a ser ativado e, na ausência dessa, à função *default*.

Os estados que estão associados a elementos instrucionais cujo tipo de dado seja

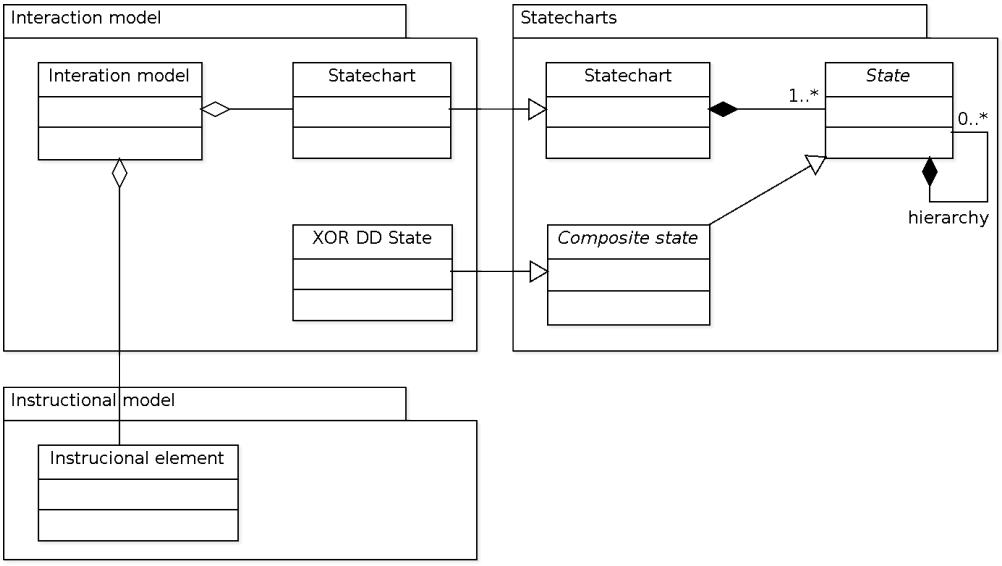


Figura 5.9: LODM: Elementos do modelo de interação.

atemporal, ou seja, que não defina um prazo para expiração, também terão um temporizador automaticamente definido. Essa alteração de comportamento do *statechart* é definida pela redefinição do estado, implementando-se esse comportamento.

Para representar o comportamento alterado do *statechart* (com a utilização dos estados com temporizadores de acordo com o tipo de elemento instrucional) e os estados XOR DD, será utilizado o perfil *LODM Interaction Profile* (apresentado na Figura 5.10). Dois estereótipos são definidos: o `DD` identifica os estados XOR DD, especificando o comportamento anteriormente mencionado; e o `ExpiringStatechart` define o novo comportamento padrão dos estados, permitindo a execução automática do objeto de aprendizagem quando na ausência de interações (e geração de eventos) externas por parte do usuário.

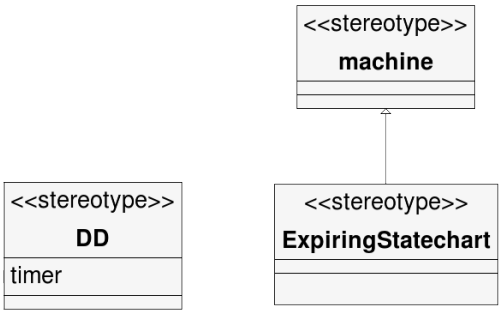


Figura 5.10: LODM: Diagrama dos elementos do perfil de interação.

Considerando o modelo instrucional definido em Figura 5.7, um possível modelo de interação, definido com o perfil *LODM Interaction Profile*, é aquele apresentado na Figura 5.11. Neste caso, todos os estados XOR foram definidos como XOR DD, conforme denotado pelo estereótipo `DD` acrescentados aos estados que, no modelo instrucional, eram apenas XOR. Além disso, para o *statechart*, atribuiu-se o estereótipo `ExpiringStatechart`, que estende o

estereótipo `machine` (o estereótipo padrão de um *statechart* em UML).

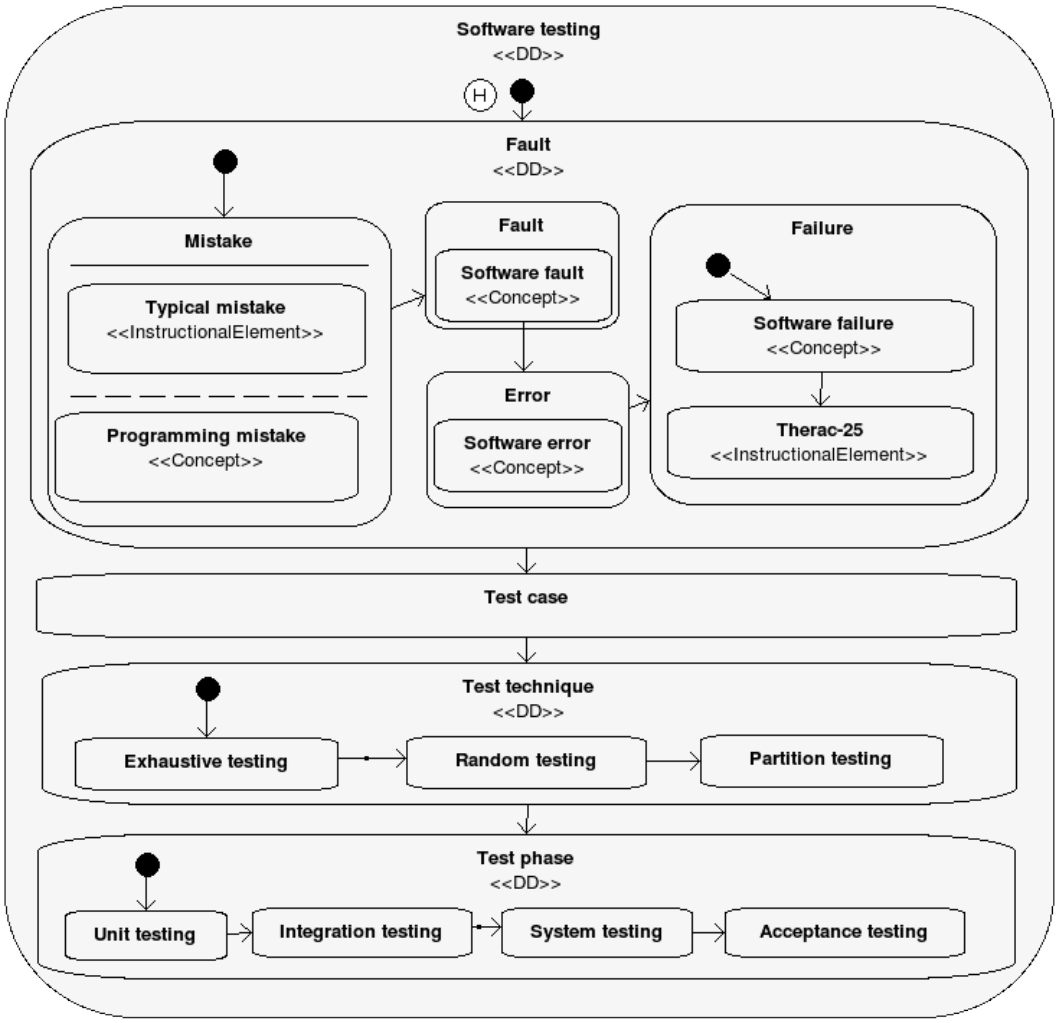


Figura 5.11: Modelo de interação para um objeto de aprendizagem sobre teste de software.

Formalmente, o modelo de interação é uma extensão do modelo instrucional, acrescentando-se mais eventos ao conjunto de eventos e se definindo condições e ações para as transições. Formalmente ele é especificado como um *statechart* $M_{UX} = S, \alpha, \beta, C_I, \lambda, R, gamma, IF, \eta, \delta, \epsilon, T, C, E, A >$, que se diferencia de M_I quanto aos seguintes termos:

- ϵ é a função parcial que define o estado inicial de estado composto com base no conjunto de estados que estavam ativos (h) ou última configuração conhecida ($h*$) , tal que $\epsilon : S_C \rightarrowtail\!\!\!\!\!\rightarrow h, h*$;
- C é o conjunto de condições;
- E é o conjunto de eventos. Além dos eventos definidos no modelo instrucional, esse conjunto contém os eventos suportados pela plataforma de execução e os específicos do modelo (e lançados a partir das ações).

- A é o conjunto de ações. As ações podem alterar valores ou disparar eventos. Por padrão, ao entrar e sair de um estado, são executadas as ações `enter` e `exit`, as quais emitem, respectivamente, os eventos `enter[state]` e `exit[state]`;
- T é o conjunto de transições entre os estados. Uma transição realiza a transferência de um conjunto de estados para outro conjunto de estados, sendo habilitada pela ocorrência de um evento e disparada quando a condição a ela associada é satisfeita. Após finalizado o disparo (transição), executa-se a ação associada com a transição. Ela é definida por uma tupla $< T_S S, T_T S, T_E, T_C, T_A >$, em que:
  - $T_S S = \mathcal{P}_1 S$ é o conjunto não vazio de estados do qual a transição é originada,
  - $T_T S = \mathcal{P}_1 S$ é o conjunto não vazio de estados para o qual a transição destina-se,
  - $T_E = \mathcal{P}E$ é o conjunto de eventos que habilitam a transição,
  - $T_C = \mathcal{P}C$ é o conjunto de condições que devem ser satisfeitas para disparar a transição,
  - $T_A = \mathcal{P}A$ é o conjunto de ações que devem ser executadas quando disparada a transição.

No modelo instrucional, define-se que o conjunto inicial de eventos é composto por conceitos (do modelo conceitual) representados pelo estado, além daqueles eventos padrões de um *statechart* (referentes à entrada e à saída de um estado). No modelo de interação, acrescentam-se aqueles eventos frutos da interação do usuário com o objeto de aprendizagem e a plataforma de execução.

5.1.4. Geração de objetos de aprendizagem

A geração dos objetos de aprendizagem consiste na transformação do modelo de interação M_{UX} em um pacote contendo o objeto de aprendizagem para a plataforma alvo desejada. Nesta tese, duas plataformas foram definidas como alvo: no início, enquanto as ferramentas para desenvolvimento e execução de aplicativos para TVD eram imaturas, foram gerados objetos de aprendizagem na forma de apresentações multimídia; posteriormente, foram gerados objetos na forma de aplicações para TVD, mais especificamente para a plataforma Ginga NCL.

Durante a geração, todos os elementos do modelo de interação precisam ser representados por artefatos. Considerando-se o modelo M_{UX}, seus elementos são realizados da seguinte forma:

- Cada estado composto do conjunto finito não vazio de estados S é transformado em um contexto. Os contextos definem os conteúdos acessíveis em um determinado instante no objeto de aprendizagem.

- Os estados básicos contêm os elementos instrucionais, cuja apresentação é de responsabilidade da plataforma de execução do objeto de aprendizagem, que escolhe o reprodutor adequado de acordo com o tipo de mídia associado com o elemento.
- Para os estados compostos do tipo XOR DD, mecanismos de acesso aos contextos definidos pelos estados aninhados devem ser fornecidos.
- Para todas as transições definidas no modelo, devem ser oferecidos mecanismos de acesso aos contextos.
- A disponibilização, habilitação e execução das transições devem obedecer aos eventos e condições que as habilitam e permitem seu disparo.
- As ações associadas às transições devem ser implementadas em uma linguagem suportada pela plataforma.

Além da transformação dos elementos do modelo de interação em objetos de aprendizagem para a plataforma destino, é necessário definir a melhor forma de implementação das principais características do objeto: a navegação (para acesso aos contextos), a apresentação dos conteúdos multimídia e a implementação das ações das interações.

Os estados compostos do tipo AND servem para estabelecer o sequenciamento das atividades de aprendizagem e de entrega dos elementos instrucionais. Em apresentações multimídia, esses componentes são representados por menus localizados nas laterais. No caso de documentos NCL para televisão digital, tal disposição é inapropriada, dado que, à distância a legibilidade do menu seria inadequada e os mecanismos para acionamento dos itens, utilizando dispositivos de interação limitados (tal como o controle remoto) não possuem resolução suficiente para escolher áreas da tela. Nesse caso, opta-se por mostrar o conteúdo do estado inicial do próximo contexto (lembrando que, no modelo de interação, sempre existe um próximo estado inicial) na tela principal e, em áreas reduzidas, nas borda da tela (tal como nos recursos PIP), apresentar o conteúdo dos demais contextos (sem o som). O usuário pode então escolher, com os botões direcionais, os outros contextos e, se desejar, escolhê-lo.

Em relação à apresentação e estética, enquanto multimídia os objetos de aprendizagem utilizam, principalmente, textos, imagens, sons e vídeos. Embora os modelos forneçam diretivas quanto à disposição e à quantidade desses recursos, é necessário ajustar manualmente o arranjo deles na tela de modo que o usuário consiga interpretar corretamente os dados sendo transmitidos. Em relação ao áudio, o cuidado a ser tomado é principalmente quanto ao volume: alto o suficiente para ouvir, mas sem discrepâncias entre os diferentes áudios utilizados no objeto de aprendizagem. Quanto ao texto, é prudente utilizar um tamanho de fonte adequado para o público-alvo e para o dispositivo de saída utilizado.

As ações associadas às transições do modelo de interação devem ser implementadas em uma linguagem suportada pelo ambiente de execução. Em apresentações multimídia, na plataforma alvo escolhida para este trabalho (apresentações PDF gerados a partir de

especificações LaTeX), não é recomendável a programação de ações, dado que o único leitor de PDF que suporta essa característica é o Adobe Acrobat (disponível em várias plataformas, mas com restrições no Linux). Em TVD, é possível definir ações em Lua, Java ou ECMAScript. A opção recomendada pelos desenvolvedores do SBTVD é o Lua para aplicações declarativas e Java para as procedurais. Na Seção 5.3 são detalhados os procedimentos para transformações e no Capítulo 6 são apresentados resultados das transformações.

5.2. LODP: Learning Object Development Process

O processo de desenvolvimento de objetos de aprendizagem envolve diversos atores de áreas distintas de conhecimento em equipes ou comunidades de níveis distintos de maturidade. Sabe-se que um processo bem definido influencia positiva e significativamente a qualidade dos produtos gerados, mas também é de conhecimento de que não existe um processo universal, que possa ser aplicado para qualquer situação. No entanto, é possível definir modelos de referência de processo, ou processos-padrão, que sejam adaptáveis para cada comunidade e para cada projeto.

O primeiro modelo de processo para desenvolvimento de objetos de aprendizagem definido nesses moldes foi o SP-DEM (BARBOSA; MALDONADO, 2006b), apresentado na Seção 2.3.2.1. Ele define um modelo a partir do arcabouço de processos-padrão da norma ISO/IEC 12207:1995 (ISO/IEC, 1995). Na próxima seção, a partir do SP-DEM, dos requisitos levantados nos capítulos anteriores e com base na versão atual das normas ISO/IEC 12207 (ISO/IEC, 2008a) (específica para software) e ISO/IEC 15288 (ISO/IEC, 2008b) (voltada para sistemas), da qual a ISO/IEC 12207 é uma especialização, define-se um processo padrão para o desenvolvimento de objetos de aprendizagem denominado SPLOD (*Standard Process for Learning Objects Development*). A partir desse, define-se um processo, denominado LODP, voltado para o desenvolvimento de objetos de aprendizagem multimídia e interativos, com foco em televisão digital.

5.2.1. SPLOD: modelo de referência para objetos de aprendizagem

Um modelo de referência de processo, ou processo padrão, define todos os processos necessários ao desenvolvimento de um produto, considerando desde sua concepção até a retirada de uso. Representado na Figura 5.12, ele define grupos de processos (`Process`) agrupados por sua principal finalidade. Cada processo define um conjunto de resultados observáveis (`Artefact`) e atividades (`Activity`). As atividades recebem recursos gerados por outras atividades ou artefatos gerados pelos processos e geram uma saída como resultado da execução de uma

coleção de tarefas (`Task`). Cada tarefa corresponde a um requisito, recomendação ou ação relacionada à execução da atividade (ISO/IEC, 2008a).

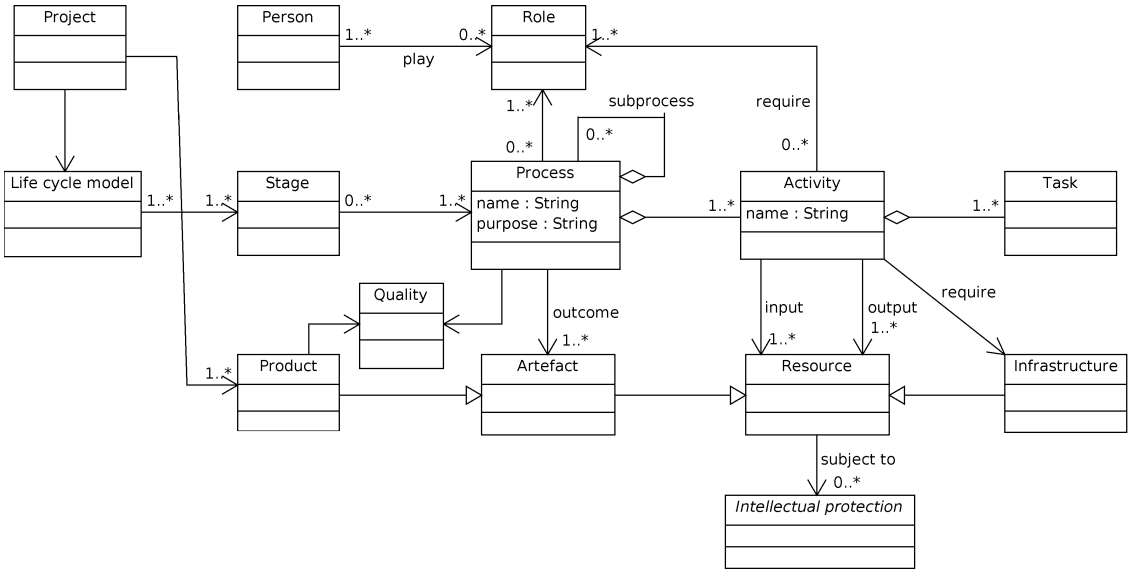


Figura 5.12: Elementos de um modelo de referência para processos.

A organização dos grupos de processos, processos, partes e atividades observa dois princípios: a modularidade e a responsabilidade. Modularidade significa que cada elemento possui alta coesão e baixo acoplamento. Para um grupo de processo, isso se traduz em cada grupo ser responsável por um interesse bem definido do processo (alta coesão) e com uma dependência bem definida e o quão reduzida possível quanto aos outros grupos para conduzir os processos e atividades nele contidos. Esse princípio se aplica recursivamente, até o elemento mais simples, que são as tarefas.

Observando esses princípios e as normas ISO 15288:2008 e 12207:2008, o *Standard Process for Learning Objects Development* (SPLOD) organiza-se em quatro grupos de processos, conforme apresentado na Figura 5.13: acordo (`Agreement`), organizacionais (`Organizational project-enabling`), projeto (`Project`) e técnicos (`Technical`). Nas próximas subseções, cada um desses grupos de processos será descrito, definindo-se suas atividades e o relacionamento com as normas ISO em questão.

5.2.1.1. Processos de acordo

Os processos de acordo (`Agreement processes`) definem a relação entre as organizações envolvidas no desenvolvimento de um objeto de aprendizagem. Este grupo de processo organiza-se em processo de aquisição (`Acquisition process`) e de fornecimento (`Supply process`), tal como apresentado na Figura 5.14.

O processo de aquisição possui o propósito de obter um produto ou serviço de acordo com um conjunto de requisitos. O produto é o objeto de aprendizagem e os requisitos

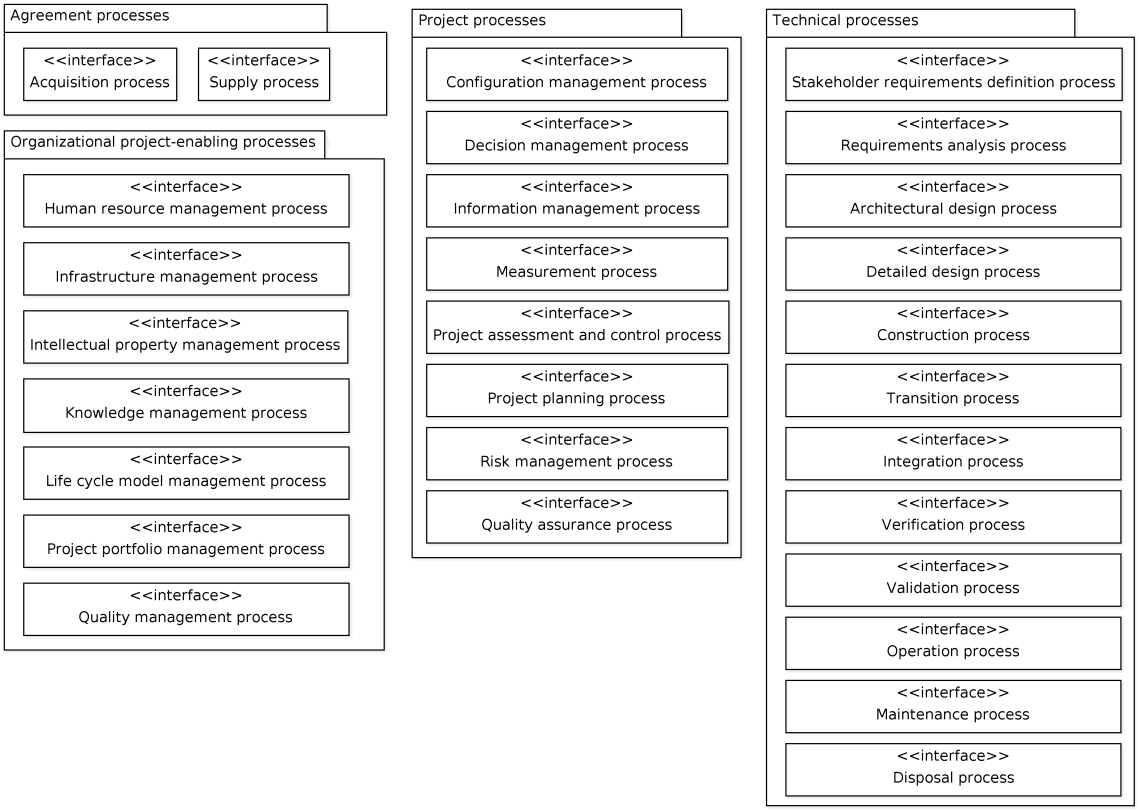


Figura 5.13: SPLOD: Visão geral do modelo de referência para desenvolvimento de objetos de aprendizagem.

compreendem os objetivos de aprendizagem e demais elementos relacionados às características do projeto instrucional desejado: formalidade, modalidade, forma de acesso, avaliações e interatividade. Os resultados esperados são: definição de uma estratégia de aquisição, seleção de fornecedores, estabelecimento de um canal de comunicação com o fornecedor, definição de um acordo sobre a aquisição do produto ou serviço, recebimento do produto ou serviço e pagamento.

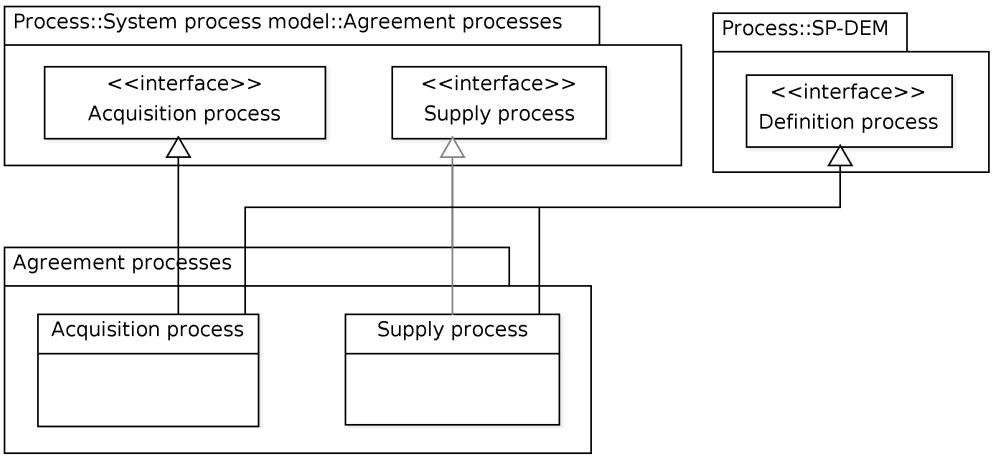


Figura 5.14: SPLOD: Grupo de processos de acordo e relação com os processos das normas ISO 15288:2008/12207:2008 e com o SP-DEM.

O processo de fornecimento tem o propósito complementar ao de aquisição: fornecer um produto ou serviço de acordo com os requisitos definidos pelo adquirente. Os resultados esperados são: identificação de consumidores do produto ou serviço ou atendimento a um requerente, envio de proposta ao consumidor ou requerente, estabelecimento de um canal de comunicação com ele, definição de um acordo sobre o fornecimento do produto ou serviço, fornecimento do produto e pagamento.

Em educação, a tendência é a execução dos dois processos simultaneamente e de forma única, criando não uma relação cliente-fornecedor, mas uma comunidade que ensina e aprende simultaneamente, com o compartilhamento de objetos de aprendizagem e a adaptação para cada contexto (e não com a simples aquisição de objetos de aprendizagem prontos). Mesmo nos casos dos cursos abertos, existe uma relação de fornecimento para as universidades por parte dos alunos na forma de correções, esclarecimentos e definição de novas atividades para complementar as já definidas e por parte de outras instituições que adaptam o material e criam os seus próprios cursos abertos. Dessa forma, uma organização não pode se dispor apenas a fornecer sem colocar-se na posição de consumir. Trata-se de uma via de duas mãos, necessária para formar uma comunidade de aprendizagem.

No processo padrão SP-DEM, define-se o *processo de definição* (*definition process*) para determinar problemas de aprendizado a serem solucionados, identificando as necessidades do adquirente, definir requisitos do interessado, analisar a viabilidade do projeto, determinar o escopo, construir repositório de terminologias, documentar o processo, revisar e aprovar o projeto. No entanto, algumas dessas atividades pertencem a processos dos grupos organizacional e de projeto. De fato, apenas a primeira atividade precisa ser tratada no escopo do estabelecimento dos acordos que definem o projeto. Assim, o processo de definição do SP-DEM é incorporado nos demais processos no SPLOD.

Entretanto, é importante ressaltar que, embora isso seja válido para o desenvolvimento do produto "objeto de aprendizagem", existem outros produtos ou serviços necessários para a implantação do objeto e que estão fora do escopo desta tese, embora importantes para realizar o ciclo de vida de um objeto de aprendizagem. Por exemplo, os ambientes virtuais de aprendizagem são um elemento distinto, porém necessário, para a execução de objetos de aprendizagem. Requisitos quanto aos padrões de empacotamento, sequenciamento e metadados podem ser implementados de forma independente do objeto. O mesmo se aplica ao ambiente de execução dessas implementações, em servidores de terceiros ou em plataformas de computação nas nuvens (*cloud computing*). Nesse caso, adotam-se os processos e atividades previstos na ISO 12207:2008.

5.2.1.2. Processos organizacionais

O grupo de processos organizacionais (*Organizational Project-Enabling Processes*) garante a capacidade da organização em executar – iniciar, apoiar e controlar – projetos de desenvolvimento de objetos de aprendizagem. Tal capacidade é assegurada pela definição e manutenção de políticas, processos, modelos de ciclo de vida e procedimentos necessários para a criação e a execução de novos projetos. Conforme exposto na Figura 5.15, são cinco os processos que compõem este grupo: gerenciamento de modelos de ciclo de vida, gerenciamento de infraestrutura, gerenciamento de portfólio de projetos, gerenciamento de recursos humanos e gerenciamento de qualidade.

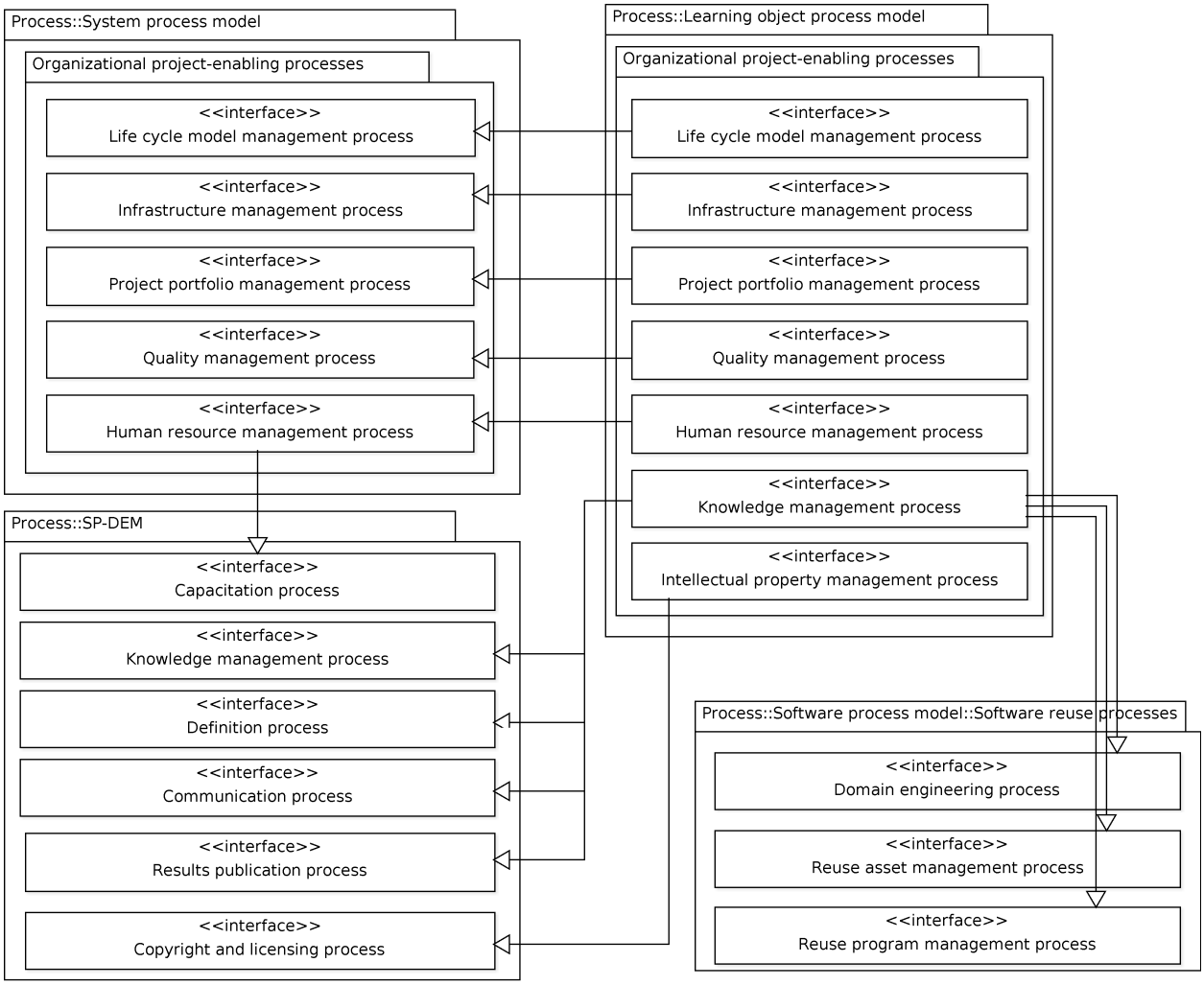


Figura 5.15: SPLOD: Grupo de processos organizacionais e relação com os processos das normas ISO 15288:2008/12207:2008 e com o SP-DEM.

O **processo de gerenciamento de ciclo de vida** (*Life cycle model management process*) define, mantém e garante a disponibilidade das políticas, processos, modelos e procedimentos utilizados durante o ciclo de vida de um objeto de aprendizagem em função dos objetivos da organização. Os resultados esperados são políticas e procedimentos para a

implantação de modelos de ciclo de vida e de processos; definição de responsabilidades para o gerenciamento do ciclo de vida; definição, implantação e melhoria dos modelos, processos e procedimentos de ciclo de vida; priorização da implementação de melhorias nos processos, modelos e procedimentos. Esses resultados são frutos de atividades de estabelecimento, avaliação e melhoria do processo de gerenciamento do ciclo de vida do desenvolvimento de objetos de aprendizagem.

O **processo de gerenciamento de infraestrutura** (*Infrastructure management process*) define os recursos físicos e serviços necessários para a execução de projetos de desenvolvimento de objetos de aprendizagem. Os resultados esperados são a identificação dos requisitos de infraestrutura e a seleção, desenvolvimento, aquisição, instalação e manutenção de elementos de infraestrutura que atendam os requisitos estabelecidos.

O **processo de gerenciamento de portfólio de projeto** (*Project portfolio management process*) define o início e a manutenção de projetos para atender aos objetivos estratégicos da organização. Os resultados esperados são a identificação de oportunidades e necessidades de investimento em projetos, a alocação de recursos, a austeridade dos projetos em execução e o cancelamento de projetos. Nenhum projeto é iniciado sem antes verificar se ele atende ao propósito da organização e se esta possui os recursos para sua execução. Toda atividade, infraestrutura e recurso humano possui um custo que deve ser contabilizado para monitorar a austeridade do projeto.

O **processo de gerenciamento de recursos humanos** (*Human resource management process*) provê e capacita recursos humanos. Os resultados esperados são a identificação das habilidades necessárias para a execução de um projeto, a contratação e a capacitação de recursos humanos, a alocação de pessoas para os projetos, gerenciamento de conflitos entre as demandas de vários projetos e a reunião, compartilhamento e reutilização de conhecimento. Em relação ao SP-DEM e de acordo com as normas ISO 15288:2008 e 12207:2008, este processo substitui o *processo de capacitação*. Especificamente para o SPLOD, este processo incorpora o *processo de treinamento* do SP-DEM, que trata da capacitação no nível organizacional (e não apenas específico aos projetos), identificando carências de caráter não-técnico (*e.g.*, gerenciais) e desenvolvendo programas de treinamento e contratação quando necessário.

Observa-se a existência de atividades de gerenciamento de conhecimento – aquisição, compartilhamento e reúso de conhecimento das pessoas – na definição do processo de gerenciamento de recursos humanos da ISO 15288:2008. Embora exista relação entre treinamento e os conhecimentos das pessoas, os mecanismos para que a base de conhecimentos seja formada envolve o controle de artefatos e mecanismos de comunicação, violando o princípio de coesão e responsabilidade que a própria norma utiliza como base. Assim, tal como no SP-DEM, definiu-se um **processo de gerenciamento de conhecimento** (*Knowledge management process*) cujo propósito é permitir a comunicação das pessoas para

o compartilhamento de conhecimento, a captura do conhecimento tácito em um formato explícito e documentado, e a recuperação desse conhecimento. Os resultados esperados do processo são: identificação, criação e manutenção de meios e protocolos de comunicação; definição de planos para reúso de conhecimento; identificação de fontes de conhecimento (pertencentes ou não à organização). Por exemplo, pode-se definir uma rede social dos indivíduos que desenvolvem e utilizam um objeto de aprendizagem, com o compartilhamento dos conteúdos e modelos em *wikis* e com a criação de fóruns para discussão de dúvidas quanto ao desenvolvimento e ao uso de objetos de aprendizagem. Finalmente, o propósito e as atividades desse processo substituem os processos de *comunicação* e *publicação de resultados* e a atividade de definição de terminologias do *processo de definição* do SP-DEM.

O **processo de gerenciamento de qualidade** (*Quality management process*) assegura que os produtos do projeto e os processos implementados tenham atributos mínimos de qualidade. Os resultados esperados são a definição de políticas, procedimentos e objetivos de qualidade (tanto do produto quanto do processo), a definição da responsabilidade da verificação da qualidade, o monitoramento da satisfação dos clientes e a tomada de ações quando forem observados desvios quanto aos objetivos de qualidade estabelecidos. Cabe destacar que este processo não define os processos que aferem a qualidade do processo e do produto, mas sim os objetivos e as ações a serem tomadas quando os objetivos não forem alcançados. A determinação do grau de qualidade do produto e do processo é realizada por processos do grupo técnico.

O **processo de gerenciamento de propriedade intelectual** (*Intellectual property management process*) é responsável pela atribuição e gerência dos direitos autorais do produtos, licenças para distribuição de material de autoria da organização, o gerenciamento de licenças de uso de produtos e serviços de terceiros, depósito e utilização de patentes, resolução de conflitos de licenças e patentes. Este processo engloba o processo *Processo de copyright e licença* do SP-DEM, acrescentando como resultados esperados um plano para resolução de conflitos de propriedade intelectual, seja de violações da própria organização ou de outras organizações em relação aos produtos que detêm os direitos; o gerenciamento de depósitos e licenciamento de patentes (*e.g.*, patentes relacionadas aos padrões MPEG-4 utilizados em televisão digital); e o gerenciamento de marcas registradas.

5.2.1.3. Processos de projeto

Os processos de projeto estabelecem o gerenciamento de um projeto em detalhes, definindo o planejamento e o controle do projeto. Ele se organiza em sete processos: planejamento de projeto, avaliação e controle de projeto, gerenciamento de decisões, gerenciamento de risco, gerenciamento de configuração, gerenciamento de informação e medição (Figura 5.16).

O **processo de planejamento de projeto** (*Project planning process*) tem o propósito

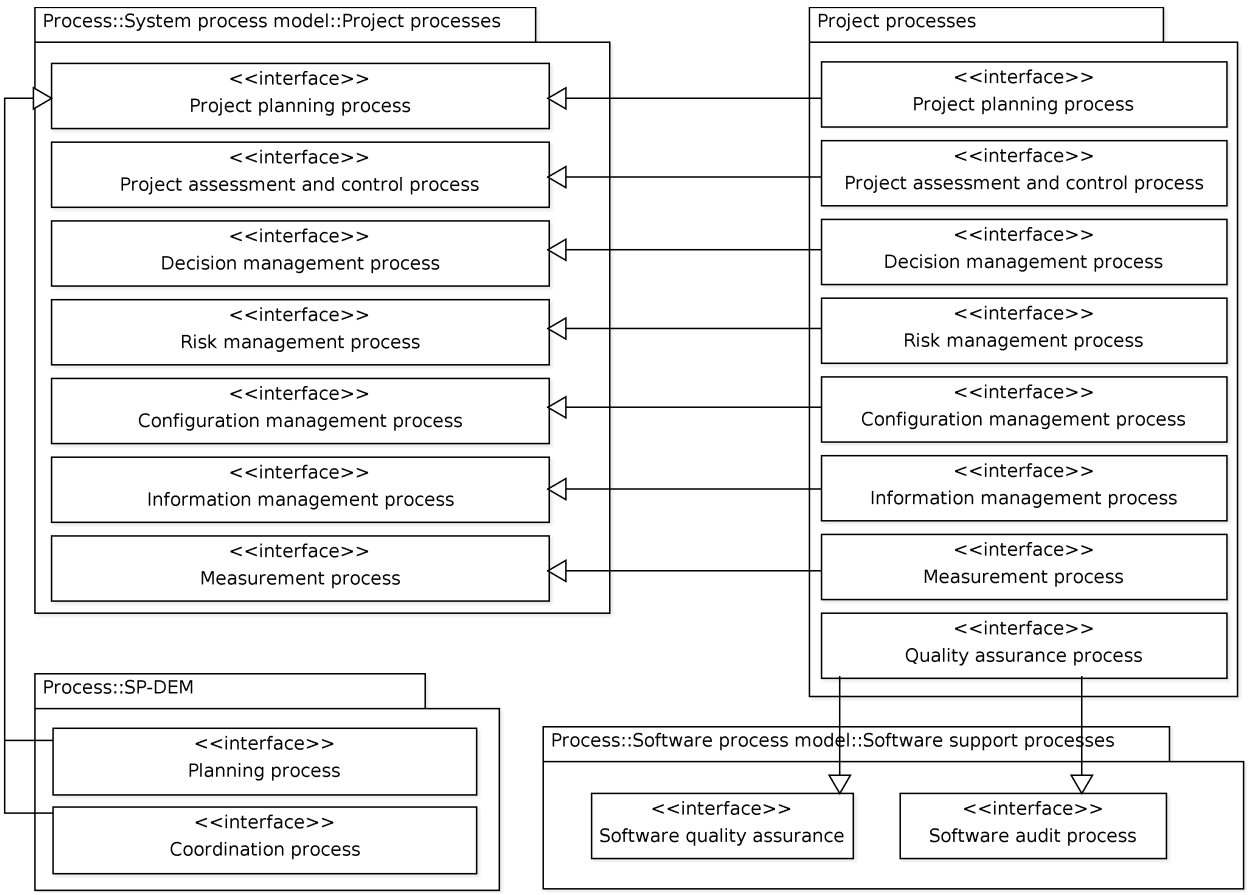


Figura 5.16: SPLOD: Grupo de processos de projeto e relação com os processos das normas ISO 15288:2008/12207:2008 e com o SP-DEM.

de definir o plano do projeto, o que inclui a delimitação de escopo, identificação de resultados esperados de cada processo, estabelecimento de cronograma para as atividades, definição dos recursos requeridos e produzidos pelas atividades, designação de papéis, estimativa de custos de cada atividade. A delimitação do escopo de um objeto de aprendizagem deve compreender a escolha das características educacionais da atividade de aprendizagem na qual ele será utilizado (formalidade, modalidade, acesso, avaliação e interação) e a definição dos objetivos de aprendizagem a serem atendidos e em que nível de profundidade cognitiva, afetiva e psicomotora. Sempre que possível, deve-se estabelecer a estrutura analítica do projeto (*work breakdown structure*), identificando os produtos e subprodutos esperados e, a partir desses, escolhendo um modelo de ciclo de vida (em conjunto com o *processo de gerenciamento de ciclo de vida*) e definindo os processos e atividades a serem realizados. Para cada atividade, deve-se designar pessoas responsáveis (alocando-as em conformidade com o *processo de gerenciamento de recursos humanos*), recursos (em coordenação com o processo de infraestrutura). Planos de qualidade para avaliar a execução do processo e os resultados gerados devem ser estabelecidos, com o auxílio do *processo de gerenciamento de qualidade*. Este processo substitui os processos de *planejamento* e de *coordenação* do SP-DEM (que, de fato, são englobados pelos processos de planejamento das normas ISO 15288:2008 e

12207:2008).

O **processo de avaliação e controle de projeto** (*Project assessment and control process*) mede a execução do processo e os produtos gerados pelo projeto, comparando-os periodicamente com o cronograma e estimativas estabelecidos no planejamento. Os resultados esperados são a medição do desempenho do projeto, a adequação dos papéis e recursos originalmente alocados, análise de desvios de planejamento, adoção de ações corretivas frente aos desvios, ajustes de cronograma e de planejamento, relato do progresso do projeto. As atividades exigidas por esse processo são: avaliação do projeto quanto ao orçamento e parâmetros de qualidade, medindo-se o progresso do projeto e a efetividade do time de desenvolvimento; avaliação do progresso e efetividade do projeto (a partir dos dados medidos na atividade anterior); controle do projeto, com a identificação, análise e resolução de problemas; e determinar a conclusão do projeto a partir da satisfação dos requisitos e outros critérios estabelecidos inicialmente definidos.

O **processo de gerenciamento de decisão** (*Decision management process*) estabelece a ação mais benéfica a um projeto (quando tal opção de ação existir) quando encontrado um impasse. Os resultados esperados são a definição de plano estratégico para gerenciamento de decisão, estudo de ações alternativas e a resolução (ação escolhida), acompanhada das razões e suposições que levaram a tal escolha. O plano deve identificar as circunstâncias e a necessidade de uma decisão. As decisões devem ser tomadas após analisar as propostas de solução e os resultados delas resultantes em relação com os critérios de sucesso estabelecidos estrategicamente. Durante toda a atividade, o raciocínio (*design rationale*) utilizado para excluir soluções e escolher uma alternativa em detrimento de outras deve ser registrado para posterior reúso em ocasiões posteriores.

O **processo de gerenciamento de risco** (*Risk management process*) identifica, analisa, trata e monitora continuamente os riscos durante o ciclo de vida do objeto de aprendizagem. Os resultados esperados são a determinação do escopo em que o gerenciamento de risco deve ser empregado, a definição de estratégias para gerenciamento de riscos, identificação, análise, priorização e resolução de riscos durante o ciclo de vida.

O **processo de gerenciamento de configuração** (*Configuration management process*) estabelece e mantém a integridade dos recursos criados ou modificados pelas atividades. Espera-se deste processo a definição de uma estratégia de gerenciamento de configuração, a definição dos itens a serem gerenciados, o estabelecimento de configurações de referência (*baseline*) e o controle das alterações realizadas sobre os itens sob gerência de configuração.

O **processo de gerenciamento de informação** (*Information management process*) coleta, armazena, recupera e dissemina informações precisas sobre os produtos e processos utilizados e gerados durante o projeto. Este processo deve identificar as informações a serem gerenciadas, a representação a ser adotada para as informações, a distribuição delas para os

interessados e a avaliação da completeza, correção e validade das informações do projeto.

O **processo de medição** (*Measurement process*) coleta, analisa e relata medidas relacionadas à execução dos processos e dos produtos gerados pela organização. Os dados a serem coletados são definidos de acordo com os objetivos técnicos e gerenciais do projeto e da organização. As medidas obtidas quanto aos produtos e processos são comparadas com indicadores de qualidade e os resultados são então divulgados, permitindo a ciência do estado dos processos e produtos pelos envolvidos no projeto.

O **processo de garantia de qualidade** (*Quality assurance process*) define uma estratégia para avaliar a qualidade do objeto de aprendizagem em desenvolvimento, coordenando os processos técnicos de *verificação*, *validação*, *inspeção* e *auditoria* (processos esses provenientes da ISO 12207:2008).

5.2.1.4. Processos técnicos

Os processos técnicos são responsáveis pela definição dos requisitos de um objeto de aprendizagem e a criação e manutenção de um produto que satisfaça tais requisitos. Doze processos técnicos são necessários para alcançar esse propósito, relacionados na Figura 5.17: definição de requisitos, análise de requisitos, projeto arquitetural, projeto detalhado, implementação, integração, verificação, transição, validação, operação, manutenção e retirada.

O **processo de definição de requisitos** (*Stakeholder requirements definition process*) obtém os requisitos dos interessados no objeto de aprendizagem. Para isso, as seguintes atividades são necessárias: identificação dos interessados; definição, análise e validação dos requisitos definidos ou obtidos dos interessados. A definição de requisitos resulta em uma especificação das necessidades e desejos dos usuários do objeto de aprendizagem e os atributos de qualidade a serem seguidos (notadamente usabilidade, dentre outros). No mínimo duas classes de interessados devem ser definidas e consideradas: aprendiz e professor. Ambas compartilham, como requisito, um subconjunto de objetivos educacionais: o professor extrai parte desses objetivos de currículos e de sua experiência enquanto o aprendiz de suas necessidades e expectativas. Um objeto de aprendizagem deve, idealmente, atender a todos esses objetivos educacionais. Além dos objetivos educacionais, deve-se estabelecer os requisitos quanto à instrução (*i.e.*, qual o método instrucional ou ao menos teoria pedagógica a ser utilizada), recursos a serem utilizados (computadores, televisão, etc), duração esperada para o uso do objeto de aprendizagem pelo aprendiz, tempo necessário pelo professor para preparar uma aula com o objeto, nível de personalização e interatividade.

Uma forma de representar os diversos requisitos acima definidos é pela descrição de casos de uso, ou seja, breves sequências de atividades a serem realizadas pelos interessados com o

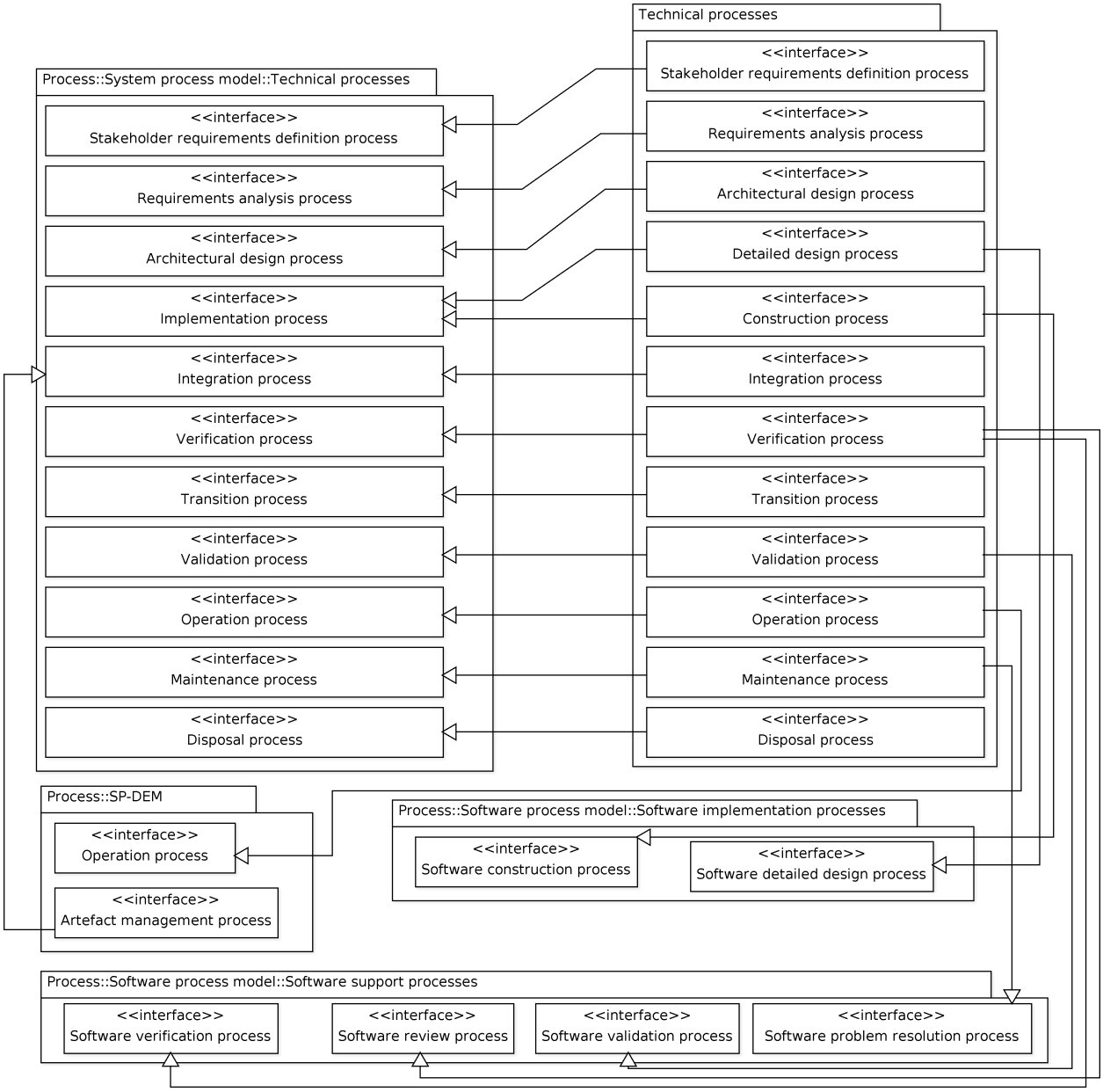


Figura 5.17: SPLOD: Grupo de processos técnicos e relação com os processos das normas ISO 15288:2008/12207:2008 e com o SP-DEM.

objeto de aprendizagem. Posteriormente, tais casos de uso podem ser utilizados para validar o sistema.

O **processo de análise de requisitos** (*Requirements analysis process*) detalha a especificação de requisitos dos interessados, analisando-a quanto à completeza, correção, inexistência de ambiguidade, rastreabilidade, consistência, verificabilidade e modificabilidade. Garantida a qualidade dos requisitos, deve-se priorizar os requisitos a serem implementados.

O **processo de projeto arquitetural** (*Architectural design process*) sintetiza um projeto que atenda aos requisitos do sistema, definindo um projeto arquitetural, os principais componentes e suas interfaces com outros componentes, objetos de aprendizagem e aplicativos

educacionais.

O **processo de projeto detalhado** (*Detailed design process*) define o projeto detalhado do objeto de aprendizagem, o que compreende o projeto instrucional e o projeto detalhado, de acordo com o paradigma de desenvolvimento de software escolhido, para os módulos de software do objeto de aprendizagem. Para este processo, deve-se utilizar métodos de modelagem, tais como os abordados na Seção 2.3.1 e na Seção 5.1 deste capítulo. Este processo é derivado do *processo de projeto detalhado de software* da norma ISO 12207.

O **processo de construção** (*Construction process*) implementa o projeto detalhado em uma linguagem de programação e gera objetos de aprendizagem executáveis (e possivelmente em pacotes padronizados conforme definido na Seção 2.2.3). Cada módulo do objeto de aprendizagem deve ser testado (teste de unidade), verificando-se a satisfação dos requisitos a ele alocados.

O **processo de integração** (*Integration process*) realiza a integração dos componentes do objeto de aprendizagem e do sistema de acordo com o projeto arquitetural definido. Para isso, é necessário o estabelecimento de uma estratégia de integração, atentando-se à verificação da conformidade do sistema com os requisitos, o teste do sistema integrado e o registro de erros (não-conformidades) encontrados. O *processo de controle de artefatos* do SP-DEM, que trata da integração dos componentes de um objeto de aprendizagem, é suplantado por este processo.

O **processo de verificação** (*Verification process* da norma ISO 15288:2008 ou *System qualification testing process* da norma ISO 12207:2008) tem o propósito de verificar a satisfação dos requisitos definidos na especificação do sistema e se os artefatos gerados estão condizentes com o preconizado pelos processos.

O **processo de transição** (*Transition process* da norma ISO 15288:2008 ou *Software installation process* e *Software acceptance support processes* da norma ISO 12207:2008)) estabelece a capacidade do objeto de aprendizagem funcionar corretamente no ambiente operacional. Neste ponto, devem ser considerados a instalação do objeto no sistema alvo, a configuração do ambiente para a execução do objeto, o correto funcionamento do objeto no ambiente operacional, o registro das configurações necessárias e das ações corretivas tomadas.

O **processo de validação** (*Validation process*) é responsável por verificar se o objeto de aprendizagem satisfaz os interessados, atendendo seus requisitos e expectativas, ou, caso não os satisfaça inteiramente, efetuar o registro dos erros observados para posteriormente corrigi-los.

O **processo de operacionalização** (*Operation process*) consiste no monitoramento e suporte prestado na utilização do objeto de aprendizagem e da aplicação educacional que o emprega, observando os requisitos definidos pelo usuário e corrigindo os erros detectados. Este processo também engloba o *processo de disponibilização* do SP-DEM, que trata da

disponibilização e acompanhamento do uso do objeto de aprendizagem, inclusive com suporte instrucional e identificação de problemas e melhorias.

O **processo de manutenção** (*Maintenance process*) garante a capacidade do objeto de aprendizagem em atender os requisitos e expectativas do usuário, realizando a análise de erros e a execução de manutenção corretiva, adaptativa, perfectiva e preventiva.

O último processo técnico é o **processo de retirada/descarte** (*Disposal process*). Ele determina o final do ciclo de vida do objeto de aprendizagem. Deve-se notificar os interessados da aposentadoria do objeto e o fim de seu processo de desenvolvimento. Todos os processos relacionados ao projeto devem ser finalizados e todos os artefatos produzidos pelo projeto devem ser disponibilizados aos interessados e usuários do objeto de aprendizagem.

5.2.2. LODP: atores

Os atores correspondem aos interessados no objeto de aprendizagem de acordo com as três perspectivas estudadas nesta tese: educacional, computacional e de interação com televisão digital. Na perspectiva educacional, os usuários são os aprendizes, professores e projetistas instrucionais. Deve-se considerar que, na prática, os professores também são os projetistas instrucionais.

A perspectiva computacional reúne os atores responsáveis pelo desenvolvimento dos modelos computacionais a partir dos requisitos e modelos instrucionais de responsabilidade da perspectiva educacional. Dada a semelhança entre objetos de aprendizagem e software quanto às técnicas de desenvolvimento, os papéis também são semelhantes: engenheiro de requisito, analista/projetista, programador e testador. Esses papéis podem ser desempenhados pela mesma pessoa, mas é recomendável que o engenheiro de requisitos tenha mais familiaridade com a parte educacional (embora com pleno domínio das técnicas de análise e modelagem computacional).

As interações a serem desenvolvidas estão atreladas às atividades de aprendizagem definidas no projeto instrucional e vice-versa: cabe ao ator responsável pelo projeto das interações estimular os atores da perspectiva educacional a utilizarem métodos instrucionais diferenciados, explorando as diferentes formas de interação (tais como aquelas apresentadas no Capítulo 3). Além da questão de interação e de estética, deve-se tratar também da usabilidade. Eventualmente, os mesmos atores podem desempenhar ambos os papéis.

5.2.3. LODP: processos

O desenvolvimento de objetos de aprendizagem multimídia e interativos deve ocorrer com a participação dos interessados, em especial dos professores e aprendizes. Embora seja possível o desenvolvimento de objetos de aprendizagem em um contexto desvinculado, em que uma

organização estabelece-se como fornecedora e outra instituição de ensino define-se como consumidora, o contexto para o qual este processo foi projetado prevê a interação próxima e aberta entre os interessados, constituindo uma comunidade de desenvolvimento de objetos educacionais abertos.

O processo definido para o desenvolvimento desse tipo de objetos de aprendizagem organiza-se em quatro grupos, conforme exposto na Figura 5.18: processos de habilitação da comunidade, processos de estabelecimento da comunidade de desenvolvimento, processos de projeto e processos técnicos.

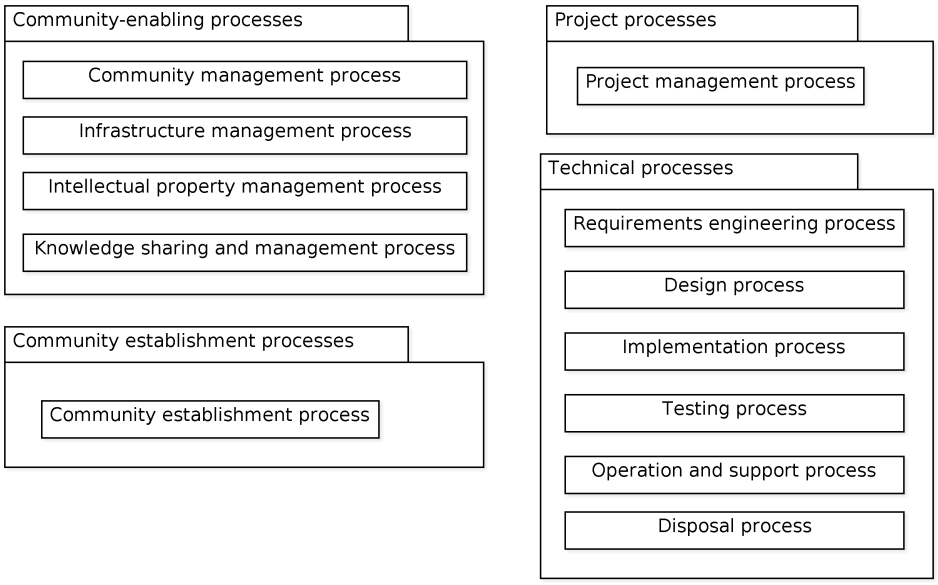


Figura 5.18: LODP: Visão geral.

5.2.3.1. Processo de estabelecimento da comunidade

Um passo anterior ao processo de desenvolvimento de um objeto de aprendizagem corresponde ao estabelecimento de um processo de ensino e aprendizagem no qual os objetos de aprendizagem serão utilizados. O entendimento é que os objetos desenvolvidos (criados ou reutilizados com ou sem modificações) são utilizados em ambientes de aprendizagem ou algum outro software educacional para a execução de uma atividade de aprendizagem (por atividade de aprendizagem, entenda-se um curso, as subatividades "disciplinas" do curso, as "aulas" de uma disciplina e assim por diante).

As oportunidades para realizar a atividade educacional possuem várias fontes: o currículo de um curso, as necessidades de uma comunidade local (*life-long learning*), as necessidades de uma empresa para o desenvolvimento de um produto inovador (*treinamento*) ou até mesmo uma curiosidade. Em resumo, as oportunidades podem atender às demandas explícitas ou serem oportunamente identificadas e sugeridas.

Identificadas as oportunidades e as pessoas interessadas tanto em ensinar quanto a aprender, estabelece-se um acordo, formal ou informal, para o preparo das atividades e objetos de aprendizagem necessários e a execução do processo educacional. O comprometimento entre as partes engloba a definição das metas a serem alcançadas, os prazos para concluir uma versão dos objetos, a periodicidade com que o desenvolvimento ocorrerá e os termos gerais de compartilhamento de informações. Esse comprometimento é selado com a publicação do acordo, materializado pela criação de um local público para a divulgação do projeto, reconhecimento de seus integrantes e a entrega pública dos objetos de aprendizagem (produto final) e artefatos gerados pelos demais processos e atividades.

O suporte é realizado desde o estabelecimento do projeto que desenvolve o objeto de aprendizagem, com a divulgação de artefatos e versões preliminares do produto. Canais de comunicação – fóruns, listas de discussão, espaços para conversa em tempo real por texto ou voz – são estabelecidos, definindo-se um meio de comunicação entre as partes envolvidas no desenvolvimento e uso dos objetos.

As atividades expostas já ocorrem, rotineiramente, em alguns contextos. Por exemplo, um professor universitário, ao assumir uma disciplina (identificação de oportunidade), estabelece acordos com os professores anteriores e do departamento para compartilhar ideias e material. Antes do período letivo, em conjunto com assistentes de docência (monitores e estagiários), refina a proposta didática e divulga publicamente o plano de curso, estabelecendo um acordo com os alunos. A execução de atividades didáticas e utilização de objetos de aprendizagem podem ou não ser publicadas, servindo ao propósito de avaliação, realizada ao final da disciplina. Os resultados das avaliações e as sugestões e críticas dos alunos são então utilizados para uma nova iteração do processo.

Um ponto interessante quanto ao desenvolvimento de objetos de aprendizagem é que ele raramente é definitivamente concluído: a cada nova iteração de um processo educacional, um novo contexto se estabelece e o objeto precisa ser alterado. Tal como um software livre, um objeto de aprendizagem é continuamente aperfeiçoado. Mantendo a linha de raciocínio de software livre, um objeto de aprendizagem pode ter um acordo de duração ilimitada e sem requisitos plenamente estabelecidos: enquanto a comunidade tiver disposição, os objetos de aprendizagem (tal como no caso de software) continuam em desenvolvimento. No caso de objetos de aprendizagem abertos/livres (*open educational resources*), nem mesmo a desistência de uma das partes quanto ao acordado significa o fim do processo: sempre é possível continuar o desenvolvimento. Assim, o fim do acordo ocorre unilateralmente, com a parte restante mantendo os demais acordos e realizando novos, mantendo a comunidade ao redor do objeto de aprendizagem ativa.

O **processo de estabelecimento de comunidade** (*Community establishment process*) compreende todas as atividades relacionadas aos processos de aquisição e fornecimento de

objetos de aprendizagem definidos no grupo de processos de acordo do processo padrão, conforme apresentado na Figura 5.19.

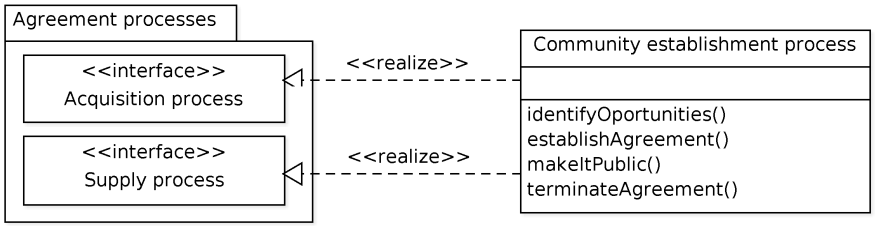


Figura 5.19: LODP: Processo para estabelecimento de comunidade e relação com os processos do SPLOD.

O processo de estabelecimento da comunidade também compreende algumas atividades de formação de recursos humanos. Novos membros podem ser convidados para a comunidade de acordo com as necessidades dos projetos. Além disso, existe o processo de compartilhamento de conhecimento pelo qual os membros documentam técnicas e processos. A aquisição de novas habilidades pelos membros existentes realiza-se pela comunicação com os demais membros e com o material compartilhado pela comunidade.

5.2.3.2. Processos de habilitação e manutenção da comunidade

Uma vez criada a comunidade, é necessário mantê-la em funcionamento. O grupo de processos para habilitação/manutenção da comunidade, mostrados na Figura 5.20, atende a esse propósito. Uma característica de desenvolvimento de software livre, aplicável ao desenvolvimento de objetos de aprendizagem, é a liberdade que os membros da comunidade têm em sua atuação de acordo com as políticas do projeto. As políticas – regras de conduta e responsabilidade – são inicialmente definidas pelo iniciador da comunidade e posteriormente revisadas e ampliadas pelos demais membros, conforme a necessidade dos projetos executados. A atribuição de responsabilidades deve acontecer por mérito, o qual se obtém pelos resultados gerados durante a execução de atividades relacionadas aos processos. Os responsáveis pela gerência da comunidade e dos processos devem definir, com o apoio dos demais membros, os processos de ciclo de vida a serem adotados e procedimentos específicos do projeto. Mudanças nessas definições podem e devem ser realizadas periodicamente, após o alcance de marcos (*milestones*) dos projetos (em geral, disponibilização de uma versão de um objeto de aprendizagem), com base na evolução dos artefatos produzidos pelos processos e indicadores de qualidade.

Uma comunidade de desenvolvimento de objetos de aprendizagem nasce com um propósito – um projeto – em específico. Dentro desse projeto, outros se desdobram, conforme definido no roteiro (*roadmap*). O conjunto de projetos e subprojetos define o portfólio de projetos da comunidade. Cada subprojeto seleciona e refina requisitos do projeto do qual ele faz

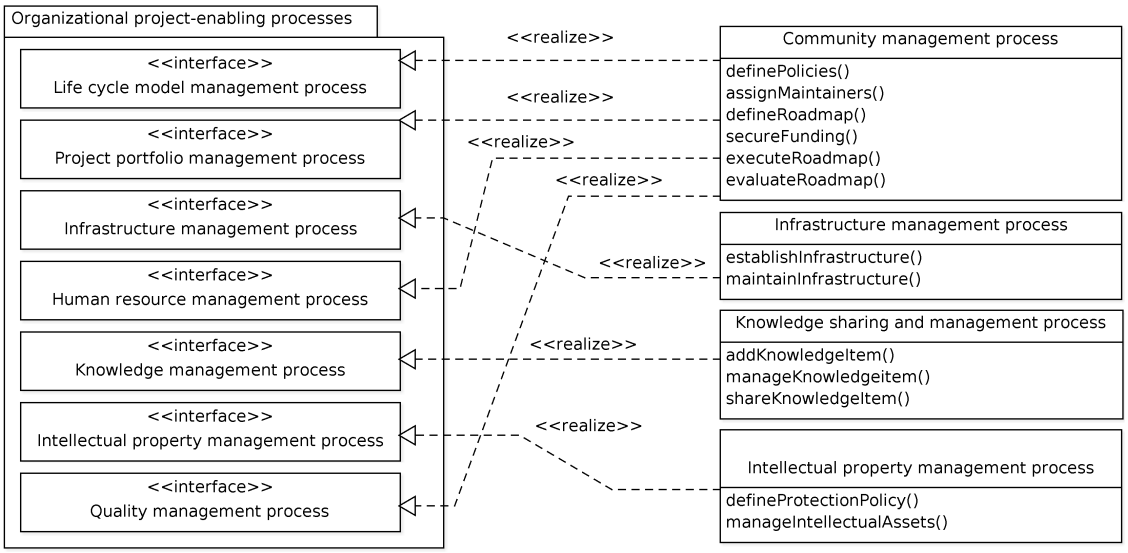


Figura 5.20: LODP: Processos para manutenção de comunidade e relação com os processos do SPLOD.

parte e define resultados esperados que contribuam para o objetivo global da comunidade. Cada projeto possui um mantenedor, o qual gerencia a execução dos processos e atividades relacionados, relatando os resultados para o mantenedor do projeto-pai, que por sua vez monitora os projetos-filho e intervém quando requisitado ou quando detectado um problema relevante.

Um elemento essencial para a manutenção de uma comunidade é o fornecimento de infraestrutura para execução dos projetos. Em comunidades pouco maduras, os próprios membros, em especial os mantenedores, proveem os recursos necessários para a execução dos processos de maior exposição (processos organizacionais e de apoio) e relegam a infraestrutura para os processos técnicos às pessoas que vão executá-los. Serviços públicos são geralmente utilizados para esse fim, em especial repositórios (*forges*) e serviços flexíveis provido por terceiros (*cloud computing*). A escolha dos serviços e dispositivos (*hardware*) necessários deve ser realizada por consulta aos mantenedores e demais membros, e deve estar associada às atividades específicas de algum dos processos aqui definidos. No entanto, existem recursos comuns a qualquer projeto de desenvolvimento de objeto de aprendizagem e que devem ser disponibilizados em um primeiro momento: serviços para controle de versão, compartilhamento de texto e arquivos (tal como uma *wiki*), canais de comunicação (fóruns de discussão e listas de *email*) e um ambiente para teste dos objetos gerados, preferencialmente com acesso público e implantação periódica dos objetos de aprendizagem produzidos (seja em sua versão estável ou de desenvolvimento). No caso de objetos para a televisão digital, na impossibilidade de permitir o acesso facilitado e ágil em dispositivos reais, deve-se fornecer uma máquina virtual que permita a execução dos objetos ou, ao menos, vídeos demonstrativos (nesse caso sem possibilidade de interação).

O **processo de compartilhamento de conhecimento** estabelece as políticas e mecanismos para o compartilhamento de informações entre os membros da comunidade e com qualquer outro interessado. As informações compartilhadas compreendem os artefatos gerados pelos processos, informações sobre técnicas e procedimentos utilizados, documentação do processo, tutoriais e qualquer outro tipo de material que agregue conhecimento ao projeto e possa ser útil a algum de seus membros.

Concomitante ao compartilhamento de conhecimento e aos outros processos, existe a atuação do **processo de proteção intelectual**. Ele se aplica a todo conhecimento e artefatos criados para ou utilizados pelo projeto. A política de proteção intelectual deve definir as licenças de uso que serão adotadas para os artefatos produzidos, quais as licenças compatíveis com as licenças adotadas, o registro e proteção da marca comercial (se este for o caso), a consulta ao banco de patentes e o licenciamento daquelas pertinentes ao projeto. Em objetos de aprendizagem, de forma a encorajar o reúso, em geral evitam-se patentes e se adotam licenças livres, tais como as Creative Commons[1] ou similares, para os artefatos gerados. No caso de televisão digital, em geral não constituem problemas para os desenvolvedores as patentes e marcas registradas relacionadas com os aplicativos interativos (*e.g.*, patente dos algoritmos utilizados no MPEG-4 AVC ou direito de uso da marca Java caso utilizado o subsistema Ginga-J), mas é dever alertar os usuários dos objetos de aprendizagem dessas questões e facilitar o contato para o devido licenciamento e pagamento (se realmente necessário) dos itens protegidos intelectualmente. Também faz parte do processo de proteção intelectual assegurar que terceiros não estão violando os direitos da comunidade, utilizando objetos de aprendizagem em desacordo com as licenças e acordos estabelecidos.

5.2.3.3. Processos de projeto

Conforme afirmado anteriormente, a própria comunidade é um projeto para o desenvolvimento de um conjunto de objetos de aprendizagem. Por questões lógicas, divide-se o projeto em subprojetos, cada qual administrado independentemente, mas sob a coordenação do *processo de gerenciamento da comunidade* (Figura 5.21).

A criação de um projeto ocorre por iniciativa da comunidade, que instancia o **processo de gerenciamento de projeto** (*Project management process*). Esse processo é uma versão em escala reduzida daquele de gerenciamento da comunidade, com um escopo bem definido. Tal como o *processo de planejamento* (do modelo de referência), cabe a ele a execução de uma gama de atividades e tarefas. A primeira é a definição do escopo do projeto, com a definição das características, dos requisitos (objetivos de aprendizagem) e dos usuários do objeto de aprendizagem (professores e aprendizes). De acordo o escopo, define-se o modelo de processo, de ciclo de vida e planos de qualidade a serem adotados, de acordo

[1] http://creativecommons.org/

com aqueles estabelecidos no projeto-pai ou outros subprojetos. O objeto de aprendizagem é decomposto em partes (projeto instrucional, mídias, software) e estimam-se os recursos necessários para a execução das atividades para cada parte identificada. Atribui-se, para cada processo e respectivas atividades, os indivíduos responsáveis para cada papel e os recursos – infraestrutura, financeiros, humanos – em coordenação com o *processo de gerenciamento de comunidade*. Entre os papéis, certamente existirão os de projetista instrucional, instrutor (professor), desenvolvedor de software, projetista (*designer*) gráfico e aprendizes, dentre outros que devem fazer parte da comunidade ou, caso contrário, contratados ou convidados a participarem do projeto.

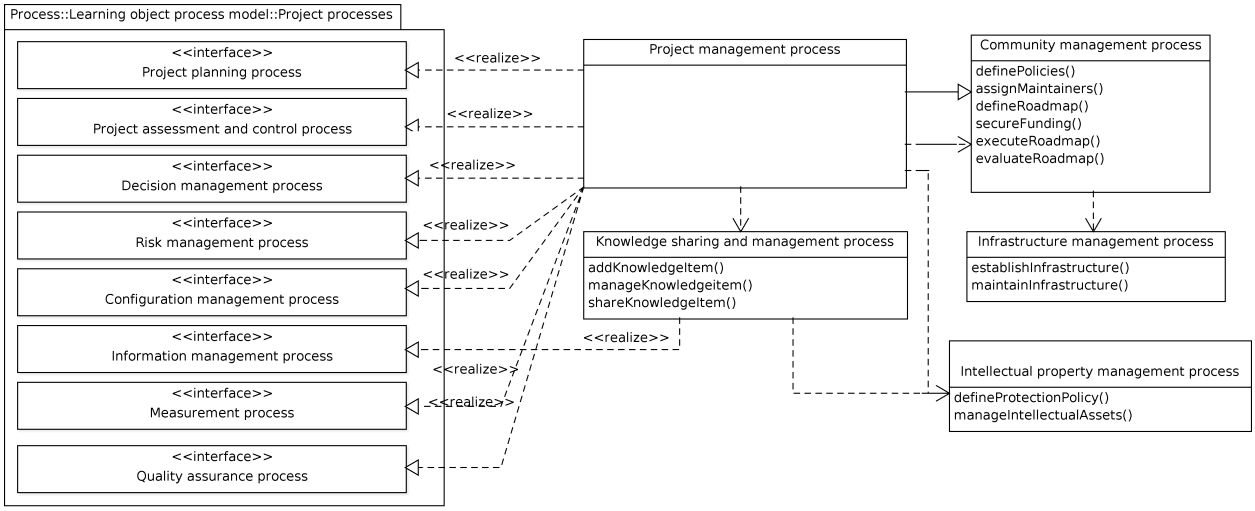


Figura 5.21: LODP: Processos de projeto e relação com os processos do SPLOD.

O *processo de gerenciamento de projeto* também é responsável pelo controle e avaliação do projeto, conforme definido no *processo de controle e avaliação de projeto* do processo padrão. O monitoramento é realizado a partir das informações compartilhadas do projeto e extraídas de acordo com métricas requeridas no plano de qualidade. Na ausência delas, aciona-se o *processo de compartilhamento de informação* para que os dados necessários sejam coletados. Na presença delas, estabelecem-se as medidas conforme o plano de qualidade e avaliam-nas quanto ao progresso do projeto comparando com o estabelecido no planejamento e, se necessário, tomando medidas para corrigir os erros detectados.

Um objeto de aprendizagem possui requisitos estringentes quanto ao cronograma: as atividades de aprendizagem usualmente têm uma data inicial bem definida. Ao mesmo tempo, um objeto é reutilizado em diversas oportunidades: em educação a distância, por exemplo, apesar da data inicial de disponibilização da atividade ser fixa, a execução pode ser em um momento muito posterior, o que permite a atualização periódica do objeto de aprendizagem (no caso da televisão digital, pela retransmissão de uma nova versão da aplicação interativa). Assim, apesar dos desvios de execução do cronograma, é possível entregar o objeto, embora com uma qualidade aquém da desejável. Quando se estabelece que

o produto está pronto, deve-se guardar todos os modelos necessários para o desenvolvimento do objeto de aprendizagem, atualizando-se o gerenciador de conhecimento organizacional com as lições aprendidas durante o projeto e, se necessário, corrigindo os modelos de processo e de ciclo de vida definidos pela organização.

Os impasses detectados no processo, sejam eles apresentados espontaneamente ou detectados nas avaliações de progresso, são resolvidos pelo responsável pelo projeto, em cooperação com os membros da comunidade que estão envolvidos com o elemento que causou o impasse. Sempre que possível, tais impasses devem ser discutidos publicamente e, preferencialmente, com o auxílio de um sistema de gerenciamento de erros (parte do gerenciamento de configuração).

O planejamento do projeto e as decisões tomadas em situações de impasse incorrem em riscos ao projeto. Em geral, o processo de gerenciamento de projeto trata dos riscos pela transparência dos processos e produtos, sujeitos ao escrutínio dos demais membros. Dessa forma, o foco do tratamento de riscos, previsto no *processo de gerenciamento de riscos* do processo padrão, deve ser nas interfaces dos produtos do projeto com os demais produtos desenvolvidos pela comunidade (conforme definido na estrutura analítica do projeto definida durante o planejamento do projeto). Os riscos devem ser analisados e tratados de forma análoga aos impasses, publicamente e sob controle de configuração. Caso não seja possível reunir elementos para aferir adequadamente o risco, deve-se criar subprojetos que implementem as medidas sugeridas e, com base nos resultados, decidir a melhor solução para gerenciá-lo.

O gerenciamento de configuração é realizado no contexto dos demais projetos desenvolvidos pela comunidade, facilitando o compartilhamento de artefatos entre projetos distintos. Mecanismos de controle de versão e gerenciamento de erros são combinados para atender às atividades do *processo de gerenciamento de configuração* do processo padrão.

A garantia da qualidade do projeto é estabelecida pela execução de processos técnicos de verificação, validação e teste de acordo com o plano estabelecido pelo *processo de gerenciamento de projeto*.

5.2.3.4. Processos técnicos

O grupo de processos técnicos (Figura 5.22) organiza-se em seis processos: engenharia de requisitos, projeto, implementação, teste, operação/suporte e retirada.

O **processo de engenharia de requisitos** (*Requirements Engineering Process*) compreende os processos de *análise de requisitos dos interessados* e *análise de requisitos do sistema* do processo-padrão. Ele consiste na identificação de pessoas que representem os principais interessados no objeto de aprendizagem (aprendizes e instrutores) e quaisquer outros que sejam julgados relevantes para o objeto (*e.g.*, projetistas gráficos, engenheiros de usabilidade).

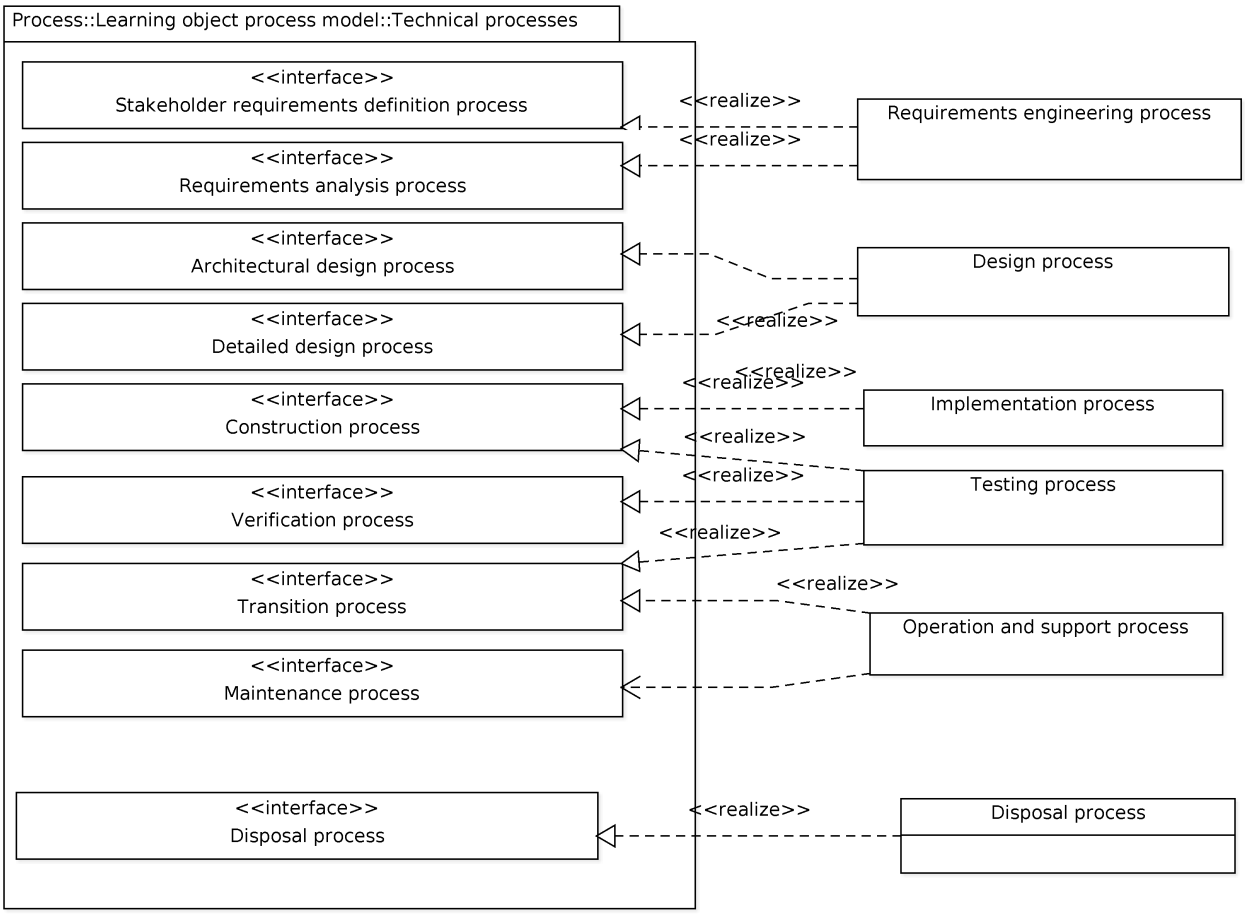


Figura 5.22: LODP: Processos técnicos e relação com os processos do SPLOD.

Ressalta-se que, para cada tipo de interessado identificado, deve existir ao menos um membro do projeto que o represente. Caso isso não se verifique, é mandatório que seja procurado um novo membro que satisfaça essa condição.

A partir da identificação dos interessados, elicitam-se os seus requisitos. Caso o objeto de aprendizagem seja integrante de outra atividade de aprendizagem (*e.g.*, um curso), os objetivos educacionais daquela atividade devem ser considerados como o conjunto inicial de requisitos. Os interessados devem descrever os seus requisitos, necessidades e expectativas quanto ao objeto de aprendizagem, tal como exposto no processo-padrão. Todo requisito deve ser registrado e documentado como um item de conhecimento (*processo de gerenciamento de conhecimento*). Os membros do projeto devem, então, definir os requisitos prioritários (mas sem necessariamente definir a prioridade de todos os requisitos), os quais serão considerados inicialmente para implementação. Todo requisito escolhido para implementação deve ser mantido sob controle de configuração. Após definido como item de configuração, o requisito deve ser analisado quanto à sua qualidade (completeza, correção, dentre outros atributos estabelecidos no processo padrão), com o controle de toda alteração e supervisão do membro responsável pela engenharia de requisitos e dos interessados que o definiram. Casos de uso devem ser descritos com cada requisito (se ele não for um caso de uso), capturando cenários

de uso do objeto de aprendizagem que exercitem o requisito.

O **processo de design** é responsável pelo projeto arquitetural e detalhado do objeto de aprendizagem, logo implementando os respectivos processos do SPLOD. O desenho/projeto do objeto de aprendizagem inicia-se com a definição da arquitetura do objeto de aprendizagem e as interfaces dele com outros objetos e com o sistema. Dentre os requisitos definidos na especificação, devem constar os aplicativos educacionais (*e.g.*, ambientes virtuais de aprendizagem) em que o objeto será utilizado ou, ao menos, os padrões de representação de objetos de aprendizagem a serem adotados e, na inexistência ou insuficiência desses padrões, de considerações quanto ao uso deles em plataformas específicas. O projeto arquitetural deve apresentar uma visão global do sistema e que permita a satisfação de todos os requisitos escolhidos para implementação.

Realizado o projeto arquitetural, inicia-se o projeto detalhado do objeto de aprendizagem. Esse projeto deve especificar tanto o projeto instrucional do objeto de aprendizagem quanto o projeto dos módulos de software necessários à sua construção. O projeto instrucional deve ser modelado com alguma técnica específica, tal como as apresentadas no Seção 2.3 ou, preferencialmente, a técnica LODM definida neste capitulo. Os modelos dos módulos de software devem ser definidos de acordo com o paradigma escolhido para a construção (procedural, orientado a objeto ou declarativo).

A implementação do projeto detalhado é realizado pelo **processo de implementação**, correspondente às atividades do *processo de construção* do SPLOD. Esse processo é dedicado, principalmente, à implementação dos módulos de software produzidos de acordo com os modelos instrucionais e modelos detalhados do *processo de design*.

O teste do objeto de aprendizagem deve ser realizado em três fases: unidade, integração e de sistema. Todas elas são definidas dentro do escopo do **processo de teste** (*Testing process*). O teste de unidade corresponde à atividade de teste unitário do *processo de construção* do SPLOD. O teste de integração deve certificar que os módulos do objeto de aprendizagem funcionam corretamente quando integrados. O teste de sistema deve verificar se os objetos de aprendizagem funcionam corretamente com outros objetos de aprendizagem, com o software educacional em que ele será instalado e com o restante do sistema em que ele será operado. Critérios devem ser estabelecidos com base na especificação de requisitos e na própria estrutura do objeto de aprendizagem, derivando-se requisitos de teste que devem ser satisfeitos antes do objeto de aprendizagem ser validado pelos interessados. A execução dos testes de integração satisfaz o *processo de verificação* e parte do *processo de operação* (atividades relacionadas ao teste de sistema) do SPLOD.

As demais atividades do *processo de operação* (SPLOD) são realizadas pelo **processo de operação e suporte** (*Operation and support process*), que também satisfaz o *processo de operação* da ISO 15288:2008. Definem-se as instruções para a instalação do objeto de

aprendizagem e a disponibilização de pacotes de instalação deles. Os meios de comunicação disponibilizados para o projeto pelo *processo de compartilhamento de conhecimento* são utilizados para prestar suporte aos usuários.

O suporte também é responsável pela identificação de novos requisitos ou alteração dos atuais para o *processo de engenharia de requisitos*, de forma que próximas versões do objeto de aprendizagem atendam às novas demandas dos usuários (manutenção corretiva, adaptativa e melhorias). Os próprios membros do projeto, pela execução de atividades de garantia de qualidade, podem definir novos requisitos ou detalhar os existentes, realizando manutenção perfectiva. Independente do tipo de manutenção, todas as alterações estão sujeitas ao controle de configuração (parte do *processo de gerenciamento de projeto*), documentando-as extensivamente e definindo casos de teste para evitar regressões. Essas atividades de manutenção são definidas no *processo de operação e suporte* e repassadas para o gerenciamento do projeto (e desse para o processo de engenharia de requisitos, projeto, implementação ou teste, de acordo com o tipo de manutenção a ser realizada).

Ao final do ciclo de vida de um objeto de aprendizagem, o **processo de retirada** deve ser executado, notificando-se os membros do projeto, interessados e usuários do objeto quanto à aposentadoria dele e disponibilizar, pelo máximo de tempo possível, os artefatos gerados pelo projeto. Todos os processos do projeto devem ser encerrados, mantendo-se apenas o necessário da infraestrutura para permitir o acesso (restringindo-se pela questão custo e pelo tempo de vida da comunidade que originou o projeto).

5.2.4. LODP: ciclo de vida

O ciclo de vida de desenvolvimento de objetos de aprendizagem configura-se em estágios/fases semelhantes a de um software: concepção, análise, desenvolvimento e implantação. Durante cada estágio, todos os processos técnicos são executados sob a coordenação do processo de gerenciamento de projeto. A execução dos processos técnicos faz uso constante dos processos de apoio à comunidade de desenvolvimento (principalmente quanto ao compartilhamento de conhecimento). Observando-se os processos definidos no LODP, o ciclo de desenvolvimento configura-se como apresentado na Figura 5.23.

Os processos são executados de forma iterativa, conforme observado no quadrante inferior esquerdo da Figura 5.23, em atendimento aos requisitos de processo definidos no Capítulo 4. Na fase inicial, concepção, a ênfase será no processo de engenharia de requisitos, mas todos os demais processos também serão executados, obtendo-se ao final um objeto de aprendizagem, ainda que na forma de protótipo, mas que permita a avaliação pelos interessados. A partir do momento em que a quantidade dos requisitos não se alterar significativamente, os processos de projeto são os mais exigidos, detalhando-se os requisitos (em especial para atender aos atributos de qualidade) e definindo-se projetos instrucional, arquitetural e detalhado estáveis

para o objeto de aprendizagem. Uma vez definidos os projetos, o esforço desloca-se para os processos de construção e teste. Conforme mais requisitos são implementados, o objeto de aprendizagem pode ser efetivamente utilizado em atividades de aprendizagem, exigindo as atividades do processo de operacionalização e suporte.

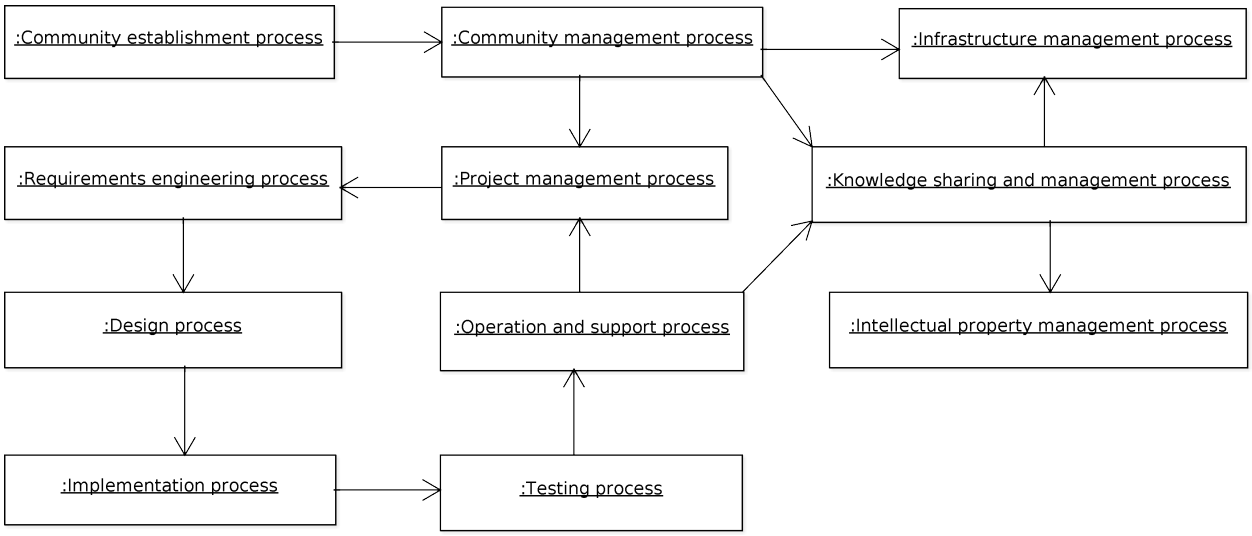


Figura 5.23: LODP: Ciclo de vida de um objeto de aprendizagem.

Além do ciclo de vida de cada objeto de aprendizagem, existe o ciclo de vida da comunidade como um todo. Cada iteração de cada projeto contribui para o estabelecimento da infraestrutura e compartilhamento de conhecimento da comunidade, o que, por sua vez, contribui indiretamente para o sucesso de todos os projetos executados sob sua alçada.

Embora não apresentado como parte do ciclo de vida de um objeto de aprendizagem, deve-se associá-lo com o ciclo do processo instrucional. Considerando o modelo ADDIE como referência, os principais requisitos de um objeto de aprendizagem são provenientes do processo de análise do ADDIE. O projeto instrucional, com a definição das atividades de aprendizagem e a sequência de execução delas, é resultado do processo de projeto do ADDIE. Já o processo de desenvolvimento corresponde ao ciclo de vida do desenvolvimento de um objeto de aprendizagem até o alcance do estágio de implantação. O objeto resultante é utilizado pelo professor e aprendizes na condução da implementação do ADDIE. Os resultados das atividades de aprendizagem e do processo de avaliação do ADDIE servem como novos requisitos para a fase de análise e, consequentemente, para novas iterações de desenvolvimento do objeto de aprendizagem.

5.3. LODE: Learning Object Development Environment

A escolha por uma abordagem baseada em modelos possibilita a automatização de uma parcela significativa das atividades e os métodos associados a sua execução. No caso específico do LODM, a geração de modelos conceituais a partir de materiais didáticos, as transformações de modelo conceitual para instrucional e de instrucional para interação e, finalmente, a geração de objetos de aprendizagem a partir do modelo de interação e da escolha de uma plataforma de execução são as atividades que oferecem um maior ganho com a automatização.

O ambiente LODE implementa, total ou parcialmente, as regras e procedimentos para geração e transformação a partir de modelos definidos em linguagens específicas de domínio passíveis de edição em ferramentas comuns de desenvolvimento. Por exemplo, para o modelo conceitual, adotam-se documentos CXL, armazenados localmente ou em servidores de mapas conceituais. Quanto aos modelos instrucionais e de interação, eles são representados por documentos *XML Metadata Interchange* (XMI) que definem modelos especificados de acordo com a UML, dispondo de diversas ferramentas compatíveis.

Na atual versão de desenvolvimento do ambiente, o objetivo é permitir o uso de ferramentas maduras de edição de modelos e que suportem padrões abertos. Assim, no âmbito deste trabalho foram utilizados o CmapTools (CANAS *et al.*, 2004) para a autoria de mapas conceituais e o CmapServer (CANAS *et al.*, 2006) para o armazenamento, indexação e recuperação dos mapas; os modelos UML foram editados com o ArgoUML (ROBBINS, 1999) e o DoUML [2], ambos software livre, e o UModel (Altova, 2005).

As transformações foram definidas em Java, utilizando bibliotecas para mineração de texto e definição de metamodelos e modelos. A ferramenta é organizada conforme apresentado no diagrama da Figura 5.24: o pacote que define os modelos (`Model`) utilizados e definidos no método LODM; o pacote responsável pela extração de dados e processamento de linguagem natural (`Miner`), utilizado principalmente na geração do modelo conceitual; e o pacote que define a geração e a transformação de modelos (`Transformer`).

O pacote de modelos (`Model`) é independente de qualquer outro pacote ou classe do LODE e possui poucas dependências para bibliotecas externas. Seus principais elementos são o modelo conceitual, o *statechart* e os modelos definidos pelo LODM para a representação dos modelos instrucional e de interação, e os modelos associados aos formatos em que são gerados os objetos de aprendizagem.

O pacote de mineração de dados (`Miner`) é organizado em quatro pacotes, conforme configurado na Figura 5.25a. O primeiro, `Acquisition`, trata da obtenção dos dados a serem processados. No caso específico do LODE, suas principais funções são a recuperação de

[2] Disponível em `https://github.com/leonardo2d/douml`.

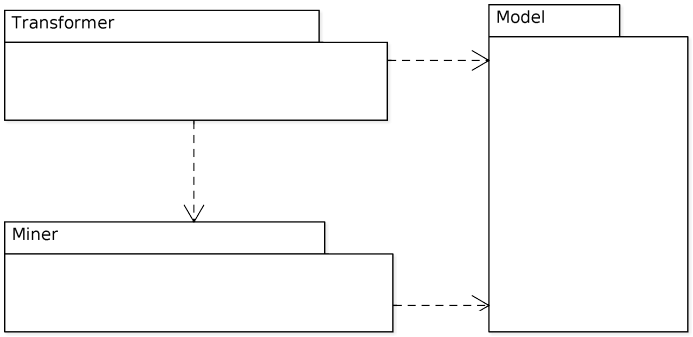
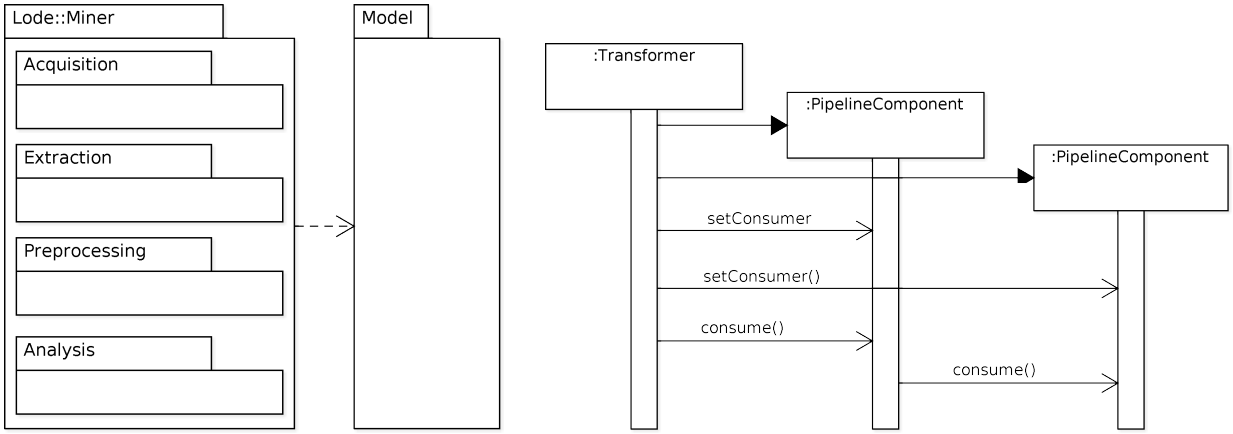


Figura 5.24: LODE: Diagrama dos principais pacotes.

arquivos contendo material didático (documentos PDF) e mapas conceituais de arquivos ou servidores compatíveis com os serviços Web definidos por Canas *et al.* (2006). O pacote de extração (`Extraction`) analisa os artefatos recuperados e extrai elementos, conforme os modelos do pacote `Model`. Por exemplo, de um documento PDF, extraem-se apenas textos; de servidores de mapas conceituais, obtém-se os conceitos e as proposições especificados no mapa.

Dependendo dos requisitos da transformação, torna-se necessário processar os elementos extraídos, eliminando dados pouco úteis e transformando o restante em elementos mais complexos e úteis para a construção dos modelos do LODM. A responsabilidade de tratamento dos elementos cabe às classes do pacote de pré-processamento (`Preprocessing`). Especificamente para a geração de modelos conceituais, são utilizadas classes para remoção de símbolos indesejados, remoção de palavras comuns (*stopwords*), extração da raiz das palavras (*stemming*), dentre outras. Com os elementos tratados, procede-se à análise (pacote `Analysis`), para a priorização dos termos e proposições a serem considerados para o modelo conceitual.



(a) LODE: Elementos do pacote responsável pela mineração de dados.

(b) LODE: Interação entre as principais classes do pacote de mineração de dados e de transformação de modelos.

Figura 5.25: LODE: Pacotes de mineração de dados e de transformação.

As classes do pacote de mineração foram desenhadas para a composição de *pipelines*,

permitindo a flexível configuração das operações a serem executadas. Isso é especialmente útil para as atividades de pré-processamento, mas também apresenta vantagens para as demais etapas de mineração. No LODE, elas são utilizadas pelos transformadores, do pacote de transformação. Eles configuram uma *pipeline* para a aquisição, extração, pré-processamento e análise de elementos dos modelos suportados pelo ambiente, instanciando objetos adequadas para cada tarefa, tal como apresentado na Figura 5.25b. Concluída a configuração, ele aciona o primeiro objeto da *pipeline*, seja por uma mensagem de inicialização (`start`) ou pelo envio de um elemento pela *pipeline* (mensagem `consume`). A partir desse momento, os demais objetos são acionados automaticamente, repassando os elementos processados até que o último componente entregue os dados para um componente de armazenamento (*buffer*) para posterior recuperação pelo transformador.

Os principais transformadores do LODE são os associados com as transformações de material didático em modelos conceituais, de modelos conceituais em instrucionais, de instrucionais para de interação e de interação em objetos de aprendizagem para televisão digital ou apresentações multimídia (Figura 5.26). Todos eles dependem do pacote de modelos para operação e, em menor proporção, do pacote de mineração (com exceção do transformador de material didático para modelos conceituais).

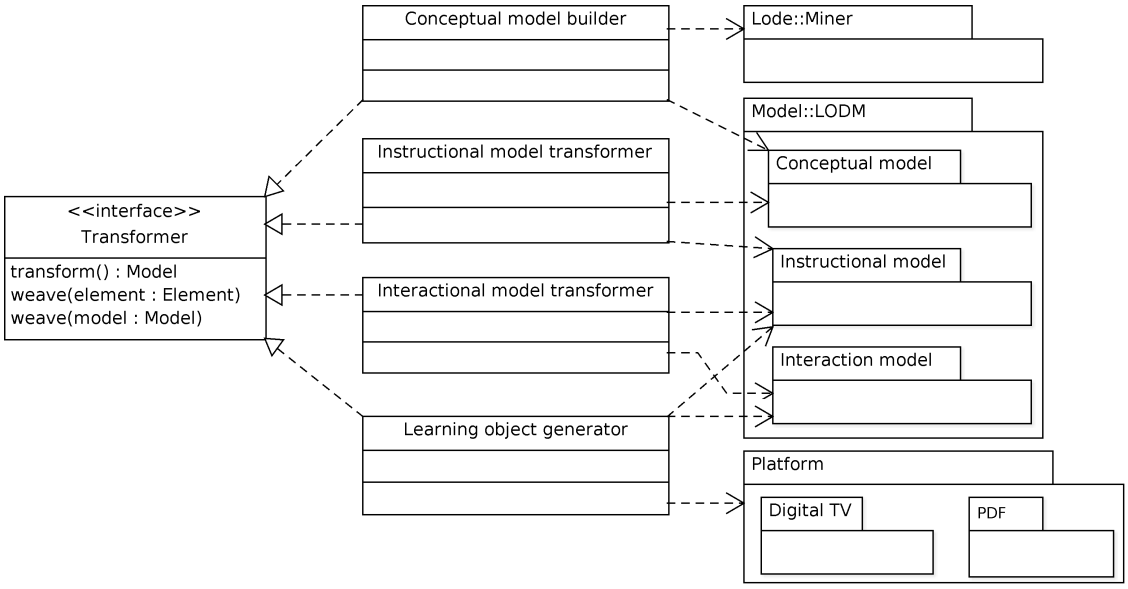


Figura 5.26: LODE: Principais classes do pacote de transformadores.

A transformação de material didático em modelo conceitual é um diferencial do método LODM, construindo-se o modelo conceitual com o auxílio de dados extraídos de textos didáticos. A implementação da extração é realizada com técnicas de mineração de dados e de processamento de linguagem natural, primeiramente para obter a lista de termos candidatos a conceitos e, posteriormente, para recuperar as proposições que vão compor o modelo conceitual.

A extração dos conceitos inicia-se com a aquisição de fontes de texto, representada no módulo de extração pelo pacote `Acquisition`. Esse pacote provê classes para a aquisição de fontes de recursos a partir do sistema de arquivos ou de servidores de mapas conceituais. Para servidores de mapas conceituais, é realizada a busca por serviços compatíveis com o *CmapServer* (CANAS *et al.*, 2006) e obtidos os mapas cujo título ou descrição contenham termos semelhantes ao assunto do objeto de aprendizagem a ser criado. Quanto à busca por arquivos, essa também é realizada a partir do assunto, analisando o repositório de material didático do ambiente.

A recuperação direta de mapas conceituais permite um grau elevado de confiança à extração de conceitos e proposições. No entanto, para tópicos novos, a tendência é que não existam mapas conceituais passíveis de reutilização. Além disso, os conceitos e proposições sugeridos pela extração de textos oferece uma visão complementar, permitindo atestar a qualidade dos mapas reutilizados e acrescentar novos conceitos e proposições.

A extração de texto ocorre, por enquanto, em documentos PDF. No entanto, para o mecanismo de mineração, qualquer tipo de documento ou outra espécie de fonte de dados que forneça textos como resultado é suficiente. Obtendo-se os textos realiza-se o pré-processamento, dividindo o texto em partes menores (*tokens*), correspondentes a termos de uma ou duas palavras. Nessa etapa também se eliminam termos pouco úteis à extração, tais como endereços Web, emails, sinais de pontuação, palavras de uso comum (artigos, adjetivos, advérbios, pronomes, preposições). A última etapa do pré-processamento é a substituição de termos no plural pelo singular.

Após o pré-processamento, analisa-se a frequência de cada termo encontrado no conjunto de textos obtidos, obtendo-se uma medida da importância de cada termo. Ordenando-os em função da importância, separam-se aqueles com uma classificação de percentual de 95% para termos com uma palavra e de 99% para termos com duas palavras. Esses são os termos candidatos a conceitos do mapa conceitual.

A segunda parte da extração recorre a técnicas de processamento de linguagem natural. Para esse fim, utiliza-se a biblioteca OpenNLP (Apache, 2010). A partir do mesmo documento *Portable Document Format* (PDF) utilizado para a mineração de conceitos, extrai-se o texto e o divide em frases. Após a identificação das frases, os termos delas são identificados e identifica-se o tipo de cada um deles (*pos tagging*). O próximo passo é identificar a função de um ou mais termos na frase (*chunking*). Com isso, é possível identificar o sujeito, verbos e objetos da frase, ou seja, os componentes básicos de uma proposição. As frases que contêm ao menos um termo identificado como um conceito (termos obtidos pela mineração de texto) são transformadas em proposições. Cada uma das proposições é adicionada ao modelo conceitual.

O modelo conceitual criado pela extração e posteriormente alterado em ferramentas específicas de edição de mapas conceituais é transformado em um *statechart* segundo as

diretivas definidas na Seção 5.1.2. Caso se opte pela criação do modelo instrucional sem a transformação, é possível editá-lo diretamente em ferramentas de edição de modelos em UML, configurando-se o perfil *LODM Instrucional Profile* e atribuindo os estereótipos aos estados.

Do modelo instrucional, identificam-se os estados que permitirão a escolha do estado que será ativado e a adição de novos estados para tratar a interação do usuário com o objeto de aprendizagem, representada por transições entre estados em função de eventos produzidos pelas ações do usuário.

A partir do modelo de interação, é possível gerar objetos de aprendizagem para diversas plataformas. Na atual versão do ambiente, permite-se a geração de apresentações multimídia em LaTeX/Beamer e documentos NCL para televisão digital, conforme descrito nas subseções a seguir.

5.3.1. Geração de apresentações multimídia

A implementação da geração de objetos de aprendizagem no formato de apresentação com a utilização de arquivos PDF foi realizada com a geração de um documento LaTeX, o qual foi posteriormente compilado para a geração de um documento PDF. O principal pacote LaTeX utilizado foi o `latex-beamer` (TANTAU *et al.*, 2003b).

Cada estado do modelo de interação corresponde a uma sequência de quadros (*slides*) da apresentação. A quantidade de quadros é definida em função do conteúdo a ser apresentado: caso não seja possível incluir todos os elementos instrucionais definidos em um estado básico ou composto do tipo AND em um único quadro, é necessário acrescentar quadros na sequência.

Os diferentes tipos de elementos de informação e itens de informação são apresentados nos quadros em uma caixa configurada de acordo com um modelo específico. No Beamer, modelos (*templates*) podem ser utilizados para definir as cores de primeiro plano, de fundo, o tamanho e a cor das fontes. Por exemplo, um item de informação do tipo conceito (CDT) é implementado pelos comandos apresentados na Código-fonte 5.1. Para cada elemento, define-se um ambiente (`newenvironment`). O comando `setbeamertemplate` configura o modelo a ser utilizado para esse ambiente, especificando comandos a serem executados antes e depois do ambiente.

```
\setbeamercolor{concept title}{bg=blue!30!black, fg=white}
\setbeamercolor{concept body}{bg=blue!05!white, fg=black}

\defbeamertemplate*{concept begin}{default}[1][]{%
        \par%
        \vskip\medskipamount%
        \begin{beamerboxesrounded}[upper=concept title,lower=concept body,%
        shadow=true,#1]{\usebeamerfont{block title}\insertblocktitle}%
        \usebeamerfont{block body}%
 }
```

```
\defbeamertemplate*{concept end}{default}{%
        \end{beamerboxesrounded}%
        \vskip\smallskipamount%
}

\newenvironment<>{block:concept}[1]{%
        \begin{actionenv}#2%
                \def\insertblocktitle{#1}%
                \usebeamertemplate{concept begin}%
}{%
                \usebeamertemplate{concept end}%
        \end{actionenv}%
}
```

Código-fonte 5.1: Modelo em LaTeX/Beamer para a definição de um conceito.

Em apresentações, utilizam-se basicamente quatro tipos de mídia: texto, imagens, áudios e vídeos. Textos e imagens são triviais, exigindo, no máximo, a correção da disposição dos elementos no quadro utilizando-se colunas ou tabelas. A inclusão de áudios e vídeos, no entanto, não é simples. Os principais motivos são a falta de padronização do mecanismo para inclui-los em PDF e a carência de leitores que sejam capazes de executá-los adequadamente. Após investigar o problema (SILVA *et al.*, 2011b), concluiu-se que não existe uma única solução satisfatória que funcione adequadamente em diversas plataformas. Por outro lado, é possível gerar apresentações que funcionam adequadamente para plataformas específicas (Windows, Linux e MacOS). Considerando que a apresentação é gerada automaticamente, é possível gerar versões do mesmo modelo instrucional para os diferentes perfis de utilização.

Em alguns casos, não é possível representar os itens de informação de forma direta (texto, imagem, áudio ou vídeo). Por exemplo, a utilização de mapas conceituais como item de informação se enquadra nesse caso. Embora eles pudessem ser convertidos em imagens e inseridos como tal em um quadro, perder-se-ia a capacidade de exploração (navegação) a partir do mapa, deixando de cumprir uma de suas funções instrucionais. Para sanar esse problema, foi definida uma transformação de mapas conceituais (e de modelos conceituais) para um grafo clicável. O modelo, de preferência importado de um documento CXL (o qual define, além de conceitos e proposições, o posicionamento e as características para representação gráfica dos elementos), é transformado em comandos LaTeX providos pelo pacote PGF/TikZ (TANTAU *et al.*, 2003a). A cada conceito, define-se uma referência (`ref`) para o primeiro quadro da sequência que representa um estado básico ou composto do tipo AND, que por sua vez foram gerados a partir de um conceito do modelo conceitual (todo quadro inicial é etiquetado – `label` – com o conceito que ele aborda).

Além do acesso aos quadros por mapas conceituais, a navegação em apresentações multimídia geradas de acordo com o LODM implementa a semântica do modelo de interação (desprezando-se a temporização). Uma barra de navegação é definida no canto inferior

esquerdo da tela, sem comprometer o espaço disponível para o conteúdo da apresentação propriamente dita. As seguintes ações navegacionais são suportadas: retornar ao primeiro quadro da sequência referente ao estado-pai (se existir o estado-pai), retornar para o quadro anterior da sequência referente ao estado atual (se existir um quadro anterior), avançar para o próximo quadro da sequência referente ao estado atual (se existir um próximo quadro) e avançar para o primeiro quadro do próximo estado (lembrando que no modelo de interação sempre é definida uma sequência padrão).

A implementação dessa forma de navegação (e a representação do aninhamento de estados) na apresentação depende da definição de algumas propriedades para cada quadro: a propriedade `parent` contém o rótulo do primeiro quadro da sequência relativa ao estado-pai (caso ele exista); `hasnext` define se existe um próximo quadro na sequência referente ao estado atual ou se existe um próximo estado; `hasprevious` define se existe um quadro anterior na sequência referente ao estado atual ou se existe um estado anterior. No Código-fonte 5.2 define-se o quadro inicial de um estado básico (ambas as propriedades `hasprev` e `hasnext` são falsas) cujo estado-pai contém o rótulo `cmap:mdd-lo`. Essas propriedades são definidas automaticamente a partir do modelo de interação e, portanto, não precisam ser alteradas.

```
\begin{frame}[parent={cmap:mdd-lo}, hasprev=false, hasnext=false]
\frametitle{Model-driven development of learning objects}
\label{concept:lodm}
\end{frame}
```

Código-fonte 5.2: LODE: Definição de propriedades de navegação em uma apresentação LaTeX/Beamer.

A representação de outras transições deve ser feita explicitamente utilizando-se o comando `refie`. Ele define dois parâmetros: o rótulo do estado-alvo e a âncora que ativará a transferência no estado-origem, Por exemplo, no Código-fonte 5.3 é definida uma transição para o estado rotulado `JUnit shakedown`, que é acionada pelo acionamento de um botão em um quadro do estado corrente.

```
\refie{example:junit-shakedown}{\beamerbutton{Example: JUnit shakedown}}
```

Código-fonte 5.3: LODE: Exemplo de definição de transição entre estados.

Embora a geração dos arquivos LaTeX seja automática, buscou-se uma solução que facilitasse a organização e edição manual dos conteúdos gerados. Cada estado composto do tipo XOR define uma seção da apresentação e um diretório. Os subestados são definidos por subseções e subdiretórios e assim por diante, definindo uma hierarquia no sistema de arquivo e no documento LaTeX. Cada estado básico ou compostos do tipo AND é representado por um arquivo, nomeado tal como o rótulo do estado e armazenado no diretório referente ao estado que contém. No Código-fonte 5.4 é apresentado um exemplo para a representação

dos estados compostos e aninhados `JUnit` e `Test case`; neste último define-se um estado composto do tipo AND rotulado `junit-runner`, representando um conjunto de elementos instrucionais.

```
\section{JUnit}
\include{main/junit}

\subsection{Test case}
\include{main/junit/test-case}
\include{main/junit/test-case/junit-runner}
```

Código-fonte 5.4: LODE: Exemplo de representação de estados em um documento LaTeX/Beamer.

5.3.2. Geração de documentos NCL

O desenvolvimento de aplicações declarativas para TVD inicia-se com a definição das regiões que serão utilizadas para a apresentação do conteúdo. Por exemplo, define-se uma região principal que corresponde à tela inteira do principal dispositivo de apresentação (a própria televisão) e, nesta região, definem-se outras regiões a serem utilizadas.

No caso de objetos de aprendizagem para TVD, observa-se a necessidade, além da região principal, da definição de regiões a escolha de opções, dividindo-se a tela em partes. Também é útil a definição de uma região para colocar um pequeno vídeo sobreposto a um principal (*Picture in Picture*). A cada região é definido um descritor para sua utilização. Embora outros descritores sejam necessários para implementar uma aplicação para TVD, é útil ter esses padrões já definidos. Nos exemplos apresentados nesta seção, utilizam-se regiões e descritores definidos dessa forma. Os demais elementos – conectores, regras e descritores adicionais – serão definidos de acordo com o modelo de interação do objeto de aprendizagem.

Documentos NCL descrevem aplicações para televisão digital como contextos aninhados, definindo um conjunto de mídias (vídeos, figuras, pequenos programas em Lua e Java) para cada contexto e relacionamentos entre esses contextos que, quando ativados, executam ações que podem alterar o estado dos contextos da aplicação. Esse modelo é muito próximo daquele exposto no modelo de interação, em que existem estados aninhados e transições entre os estados. A geração de um documento NCL consiste, portanto, na transformação dos estados em contextos (`context`) e das transições em relacionamentos (`link`).

Um modelo de interação define quatro tipos de estados: o estado básico e os estados compostos do tipo AND, XOR e XOR DD. O estado básico, que representa um item de informação ou elemento instrucional, corresponde exatamente a um contexto ou, de forma simplificada, a uma mídia. Como a tendência é adicionar mais mídias em um contexto durante o seu desenvolvimento/evolução, é preferível utilizar um contexto para representá-lo,

tal como exposto no Código-fonte 5.5. Todo contexto em NCL possui um identificador que deve ser único no documento. Uma escolha adequada é o próprio nome do estado que o originou e, apenas para deixar claro, pós-fixamos-o com a palavra `State`. Dessa forma, é trivial saber que o identificador se trata de um estado sobre um determinado conceito.

```
<context id="softwareFailureState">
    <port id="softwareFailureStatePort" component="softwareFailureConcept" />
    <media id="softwareFailureConcept" src="media/WhatIsASoftwareFailure.mp4"
        descriptor="wholeScreenMedia"/>
</context>
```

Código-fonte 5.5: LODE: Geração de contexto a partir de estado básico do modelo de interação.

Durante a execução de um documento NCL, ao iniciar um contexto são ativados os elementos (vértices de mídia ou outros contextos) que estão associados à porta (elemento `port`) do contexto. Para um estado básico, deseja-se iniciar apenas um vértice de mídia; logo, é criada uma porta para acesso ao respectivo vértice.

Uma característica do modelo de interação é que deve ser possível a transição entre os estados de forma automática. Por padrão, o tempo de vida de um contexto é definido pelos contextos e nós de mídia aninhados a ele. No exemplo descrito em Código-fonte 5.5, utiliza-se um vídeo MPEG-4 cujo tempo de vida corresponde à duração dele. No entanto, caso fosse utilizada uma mídia atemporal, tal como uma imagem ou texto (documento HTML), o tempo de vida seria infinito. Para contornar essa questão, basta definir, no descritor associado ao elemento (elemento `wholeScreenStaticMedia`), o tempo de duração máximo, como mostrado na `listing:lode:ncl:expiring-descriptor` no atributo `explicitDur` (no caso, atribui-se um tempo limite de 10 segundos).

```
<descriptor id="wholeScreenStaticMedia"
    region="wholeScreenRegion"
    explicitDur="10s" />
```

Código-fonte 5.6: LODE: Descritor para mídias com tempo de vida infinito.

O próximo tipo de estado a ser gerado é o composto do tipo AND. Nesse estado, todos os estados-filho são ativados, executando-os em paralelo. O código equivalente em NCL para representá-lo consiste na definição de um contexto para o estado AND e, aninhado a esse contexto, um contexto para cada estado nele definido. Considerando o estado `Mistake` do modelo de interação da Figura 5.11, o código corresponde seria este apresentado no Código-fonte 5.7.

```
<context id="mistakeState">
    <port id="mistakeStatePort1" component="typicalMistakeState" />
    <port id="mistakeStatePort2" component="programmingMistakeState" />
```

```
    <context id="typicalMistakeState">
        <port id="typicalMistakeStatePort" component="typicalMistakeInstructionalElement" />
        <media id="typicalMistakeInstructionalElement"
            src="media/SomeExamplesOfTypicalProgrammingMistakes.mp4"
            descriptor="dualScreenMediaWithPip_Main" />
    </context>

    <context id="programmingMistakeState">
        <port id="programmingMistakeStatePort" component="programmingMistakeConcept" />
        <media id="programmingMistakeConcept"
            src="media/WhatIsAProgrammingMistake.mp4"
            descriptor="dualScreenMediaWithPip_Pip"/>
    </context>
</context>
```

Código-fonte 5.7: LODE: Geração de um contexto a partir de um estado composto do tipo AND do modelo de interação.

De forma semelhante ao realizado quanto ao estado básico, define-se um contexto com um identificador único para o estado (`Mistake`). No entanto, diferentemente do estado básico, ao invés de uma porta, são definidas duas portas: uma para cada estado (contexto) que devem ser acionados. Assim, ao iniciar o contexto `mistakeState`, os contextos aninhados também serão iniciados.

Os contextos aninhados `typicalMistakeState` e `programmingMistakeState` estão associados a um vídeo cada um. Como eles serão executados simultaneamente, é necessário definir como eles serão apresentados, sem causar sobreposição de ambos. Essa configuração é realizada pelos descritores. No caso desse exemplo, considera-se que o vídeo sobre a definição de um erro de programação contenha um professor explicando o que é um erro (engano) típico. No outro contexto, é mostrada uma animação com alguns exemplos com os erros explicados e os efeitos que eles causam no software (a falha). Uma forma de apresentá-los é manter o vídeo do professor na tela inteira e o dos exemplos em uma região menor, no canto do vídeo. O Código-fonte 5.8 contém o código para implementar essa configuração. Além dos descritores para cada mídia, é preciso definir as regiões (elementos `region`) e garantir que o vídeo dos exemplos (relacionado com o descritor `dualScreenMediaWithPip_Pip`) irá sobrepor o vídeo do professor (caso contrário, ele não será visível). A ordem das regiões no dispositivo de apresentação (tela da televisão) é definida pelo atributo `zIndex` da especificação da região: a região com índice maior tem prioridade sobre as com índice inferior ao dela. Assim, ao designar um índice 100 à região do PIP e 10 para a região da tela inteira, garante-se que o contexto `mistakeState` será reproduzido corretamente.

```
<regionBase>
    <region id="wholeScreenRegion" width="100%" height="100%" zIndex="10"/>
    <region id="pipRegion" top="70%" left="70%" width="25%" height="25%" zIndex="100"/>
</regionBase>
```

```
<descriptorBase>
    <descriptor id="dualScreenMediaWithPip_Main"
        region="wholeScreenRegion" />
    <descriptor id="dualScreenMediaWithPip_Pip"
        region="pipScreenRegion" />
<descriptorBase>
```

Código-fonte 5.8: LODE: Regiões e descritores para mídias reproduzidas simultaneamente.

A representação de um estado composto do tipo XOR é semelhante ao do estado AND: define-se um contexto para o estado e um contexto aninhado para cada subestado. Considerando o estado `Failure` do modelo de interação definido na Figura 5.11, seu contexto é definido como no exemplo do Código-fonte 5.9. Define-se um contexto com um identificador único para o estado (`failureState`), mas, diferentemente do estado AND, apenas uma porta é definida. Isso se deve ao fato de, no estado XOR, apenas um subestado deve estar ativo. Logo, ao acionar o estado, a porta deve direcionar para o subestado inicial (no caso `SoftwareFailure`).

```
<context id="failureState">
    <port id="failureStatePort" component="softwareFailureState" />

    <context id="softwareFailureState">
        <port id="softwareFailureStatePort" component="softwareFailureConcept" />
        <media id="softwareFailureConcept" src="media/WhatIsASoftwareFailure.mp4"
            descriptor="wholeScreenMedia"/>
    </context>

    <context id="therac25State">
        <port id="softwareFailureStatePort" component="softwareFailureConcept" />
        <media id="softwareFailureConcept" src="media/WhatIsASoftwareFailure.mp4"
            descriptor="wholeScreenMedia"/>
    </context>
</context>
```

Código-fonte 5.9: LODE: Geração de um contexto a partir de um estado composto do tipo XOR do modelo de interação.

Como o conteúdo de cada contexto aninhado será apresentado de forma exclusiva (um por vez), eles devem ocupar a tela inteira, escolhendo-se o mesmo descritor utilizado para no exemplo do estado básico (`wholeScreenMedia`).

Em um estado XOR, a sequência com que serão acionados os contextos deve ser explicitamente informada. Por exemplo, observa-se no modelo (Figura 5.11) uma transição do estado `Software failure` para `Therac-25`. No Código-fonte 5.9, essa transição não é definida: na prática, caso aquele contexto fosse executado, apenas o vídeo associado ao contexto `softwareFailureState` seria apresentado. Para corrigir esse comportamento, é preciso definir uma transição entre os estados `softwareFailureState` e `therac25State` de forma que, ao terminar a execução do primeiro, acione-se o segundo. Isso é realizado

pela especificação de uma ligação (`link`) com um conector do tipo `onBeginStart`, tal como apresentado no Código-fonte 5.10.

```
<context id="failureState">
    [...]
    <link id="fromSoftwareFailureState_toTherac25State" xconnector="conn#onEndStart">
        <bind role="onEnd" component="softwareFailureState" />
        <bind role="start" component="therac25State" />
    </link>
</context>
```

Código-fonte 5.10: LODE: Ligação que estabelece o acionamento de um contexto ao término de outro.

Uma ligação possui um identificador (como todo elemento de um documento NCL). Para esse tipo de transição, adota-se a estratégia de nomeá-la de acordo com os identificadores dos contextos relacionados. No entanto, o atributo mais importante é o definido em `xconnector`: o tipo de transição a ser realizada. Ele é definido por conectores, mais especificamente conectores causais. Para o exemplo apresentado, é necessário um conector cuja condição de acionamento seja o término de um contexto e cuja ação seja o disparo de outro contexto. A especificação desse conector consiste daquela mostrada no Código-fonte 5.11: no papel de término, configura-se o contexto `softwareFailureState`; para a ação, configura-se o contexto para `therac25State`.

```
<head>
    <connectorBase>
        <causalConnector id="onEndStart">
            <simpleCondition role="onEnd"/>
            <simpleAction role="start"/>
        </causalConnector>
    </connectorBase>
</head>
```

Código-fonte 5.11: LODE: Definição do conector onEndStart.

O único tipo de estado do modelo de interação que resta representar é o composto do tipo XOR DD. Ele permite que apenas um de seus subestados esteja ativo em um determinado instante, tal como o estado XOR, mas também oferece a opção do usuário escolher qual dos estados, dentre todos os estados-filhos, deve ser acionado. No caso do modelo de interação, existe um tempo máximo para realizar essa escolha, após o qual o estado inicial padrão é iniciado (tal como no estado XOR normal).

A implementação desse comportamento em NCL não é tão trivial quanto nos casos anteriores. Ela envolve a criação de um contexto adicional para a realização da escolha a partir da interação do usuário e o uso de um vértice de adaptação (`switch`). Um `switch` funciona como um contexto que permite o acionamento de subcontextos de acordo com uma

regra. Considere a definição do elemento `switch` e o estado `Test technique` (XOR DD) do modelo de interação da Figura 5.11. Ao invés de definir um `context`, definimos um `switch` (contexto adaptativo), conforme apresentado no Código-fonte 5.12. A adaptação consiste na escolha de um dos três estados aninhados (cada qual representada por um contexto e uma mídia). A porta de entrada do contexto adaptativo é definida pelo elemento `switchPort`. Ele define três mapeamentos, um para cada possível adaptação. A escolha sobre quais contextos serão ativados é estabelecida por regras associadas a cada um deles, conforme especificado no elemento `bindRule`.

```
<switch id="testTechniqueState">
    <switchPort id="testTechniqueStatePort">
        <mapping component="exhaustiveTestingState" />
        <mapping component="randomTestingState" />
        <mapping component="partitionTestingState" />
    </switchPort>

    <bindRule constituent="exhaustiveTestingState" rule="stateSelectionOption1Rule" />
    <bindRule constituent="randomTestingState" rule="stateSelectionOption2Rule" />
    <bindRule constituent="partitionTestingState" rule="stateSelectionOption3Rule" />

    <defaultComponent component="exhaustiveTestingState" />

    <context id="exhaustiveTestingState">
        <port id="exhaustiveTestingStatePort" component="exhaustiveTestingStateConcept" />
        <media id="exhaustiveTestingStateConcept" src="media/WhatIsExhaustiveTesting.mp4"
            descriptor="wholeScreenMedia"/>
    </context>

    <context id="randomTestingState">
        <port id="randomTestingStatePort" component="randomTestingStateConcept" />
        <media id="randomTestingStateConcept" src="media/WhatIsRandomTesting.mp4"
            descriptor="wholeScreenMedia"/>
    </context>

    <context id="partitionTestingState">
        <port id="partitionTestingStatePort" component="partitionTestingStateConcept" />
        <media id="partitionTestingStateConcept" src="media/WhatIsPartitionTesting.mp4"
            descriptor="wholeScreenMedia"/>
    </context>
</switch>
```

Código-fonte 5.12: LODE: Geração de um contexto de adaptação a partir de um estado composto do tipo XOR DD do modelo de interação.

As regras são definidas no cabeçalho do documento NCL (`head`), dentro do elemento `ruleBase`, da forma apresentada no Código-fonte 5.13. Cada regra compara uma variável (definida pelo atributo `var`) de acordo com o operador de comparação (atributo `comparator`) quanto a um valor específico (atributo `comparator`). A resposta à avaliação da regra é uma valor booleano.

```
<ruleBase>
    <rule id="stateSelectionOption1Rule"
        var="service.currentFocus" comparator="eq" value="1" />
    <rule id="stateSelectionOption2Rule"
        var="service.currentFocus" comparator="eq" value="2" />
    [..]
</ruleBase>
```

Código-fonte 5.13: LODE: Regras para seleção de opções do contexto adaptativo que representa um estado XOR DD.

A implementação da função de escolha de estados constitui-se em mostrar um elemento de mídia relativo a cada opção e a escolha, usando os cursores do controle remoto, de uma opção. Ela é implementada em um contexto acima daquele do contexto adaptativo, conforme apresentado no Código-fonte 5.14. Nele são utilizados os cursores para selecionar uma opção dentre as disponíveis, alterando-se o valor da variável `service.currentFocus` a cada mudança de foco (elemento selecionado). Como passo preliminar para acessar essa variável, é preciso definir uma mídia do tipo propriedades do sistema e que declare o acesso a ela, conforme definido no vértice de mídia `nodeSettings`.

```
<context id="testTechniqueChoiceState">
    <media id="nodeSettings" type="application/x-ginga-settings">
        <property name="service.currentFocus"/>
    </media>
    <media id="xorDDStateChoice" src="media/stateSelectionQuestion.png"
        descriptor="xorDDStateChoiceDescriptor"/>
    <media id="testTechniqueOption1" src="media/WhatIsExhaustiveTesting.mp4"
        descriptor="xorDDStateN3Option1Descriptor" />
    <media id="testTechniqueOption2" src="media/WhatIsRandomTesting.mp4"
        descriptor="xorDDStateN3Option2Descriptor" />
    <media id="testTechniqueOption3" src="media/WhatIsPartitionTesting.mp4"
        descriptor="xorDDStateN3Option3Descriptor" />

    <!-- Links placeholder -->

    <switch id="testTechniqueState">[...]</switch>
</context>
```

Código-fonte 5.14: LODE: Definição do contexto para escolha de opção e acionamento do respectivo contexto.

A tela em que serão apresentadas as escolhas contém uma imagem no topo solicitando ao usuário que escolha uma opção e, abaixo, as opções disponíveis. Para cada opção, define-se um vértice de mídia associado (1) com um vídeo ou imagem representativo daquela opção e (2) com um descritor referente à opção. Os descritores definem a região em que será apresentada a mídia e qual o comportamento do foco ao se ativar os botões direcionais do controle remoto. Observando o Código-fonte 5.15, o atributo `focusIndex` define o valor do foco quando selecionado o descritor e os atributos `moveLeft` e `moveRight` definem qual será

o valor do foco quando se acionarem as setas para esquerda e direita, respectivamente.

```
    <descriptor id="xorDDStateChoiceDescriptor" region="xorDDStateChoiceDescriptorRegion"
        explicitDur="10s" />
    <descriptor id="xorDDStateN3Option1Descriptor" region="trioVideoRegion_1"
        focusIndex="1" moveLeft="3" moveRight="2" />
    <descriptor id="xorDDStateN3Option2Descriptor" region="trioVideoRegion_2"
        focusIndex="2" moveLeft="1" moveRight="3" />
    <descriptor id="xorDDStateN3Option3Descriptor" region="trioVideoRegion_3"
        focusIndex="3" moveLeft="2" moveRight="1" />
```

Código-fonte 5.15: LODE: Definição dos descritores referentes às opções para um estado XOR DD.

Finalmente, para implementar toda a lógica de seleção de opções, é preciso definir as ligações para (1) ativar a pergunta, (2) mostrar as opções e (3) acionar o contexto adaptativo com base na seleção e parar de mostrar a pergunta e as opções. O código referente a essas ações encontra-se no Código-fonte 5.16. Em poucas palavras, ele funciona da seguinte forma. Ao entrar no estado de escolha de opções, é mostrada a pergunta. Uma ligação, ativada pelo acionamento da pergunta, ativa as opções. O usuário move o cursor e, ao selecionar uma opção, ativa a ligação `fromXorDDStateChoiceToSelectedOption`. Essa ligação interrompe a pergunta e as opções e inicia o estado `testTechniqueState`. Ele, por sua vez, irá verificar, com base na variável `service.currentFocus`, qual das regras foram satisfeitas e acionar os contextos associados a elas.

```
<!-- 1. Start question -->
<port id="testTechniqueChoiceStatePort" component="xorDDStateChoice"/>

<!-- 2. Show options -->
<link id="fromDDStateChoiceToOptions" xconnector="onBegin1StartN">
    <bind component="xorDDStateChoice" role="onBegin"/>
    <bind component="testTechniqueOption1" role="start"/>
    <bind component="testTechniqueOption2" role="start"/>
    <bind component="testTechniqueOption3" role="start"/>
</link>

<!-- 3. Start adaptative context using selection data -->
<link id="fromXorDDStateChoiceToSelectedOption" xconnector="onSelectionNStartNStopN">
    <bind component="testTechniqueOption1" role="onSelection"/>
    <bind component="testTechniqueOption2" role="onSelection"/>
    <bind component="testTechniqueOption3" role="onSelection"/>
    <bind component="testTechniqueState" role="start"/>
    <bind component="xorDDStateChoice" role="stop"/>
    <bind component="testTechniqueOption1" role="stop"/>
    <bind component="testTechniqueOption2" role="stop"/>
    <bind component="testTechniqueOption3" role="stop"/>
</link>
```

Código-fonte 5.16: LODE: Definição de ligações referentes às opções para um estado XOR DD.

Finalmente, resta implementar o comportamento padrão, ou seja, se nenhuma opção for selecionada em um determinado espaço de tempo, deve-se acionar o contexto padrão do contexto adaptativo (definido no elemento `defaultComponent` no Código-fonte 5.10). Para conseguir isso, define-se um tempo máximo de duração para a mídia `xorDDStateChoice` em seu descritor `xorDDStateChoiceDescriptor` e uma ligação que, ao interrompê-la, interrompe também as opções e aciona o contexto adaptativo, tal como apresentado no Código-fonte 5.17.

```
<link id="fromXorDDStateChoiceToAriane5LiftOffState" xconnector="conn#onEndStartNStopN">
    <bind component="testTechniqueState" role="start"/>
    <bind component="xorDDStateChoice" role="onEnd"/>
    <bind component="xorDDStateOption1" role="stop"/>
    <bind component="xorDDStateOption2" role="stop"/>
</link>
```

Código-fonte 5.17: LODE: Definição de ligações referentes às opções para um estado XOR DD.

Com isso, encerram-se as instruções necessárias para a representação de um estado composto XOR DD.

5.4. Considerações finais

O método LODM apresenta uma abordagem integrada e dirigida a modelos para o desenvolvimento de objetos de aprendizagem multimídia e interativos, enfatizando o reúso e a transformação de modelos durante todas as atividades. Além disso, ele busca ser escalável, almejando o uso em situações reais e a elaboração de estudos experimentais para avaliar os efeitos do método na aprendizagem e no ensino, analisando tanto a perspectiva absoluta da qualidade do objeto de aprendizagem quanto às perspectivas de ensino – produtividade do professor, maior disponibilidade para definir atividades de aprendizagem mais relevantes – e de aprendizagem – consequência da qualidade do objeto de aprendizagem e do projeto instrucional que foi possível concretizar com o método.

Mapas conceituais são uma ferramenta bem conhecida para a modelagem de conhecimento (FORD *et al.*, 1991; ZANTING *et al.*, 2003; COFFEY *et al.*, 2006; HOFFMAN *et al.*, 2008), mas o emprego dela para a geração de modelos no contexto de um método de desenvolvimento de objetos de aprendizagem, combinando técnicas de mineração de dados e de processamento de linguagem natural, torna a solução única, além de oferecer um mecanismo objetivo para auxiliar o desenvolvimento do modelo conceitual de um objeto de aprendizagem.

Capítulo 6

Estudos de viabilidade

A abordagem *Learning Object Development* (LOD) foi avaliada de forma preliminar em dois contextos: no desenvolvimento de objetos de aprendizagem na forma de apresentações multimídia e na forma de uma aplicação interativa para televisão digital.

No primeiro contexto, apresentado na Seção 6.1, o objeto de aprendizagem desenvolvido destina-se ao treinamento para o emprego da ferramenta JaBUTi em atividades de teste de software. São considerados, quanto ao método LODM, os aspectos multimídia do objeto de aprendizagem e os elementos básicos de navegação definidos pelo modelo de interação e, quanto ao processo LODP, o desenvolvimento de forma aberta. O objeto de aprendizagem produzido foi avaliado por especialistas do domínio.

O segundo objeto de aprendizagem, também no contexto de teste de software, trata do aspecto interativo no ambiente de TVD (Seção 6.2). Imagina-se um cenário em que o receptor da televisão suporte interatividade, tanto de aplicações procedimentais como declarativas no padrão do SBTVD, e esteja em execução um programa que busca por objetos de aprendizagem em um repositório público ou no próprio fluxo de dados transmitido pela emissora de televisão relacionados com o programa em transmissão e visto pelo usuário. O cenário se completa com a transmissão de um documentário sobre viagens espaciais, no qual se encontra um objeto de aprendizagem sobre a explosão do foguete Ariane 5 (voo 501), um caso real em que um erro de software causou um prejuízo significativo (da escala de dezenas

de milhões de euros). O objeto de aprendizagem descreve o acidente, o erro que o causou e permite a aplicação de uma técnica de teste, baseada em análise de mutantes, que poderia ter detectado o erro e evitado o desastre. Neste estudo, são tratados os aspectos de multimídia e de interação do objeto de aprendizagem produzido.

Na Seção 6.3, compara-se a abordagem LOD com os requisitos estabelecidos no Capítulo 4 e com outras técnicas para desenvolvimento de objetos de aprendizagem baseadas em modelos ou para televisão digital.

6.1. Estudo 1 – Apresentação multimídia

Este estudo de caso foi realizado no contexto universitário, na modalidade presencial, utilizando-se uma apresentação multimídia definida em um documento PDF. O objeto de aprendizagem destinou-se ao ensino da técnica de teste estrutural de software com emprego da ferramenta JaBUTi (VINCENZI *et al.*, 2003).

6.1.1. Planejamento

O objetivo deste estudo foi a avaliação da navegação e do aspecto multimídia do objeto de aprendizagem, observando-se a implementação da transição entre os estados pelos mecanismos naturais de transição do modelo de interação.

O modelo conceitual foi elaborado a partir do manual de uso da ferramenta e outros conteúdos associados, inclusive modelos conceituais utilizados em objetos de aprendizagem sobre teste de software desenvolvidos anteriormente.

Para a construção do modelo instrucional, foram utilizados os itens de informação do CDT (conceitos, fatos, princípios e procedimentos) e mapas conceituais, conforme definidos no perfil *LODM Instructional Profile*. Vídeos da operação da ferramenta JaBUTi (na forma de *screencasts*) foram criados de acordo com as instruções definidas em estudo anterior (SILVA *et al.*, 2011d) e utilizados como elementos instrucionais para fins explanatórios. Em casos mais simples, utilizaram-se animações (sequência de figuras reproduzidas automaticamente) ao invés de vídeos, mantendo-se o propósito explanatório.

O modelo de interação considerou os eventos típicos de interação em apresentações multimídia – avançar e retroceder – e aqueles automaticamente gerados pelo modelom, tais como os conceitos tratados em cada estado). A partir do modelo de interação, as diretivas para transformação em uma apresentação multimídia foram manualmente aplicadas, com exceção da transformação de mapas conceituais, para a qual foram implementados no LODE transformadores para LaTeX utilizando diferentes pacotes (`dottexi` e `tikz`, sendo este último o adotado no material gerado).

6.1.2. Resultados

A atividade de aprendizagem tratou das funcionalidades da ferramenta JaBUTi e dos conceitos de teste de software relacionados. O objeto de aprendizagem, denominado JaBUTi-TT, foi organizado em 10 outros objetos de aprendizagem, cada qual com seus respectivos modelos conceituais, instrucionais e de interação. O primeiro objeto de aprendizagem corresponde à raiz do modelo conceitual (em uma visão global) e está ligado a todos os demais módulos, como segue:

- Visão geral sobre a JaBUTi no contexto de teste de software, representando o principal objeto de aprendizagem. Apresenta os principais conceitos associados à ferramenta JaBUTi quanto à atividade de teste de software.
- Teste de software. Compreende a definição dos conceitos de erros, técnicas, critérios e requisitos de teste.
- Teste estrutural de software. Compreende a definição dos conceitos básicos da técnica de teste baseada na estrutura do programa. Esta é a técnica utilizada pela JaBUTi.
- JaBUTi. Apresenta as informações sobre a ferramenta (técnicas de teste, licença de software, documentação, exemplos).
- Interface gráfica da JaBUTi. Apresentação da interface gráfica da JaBUTi.
- Ferramenta de análise de cobertura. Aborda o teste de software com a técnica estrutural, compreendendo a instrumentação da aplicação, a execução dos casos de teste, a análise do rastro de execução (*trace*) e o cálculo da cobertura com respeito aos critérios de teste estrutural implementados pela JaBUTi.
- Ferramenta de particionamento de código. Permite a localização de erros a partir dos dados de cobertura e a satisfação ou não de requisitos de teste gerados a partir de critérios de fluxo de controle (todos os nós).
- Ferramenta de métricas. Permite a análise estática do programa em teste e a medição de suas propriedades, servindo de subsídio para a definição de casos de teste.
- Criação de uma sessão de teste na JaBUTi. Trata da criação de um novo projeto e do carregamento de um projeto existente para o teste de um software.
- Estudos experimentais para teste de software. Definição de estudos empíricos para a avaliação do teste de software, estabelecendo a força (*strength*), a eficácia e o custo de uma técnica de teste de software.

Os modelos conceituais foram criados a partir do manual de uso da ferramenta (VINCENZI *et al.*, 2010b), de um curso sobre teste de software (VINCENZI *et al.*, 2010a) e de nossa experiência no ensino de teste de software e no desenvolvimento de melhorias para a ferramenta JaBUTi no contexto do projeto QualiPSo. Um total de 154 conceitos e 209 relacionamentos

foram identificados nos 10 modelos conceituais construídos[1] Na Figura 6.1, o modelo conceitual para o primeiro e principal objeto de aprendizagem é apresentado, mostrando os principais conceitos encontrados pelo testador durante as atividades de teste de software e as atividades às quais a JaBUTi pode ser utilizada.

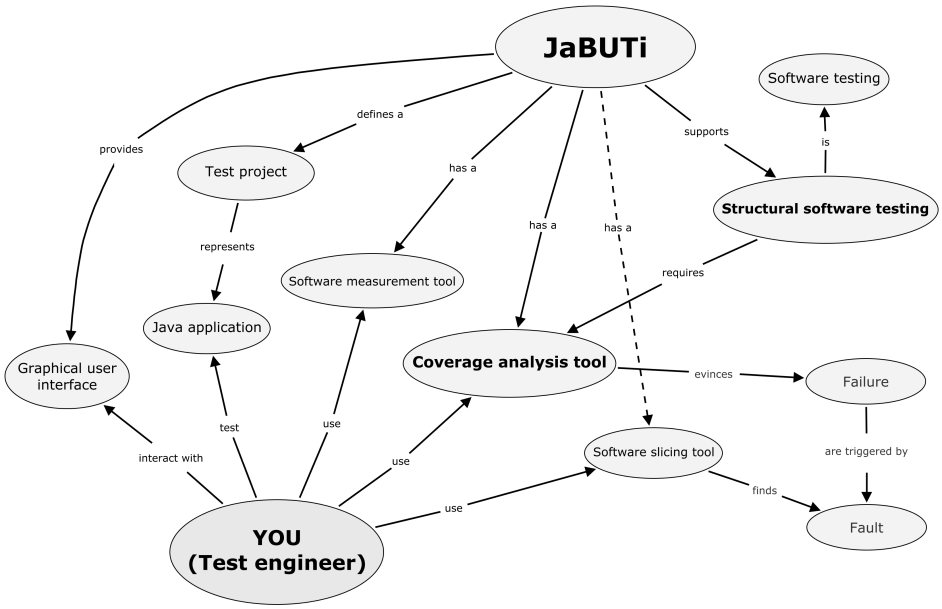


Figura 6.1: JaBUTi-TT: Modelo conceitual sobre a JaBUTi e sua relação com as atividades de teste de software.

Para cada modelo conceitual foi gerado um modelo instrucional, conforme as diretivas apresentadas no Capítulo 5. Por exemplo, considerando o modelo conceitual da Figura 6.1, foi gerado o modelo apresentado na Figura 6.2.

A partir das mesmas fontes de dados utilizadas para a construção do modelo conceitual (VINCENZI *et al.*, 2010b, 2010a), foram extraídos, para cada conceito, recursos que poderiam ser utilizados como itens de informação. Esses recursos foram armazenados em uma aplicação hipertexto do tipo *wiki*, registrados em uma página (a qual representa um conceito) e classificados quanto ao tipo de item de informação, tal como mostrado na Figura 6.3 para o conceito `test requirement`. A escolha por uma *wiki* não foi arbitrária. A abordagem LOD preconiza o desenvolvimento de objetos de aprendizagem e uma *wiki* é um mecanismo ágil e acessível para o compartilhamento de conteúdos (CUNNINGHAM, 2005; SILVA, 2005). Além disso, ela favorece o reúso do conhecimento relacionado aos conceitos, um dos objetivos do emprego da abordagem. No caso específico dos objetos de aprendizagem desenvolvidos durante este doutorado, todos os itens de informação estão disponíveis em `http://www.labes.icmc.usp.br/projects/lod/wiki/`.

[1] Os modelos encontram-se disponíveis em `http://goo.gl/7bk9N` ou `http://www.labes.icmc.usp.br:10888/rid=1H96L16KJ-1LTLQKQ-162G/JaBUTi`.

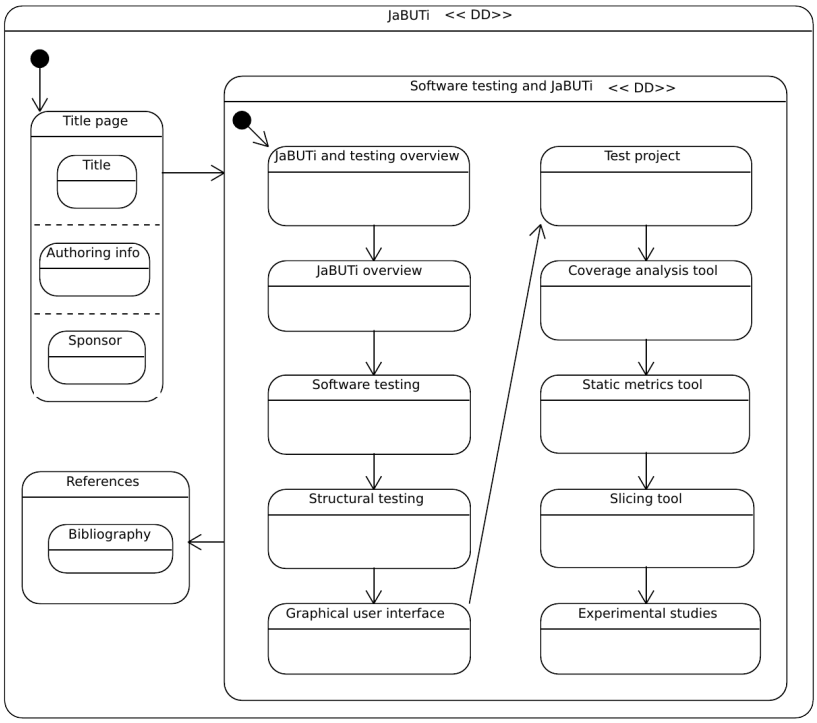


Figura 6.2: JaBUTi-TT: Modelo instrucional sobre a JaBUTi e sua relação com as atividades de teste de software.

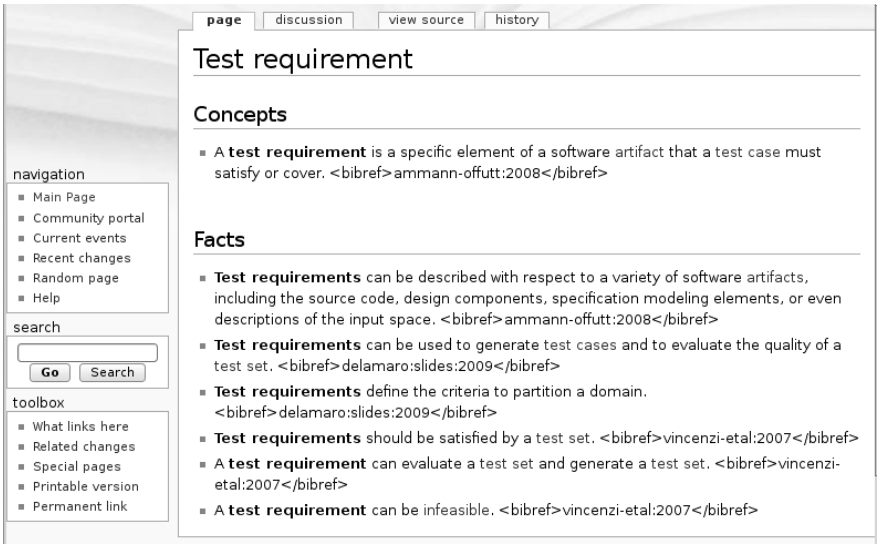


Figura 6.3: JaBUTi-TT: Página *wiki* com recursos relacionados a itens de informação associados ao conceito de requisito de teste.

Conforme dito anteriormente, o modelo de interação deste objeto de aprendizagem é, em essência, idêntico ao modelo instrucional: os eventos de interação de uma apresentação multimídia (avanço e retrocesso de quadros) são representados pelas transições implícitas do modelo.

A geração da apresentação, na forma de um documento LaTeX/Beamer, foi realizada com o auxílio da ferramenta LODE, resultando em 129 arquivos. Em linhas gerais, cada

arquivo representa um estado composto do tipo AND. No total, 301 quadros foram definidos, contendo 279 blocos com elementos instrucionais, dos quais 50 são de elementos instrucionais sem associação com um modelo de informação e 229 são blocos de itens de informação: 103 do tipo conceito, 104 de fatos, 2 de princípios e 20 de procedimentos (Tabela 6.1).

Tabela 6.1: JaBUTi-TT: Quantidade de elementos dos modelos conceitual e instrucional.

Modelo	**Elemento**	**Quantidade**
Modelo conceitual	Conceitos	154
	Proposições	209
Modelo instrucional	Conceitos	103
	Fatos	104
	Princípios	2
	Procedimentos	20
	Outros	50

Exemplificando, objeto de aprendizagem possui o quadro apresentado na Figura 6.4. Ele contém um mapa conceitual gerado a partir do modelo conceitual da Figura 6.1. Nesse quadro também é possível identificar os elementos de navegação no canto da lateral esquerda. Nesse quadro, que representa um estado básico, é permitida a transição para o estado pai (a página de título) e para o próximo estado (um estado-irmão).

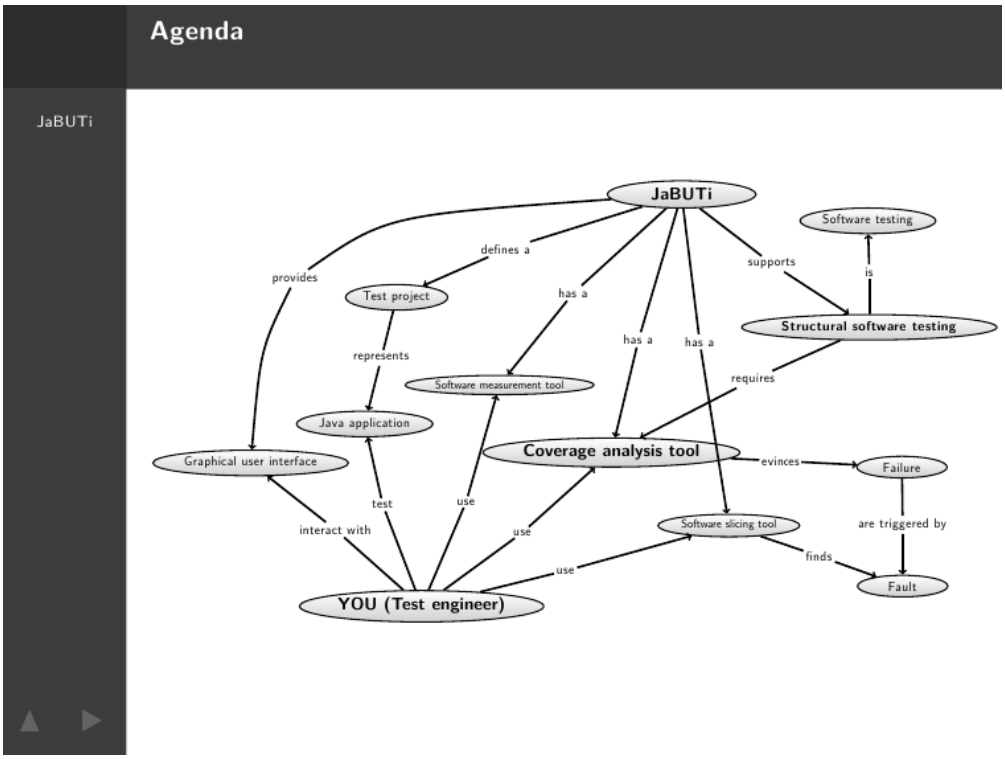


Figura 6.4: JaBUTi-TT: Quadro que representa um estado do modelo de interação.

O quadro 30 do objeto de aprendizagem gerado, apresentado na Figura 6.5, representa um estado composto do tipo AND cujo pai é um estado XOR DD. Ele trata do conceito de

oráculo e define um bloco de conceito e um bloco de fatos. A navegação é realizada pelo menu lateral da esquerda: são apresentados os estados irmãos (sobre qualidade de software, erros, teste de software), para os quais se pode navegar diretamente (tal como previsto no modelo). Além disso, na parte inferior, existem os botões direcionais para realizar a transição para o estado-pai (primeiro quadro da sequência que representa o estado pai), para o estado-irmão anterior e para o próximo estado-irmão. No canto direito, logo abaixo do bloco de fatos, é definido um botão que ativa a transição deste estado para o estado que contém um elemento instrucional explanativo, definindo um exemplo sobre o tipo de oráculo que trata este estado.

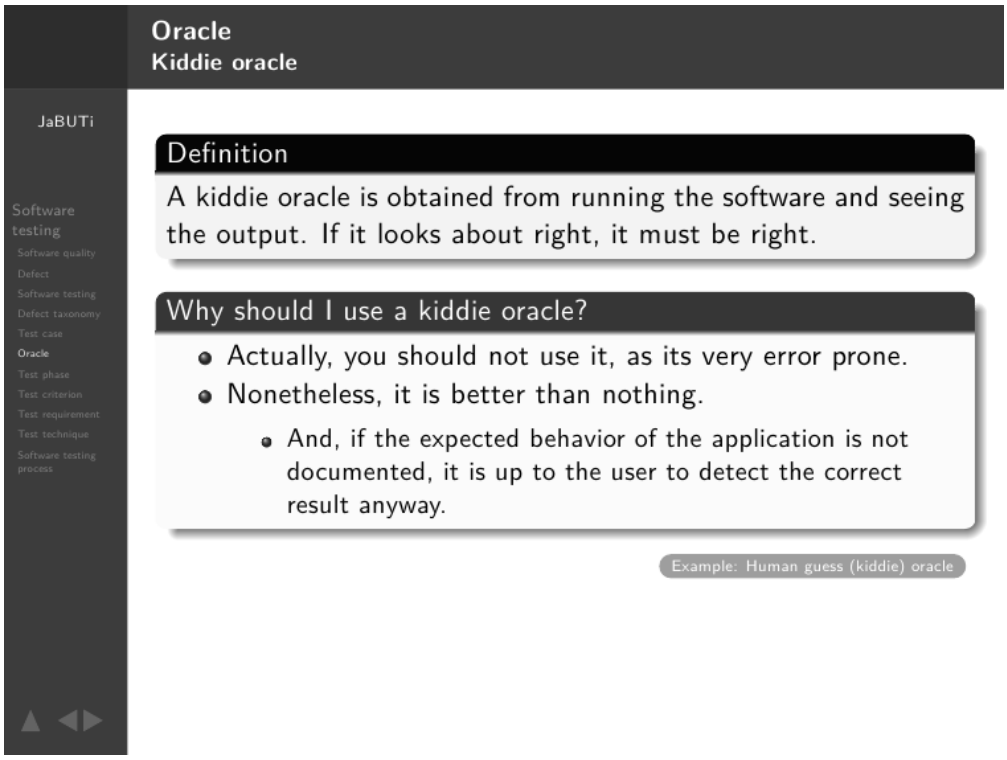


Figura 6.5: JaBUTi-TT: Quadro sobre um tipo de oráculo de teste.

Além de textos, conforme mencionado anteriormente, é possível colocar vídeos e animações na apresentação. No quadro 108 (Figura 6.6), sobre a visualização de requisitos de teste definidos em uma sessão de teste, é apresentado um bloco com o conceito na parte superior e outro na parte inferior com um elemento instrucional na forma de um vídeo para a demonstração da funcionalidade de definição dos requisitos de teste na ferramenta.

Dessa forma, foi possível construir, sistematicamente, um objeto de aprendizagem sobre a utilização da ferramenta de teste JaBUTi no contexto de teste de software. Destaca-se que, em alguns dos objetos de aprendizagem (*e.g.*, o de teste de software e teste estrutural de software) foram reutilizados modelos construídos para outros objetos de aprendizagem, demonstrando o potencial de reúso da abordagem.

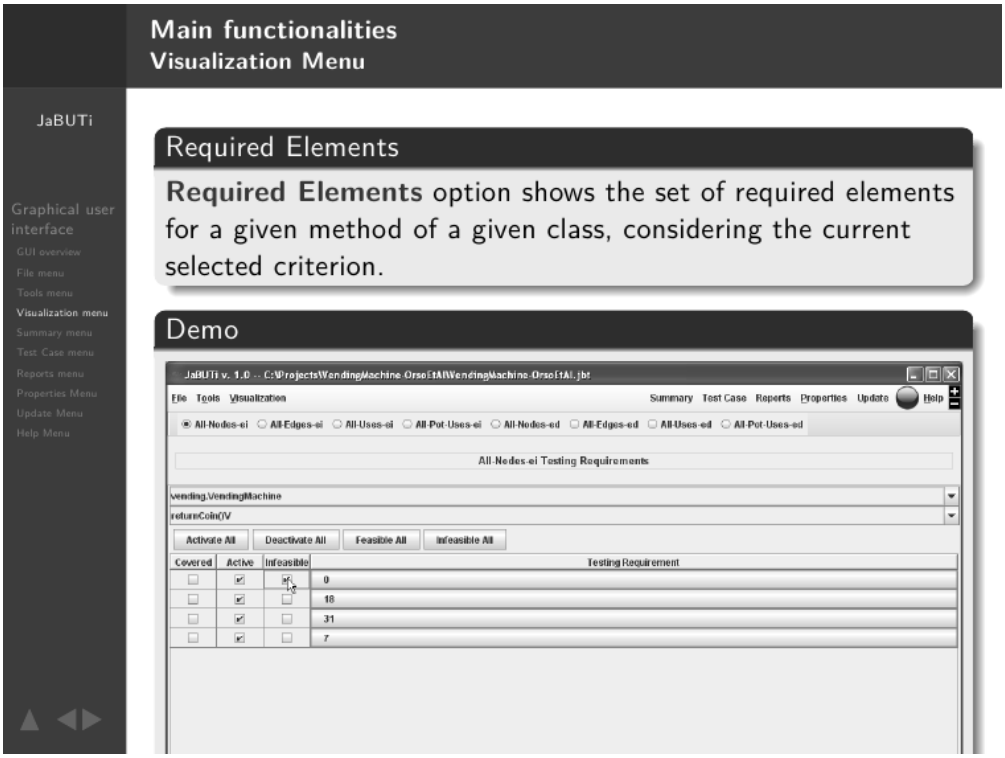


Figura 6.6: JaBUTi-TT: Quadro sobre a visualização de requisitos de teste.

Finalmente, é importante destacar que o código-fonte, no formato LaTeX, pode ser facilmente alterado pelo usuário. Por exemplo, o Código-fonte 6.1 contém o código utilizado para o quadro apresentado na Figura 6.6. Um quadro é definido da forma usual de uma apresentação em LaTeX/Beamer: a única diferença são os parâmetros que podem ser definidos (e vale destacar que o quadro atual reutiliza os valores das propriedades definidas no quadro anterior, o que reduz a quantidade de definições de propriedades). Os blocos para elementos instrucionais e itens de informação reutilizam o ambiente `block` do modelo definido pelo Beamer.

```
\begin{frame}
\frametitle{Main functionalities}
\framesubtitle{Visualization Menu}
\label{concept:required-elements}

\begin{block}{Required Elements}
\highlight{Required Elements} option shows the set of required elements for a
given method of a given class, considering the current selected criterion.
\end{block}

\begin{block}{Demo}
\includegraphics[width=\textwidth, clip]{JaBUTi-VendingMachine-TestCaseManagement}
\end{block}
\end{frame}
```

Código-fonte 6.1: JaBUTi-TT: Código do quadro sobre visualização de requisitos de teste

6.1.3. Análise

O objeto de aprendizagem foi analisado por especialistas do domínio de teste de software em duas iterações. Na primeira iteração, o objeto de aprendizagem continha somente elementos textuais, respeitando-se apenas a questão de navegação do modelo. Naquele instante, ainda não se adotava também padrões diferenciados para a representação dos elementos instrucionais e itens de informação. Os elementos multimídia eram poucos e definidos em estados que não eram aninhados ao estado que os chamavam, ou seja, era necessária uma transição entre o estado que continha os itens de informação para o estado com o elemento instrucional multimídia. Essa desvinculação tornava o entendimento do conceito difícil. Além disso, a pequena quantidade de elementos multimídia – vídeos e animações – foi criticada por tornar o material enfadonho.

A segunda iteração, a qual é apresentada como resultado desta seção, acrescentou elementos instrucionais explanativos na forma de vídeos e animações para a maioria dos conceitos representados em cada estado. Ao invés desses elementos instrucionais serem definidos em estados que estão aninhados, eles são definidos para a apresentação simultânea com os itens de informação do conceito. Em outras palavras, o texto e os vídeos ou animações que explicam o texto são apresentados no mesmo quadro. Essa característica tornou a apresentação mais aprazível.

Em relação ao desenvolvimento do objeto de aprendizagem, apresentações multimídia interoperáveis são de difícil implementação, principalmente se considerados sistemas operacionais distintos (Windows, Linux, MacOS). Embora existam visualizadores de documentos PDF que funcionem em diversas plataformas (*e.g.*, o Adobe Reader, da Acrobat), eles não implementam as mesmas funcionalidades, em especial quanto à reprodução de vídeos. Dessa forma, os modelos LaTeX utilizados para a criação da apresentação precisam contornar essas questões, sem tornar mais complexa a escrita da apresentação em LaTeX/Beamer. Embora o esforço para alterar os modelos em LaTeX seja elevado, o tempo economizado para a geração da apresentação em si é recompensador. De fato, uma vez criados os vídeos e animações, a criação da apresentação consumiu poucos dias.

6.2. Estudo 2 – Programa multimídia e interativo de televisão

Este estudo foi realizado no contexto da televisão digital interativa, no SBTVD, utilizando-se a plataforma Ginga. O objeto de aprendizagem destinou-se a um segmento de uma aula sobre teste de mutação, segmento no qual se trabalha o critério de análise de mutantes.

6.2.1. Planejamento

Este estudo deve avaliar o desenvolvimento de objetos de aprendizagem interativos para televisão digital utilizando a abordagem LOD e, mais especificamente, as questões relacionadas à interatividade e geração de aplicações interativas para TVD do LODM.

Ele consiste em um objeto de aprendizagem sobre teste de software com a técnica de análise de mutantes. Uma premissa para a execução de um objeto de aprendizagem na televisão digital é a existência de uma aplicação que o execute. Embora não seja o foco deste trabalho a definição dessa aplicação, é importante considerar a existência dessa e como ela interage com o conteúdo transmitido na televisão para tornar o exemplo completo.

Para todo programa televisivo transmitido pela televisão digital, um conjunto de metadados (descritores) são enviados pelas tabelas *Event Information Table* (EIT), dentre os quais está a descrição do evento. Parte dessa informação é acessível pela API JavaTV na forma de uma breve descrição do programa. Esses dados serão utilizados para buscar um objeto de aprendizagem interessante para os usuários da televisão a partir dos cursos em que eles estão matriculados em um ambiente virtual de aprendizagem. Imaginando um cenário, em que o receptor de TV possui o identificador do usuário e os ambientes de aprendizagem em que ele participa, uma aplicação, transmitida junto com a televisão, mantém-se em execução durante todo o tempo em que a televisão está ligada. Sempre que um novo programa de televisão é iniciado ou ocorre a mudança de canal (e, portanto, de programas), verifica-se a existência de objetos de aprendizagem, na forma de aplicações NCL, sendo transmitidos e que sejam de interesse dos usuários.

Para este estudo de caso, foi escolhido um programa de televisão sobre turismo espacial para simular o uso habitual de televisão. Uma aplicação Java, doravante denominada LODE-TV, transmitida antes, durante ou até mesmo residente no receptor de televisão, extrai o título e a descrição do programa. Ao obter as informações do usuário da televisão (telespectador), LODE-TV recupera as últimas atividades de aprendizagem realizadas a partir de um serviço Web disponibilizado pelo ambiente de aprendizagem virtual. Utilizando as informações das atividades e do programa de TVD, descobre-se (em um repositório de aprendizagem) um objeto de aprendizagem sobre o acidente ocorrido no lançamento inaugural do foguete Ariane 5, causado por um erro no software. Solicita-se então a recuperação do referido objeto, seja pelo canal de retorno (*e.g.*, pela Internet) ou pela transmissora (supondo que ela forneça um serviço que permita a solicitação de envio de aplicações sob demanda). Assim que o objeto de aprendizagem for inteiramente recuperado, LODE-TV avisa o usuário sobre a disponibilidade de uma aplicação interativa, apresentando um símbolo no canto superior esquerdo da tela e uma breve descrição em um quadro transparente no topo, sobrepondo o vídeo. Caso o usuário queira ativar a aplicação, ele deve acionar o botão de confirmação (`Ok`) ou o botão verde do controle remoto.

6.2.2. Resultados

O objeto de aprendizagem TestingAriane501 apresenta o caso do primeiro lançamento do foguete Ariane 5, em junho de 1996, que explodiu 37 segundos após o lançamento, causando um prejuízo de aproximadamente 100 milhões de dólares. Após uma investigação realizada pela *European Space Agency* (ESA), descobriu-se que a causa do acidente foi no software do sistema de referência inercial. Esse software foi reutilizado do antecessor do Ariane 4 (antecessor do Ariane 5), supondo-se, portanto, que ele estava correto e não precisava ser testado. No entanto, em um determinado trecho do código-fonte, existe uma conversão de variável para números inteiros sem sinal para com sinal que não observava a ocorrência de estouro do número (*integer overflow*). No Ariane 4, os valores da variável nunca assumiam valores altos os suficientes (acima de 32.767 ou abaixo de -32.768) para ativar o erro, mas, no Ariane 5, devido ao perfil de voo (aceleração muito mais rápida que o Ariane 4), essa condição é alcançada e o erro se manifesta. Esse erro causou a falha do sistema de referência inercial e alterou os dados de voo do foguete (mais precisamente dos foguetes sólidos presos ao motor principal), culminando no estresse da estrutura a partir dos 34 segundos, a ruptura dela no instante 37 e explosão.

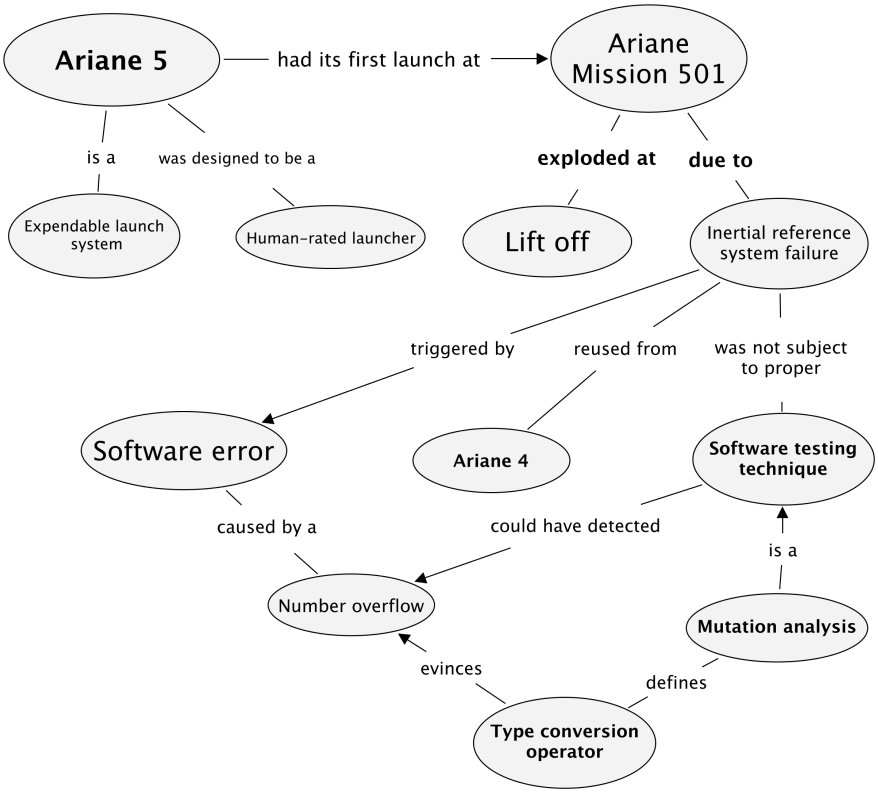


Figura 6.7: TestingAriane501: Modelo conceitual (LODM).

Os principais conceitos a serem observados nesse caso estão definidos no modelo conceitual da Figura 6.7. A relação com o programa de televisão apresentado dá-se pelo fato do Ariane

5, um lançador descartável de satélites, ter sido projetado também para o lançamento de veículos tripulados em direção às estações espaciais. Logo, a viagem de férias de uma família pode vir a ser realizada em foguetes similares a esse, que, como todo artefato humano, está sujeito a falhas. Felizmente, é possível identificar os erros que as causam com o devido emprego de técnicas, tal como aquelas de teste de software. No objeto de aprendizagem, em específico, a análise de mutantes é utilizada para testar uma unidade de código, que corresponde à que falhou no Ariane 5, com operadores para mutação da conversão de tipos de variáveis (ZENG *et al.*, 2009).

A partir do modelo conceitual apresentado na Figura 6.7, com o auxílio das diretivas definidas no LODM, foi criado o modelo instrucional para o objeto de aprendizagem. Para fins de gerenciabilidade do modelo, ele foi dividido em três modelos, aproveitando-se da possibilidade de descrever um estado composto em diagrama de *statechart* distinto. Na Figura 6.8 é apresentado o diagrama que representa os principais estados do modelo instrucional.

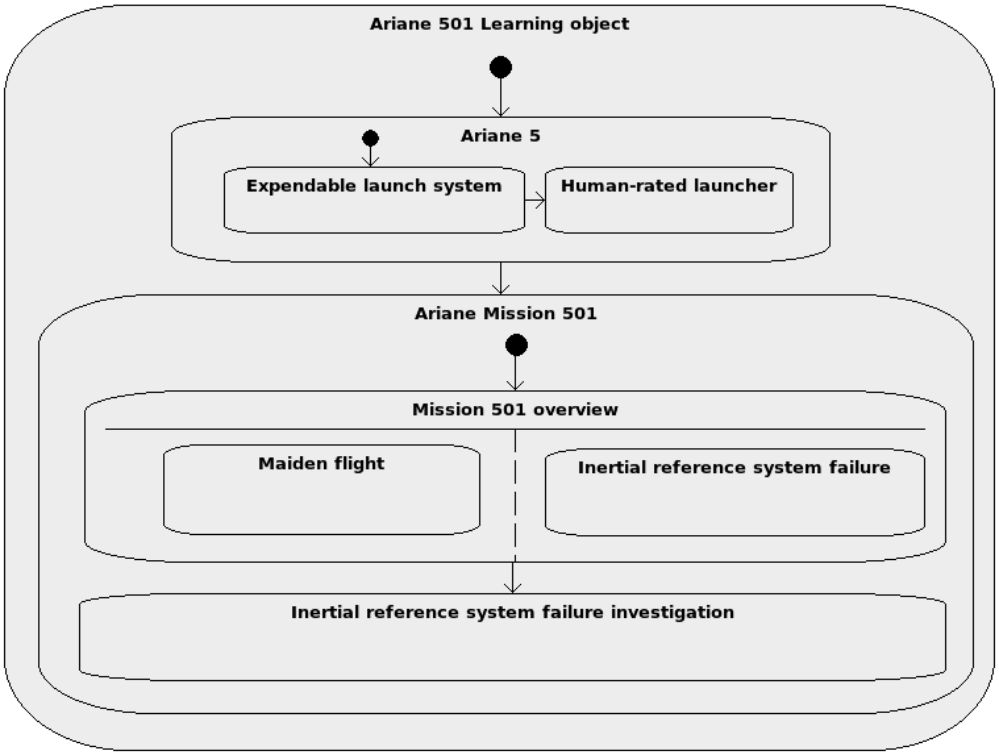


Figura 6.8: TestingAriane501: Modelo instrucional – primeira parte (LODM).

O objeto de aprendizagem inicia-se com os conceitos sobre o foguete Ariane 5 e a relação dele com o turismo espacial. Em seguida, trata-se da missão 501, o voo inaugural do foguete e que fracassou com a sua explosão. Segue-se com a investigação das causas do acidente, que é definida no diagrama da Figura 6.9. A falha que causou a explosão é mostrada em detalhes e em vários ângulos, concomitantemente com a descrição do erro que a causou. Desse estado segue-se para a explicação de como a técnica de teste de mutação poderia ter evitado tal

desastre e se aplica a técnica em um código similar ao que causou o erro, mostrando-se o código original, o operador de mutação e uma aplicação que interage com o usuário, gerando mutantes do código-fonte original com o operador de mutação de conversão de tipo e os executando com casos de teste.

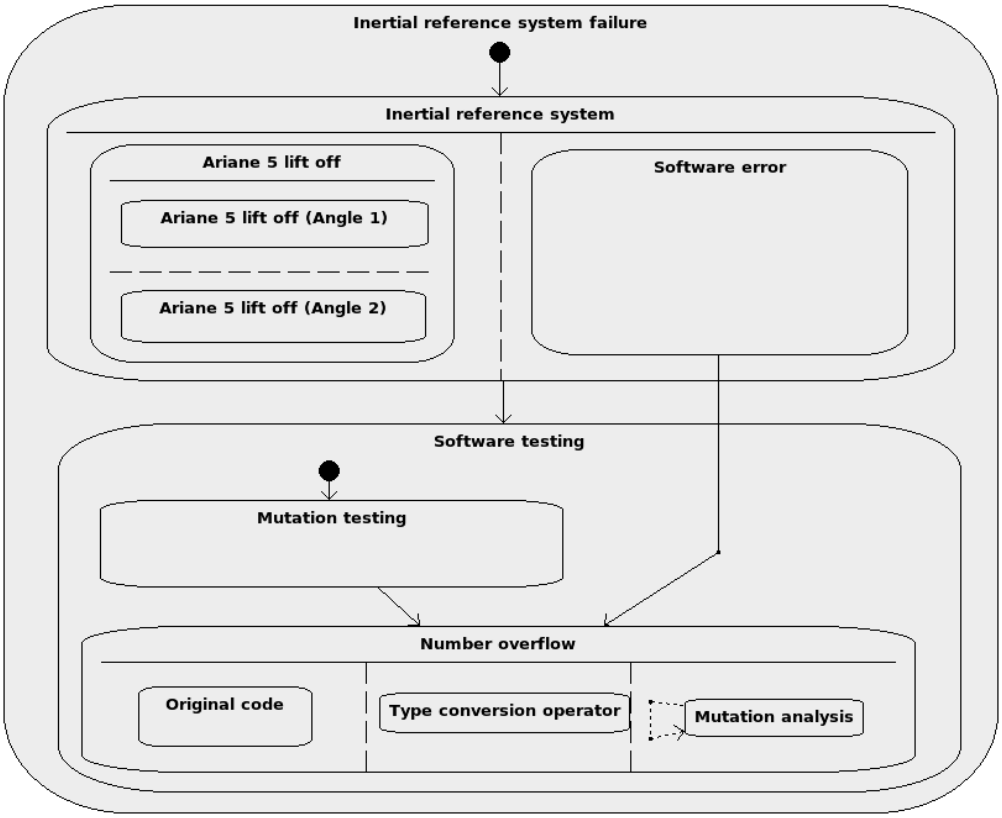


Figura 6.9: TestingAriane501: Modelo instrucional – segunda parte (LODM).

O modelo instrucional deste objeto de aprendizagem consiste da especificação dos estados XOR DD e a designação do esteriótipo *ExpiringStatechart* para a especificação. Para este objeto de aprendizagem, definiram-se como XOR DD apenas os estados `Inertial reference system` e `Software testing`, conforme apresentado na Figura 6.10.

A partir do modelo de interação, foi criado um documento NCL com a definição de todos os contextos, conforme as especificações previstas no LODE: definição de regiões, tradução de estados em contextos e definição de conexões para as transições.

6.2.3. Análise

Os resultados obtidos foram similares aos observados no primeiro estudo quanto à facilidade de desenvolvimento. As principais deficiências observadas foram quanto aos elementos multimídia e às formas restritas de interação. Um problema importante em vídeos educacionais (e cada segmento de vídeo utilizado como elemento de mídia nos contextos é um vídeo educacional)

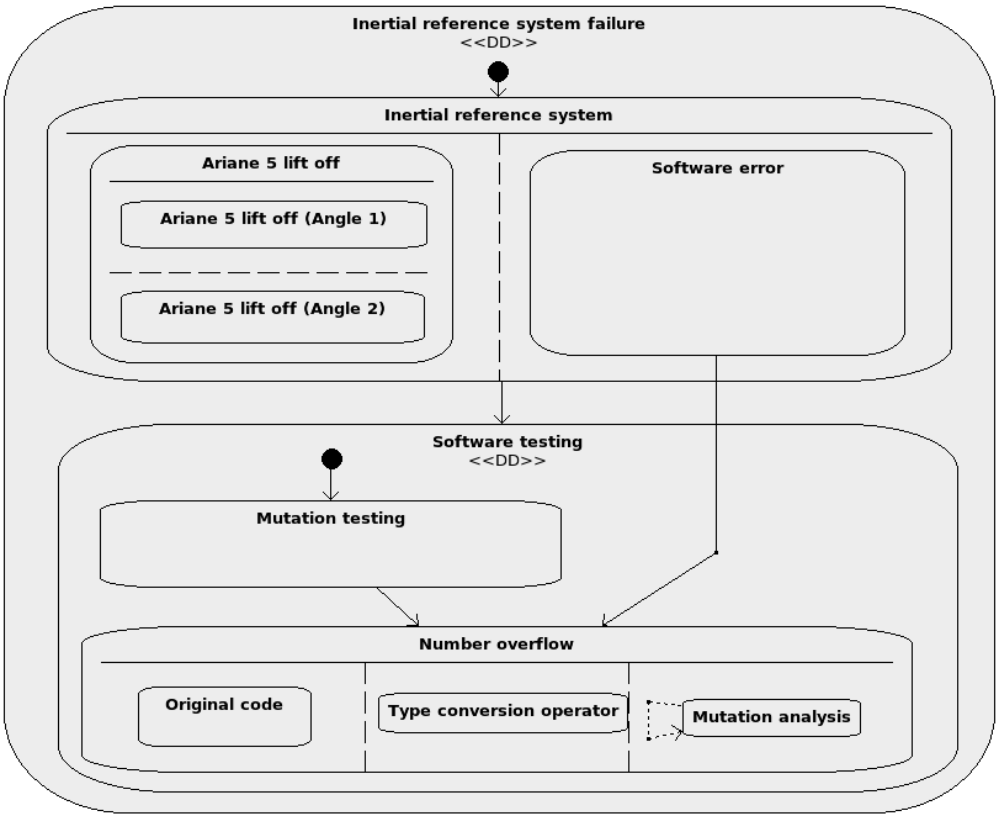


Figura 6.10: TestingAriane501: Modelo de interação (LODM).

é a dificuldade de produção (SILVA *et al.*, 2010). Essa questão é difícil de transpor no âmbito deste trabalho, sendo necessária a sua avaliação no contexto de uma organização com capacidade para a execução de serviços audiovisuais e atuante nos processos técnicos previstos no LODP.

Quanto às poucas oportunidades de interação, deve-se lembrar de que a televisão digital é, ao menos por enquanto, restrita quanto aos recursos de interação que suporta. Embora tenha potencial para integrar diversos dispositivos e formas ricas de interação, atualmente, com os ambientes de desenvolvimento para aplicações de TVD disponíveis, isso ainda não é possível. Não obstante, em futuras iterações de desenvolvimento deste objeto de aprendizagem, será considerado o acréscimo de mais atividades interativas, tal como aquela representada pela execução de teste de mutação com o critério análise de mutantes.

6.3. Análise da abordagem LOD

A partir da definição da abordagem LOD, compreendida pelo método LODM, o processo LODP e o ambiente de desenvolvimento LODE, é possível verificar, nas Seções 6.3.1 a 6.3.3, a adequação da LOD quanto aos requisitos definidos no Capítulo 4. Complementando essa análise, comparam-se abordagens similares à LOD baseadas em modelos e apresentadas na Seção 2.3.1.6 e trabalhos que tratam da geração de objetos para TVD.

6.3.1. Resultados em relação aos requisitos educacionais

Analisando-se os resultados obtidos quanto aos requisitos educacionais, os modelos conceitual e instrucional permitem a representação do conhecimento e o modelo de interação das atividades entre os usuários, representadas pelos eventos de conceitos (satisfação de objetivos de aprendizagem) e de ações que sincronizam os elementos do modelo e as atividades a eles associadas: explanação, exploração e de avaliação. As informações de contexto, tais como formalidade, acesso, modalidade, tipos de interação e de avaliação, são consideradas na definição dos modelos, mas sem a especificação de elementos específicos para elas. Entretanto, o modelo subjacente (*statecharts*) permite a criação de variáveis que, conjugadas com ações apropriadas, representem o contexto além da configuração da máquina quanto aos conceitos e atividades em execução.

Quanto ao processo instrucional, questões quanto à formalidade, acesso e modalidade são tratados pelo SPLOD e LODP. Os modelos do LODM, por permitirem a rastreabilidade dos objetivos de aprendizagem para as atividades e objetos de aprendizagem, propiciam a construção de material de qualidade e alinhado com o projeto pedagógico da organização, contribuindo para a credibilidade e respeito às demandas legais (principalmente aquelas da educação formal). Neste trabalho assume-se que os objetos serão utilizados à distância e com amplo acesso, mas nada impede a sua utilização em outros contextos. Em relação às interações, elas são definidas em função dos eventos aceitos pelo modelo de interação. Neste trabalho é considerada a utilização no ambiente de apresentações multimídia e de televisão digital, mas outras configurações podem ser tratadas, adicionando-se novos tipos de eventos e programando-se as ações das transições adequadamente. Quanto às avaliações, elas são representadas pelos elementos instrucionais com função avaliativa, sem especificar o instrumento de avaliação empregado. Trabalhos futuros podem tratar dessa questão associando objetivos de aprendizagem (os conceitos e proposições do modelo conceitual) a classificações de taxonomia e projetando elementos instrucionais de avaliação que comparem a configuração do modelo com os objetivos. Os resultados dessas avaliações, com efeito global no modelo de interação, poderiam então ser utilizados como condições para a execução de transições.

O modelo de projeto instrucional ADDIE é pareado com o modelo de desenvolvimento de objetos de aprendizagem SPLOD: o processo de análise do projeto instrucional é necessário para os processos de definição e análise de requisitos do SPLOD; o processo de projeto está associado aos processos de desenho de arquitetura e projeto detalhado; o processo de desenvolvimento é substituído pelos processos de construção, integração, verificação, integração e validação do SPLOD; o processo de implementação do ADDIE requer a execução dos processos de operação e de manutenção do SPLOD; finalmente, a avaliação da instrução no ADDIE permite a identificação de novos requisitos e alteração dos existentes, permitindo

o início de um novo ciclo de desenvolvimento.

6.3.2. Resultados em relação aos requisitos computacionais

O método LODM é um refinamento de uma abordagem anterior, a IMA-CID, que integra diversas perspectivas em modelos. Após a aplicação do último em vários objetos de aprendizagem sobre teste de software, observaram-se diversos elementos a serem aperfeiçoados, culminando em uma versão preliminar do método LODM. Uma das diferenças foi quanto à compreensão do que é exatamente um objeto de aprendizagem e o que se desejava modelar e reutilizar. Um objeto de aprendizagem pode consistir de simples recursos educacionais até complexas definições de atividades de aprendizagem, passando por recursos descritos por metadados. Considerar como um objeto de aprendizagem não apenas o resultado final da aplicação da abordagem (a aplicação de TVD ou a apresentação multimídia), mas também os modelos intermediários, representando o projeto instrucional, tal como se faz com uma linguagem de modelagem educacional, mostra-se como um importante fator para alavancar o reúso dos objetos de aprendizagem. O conceito de objeto de aprendizagem, na época da abordagem IMA-CID, era recente e, ao definir um módulo educacional, evitou-se reduzi-lo a um "simples" objeto de aprendizagem (na época visto como um recurso educacional com metadados). No método LODM, um módulo educacional é um objeto de aprendizagem, definido por seus modelos, conteúdo e metadados, definindo-se no tipo mais complexo de objeto de aprendizagem que pode ser definido e que ainda permite o reúso. Não obstante, outros objetos de aprendizagem – de qualquer tipo – podem ser utilizados como elementos instrucionais em um objeto definido com o LODM, desde recursos educacionais (figuras, vídeos, textos) a outros objetos definidos com LODM. Neste último caso, vale-se da possibilidade aninhar *statecharts* e unir modelos conceituais para construir objetos de aprendizagem cada vez mais complexos.

Em relação à representação, cada estado composto do modelo instrucional define uma atividade de aprendizagem e cada estado aninhado estabelece subatividades ou outros objetos de aprendizagem utilizados para a realização da atividade. Em última instância, os estados básicos representam os objetos de aprendizagem mais simples (recursos educacionais ou simples objetos com metadados) e cada estado composto pode ser visto como um objeto de aprendizagem em si. Como todo modelo de interação e instrucional está associado com o modelo conceitual, as principais informações do objeto sempre estão disponíveis, o que facilita a criação de pacotes para os objetos de aprendizagem, tal como preconizado pelos padrões estudados no Capítulo 2. Quanto à integração com os ambientes de execução, no contexto de aplicações educacionais orientadas a serviço, é possível representá-las como estados e as inserir no modelo de interação ou, se for o caso, considerar cada objeto como um estado definido na aplicação, repassando-se os eventos para ele e tratando os demais eventos gerados

pelo objeto de aprendizagem. Logo, é possível considerar o método LODM nesse cenário, embora ele não seja explorado neste trabalho.

Quanto às abordagens de desenvolvimento, o LODM/SPLOD, tal como o IMA-CID/SP-DEM diferenciam-se das demais soluções por modelarem não apenas o sequenciamento, mas também os conceitos e requisitos de aprendizagem, fazendo-o ainda de forma integrada. O SPLOD atualiza o SP-DEM quanto a novos padrões, mas ambos cumprem o mesmo papel. O processo definido a partir do SPLOD possui importantes elementos para o desenvolvimento de objetos de aprendizagem abertos, considerando o próprio processo como um elemento determinante da evolução e do reúso dos objetos de aprendizagem. Afinal, um processo de desenvolvimento de objetos de aprendizagem e recursos educacionais abertos, assim como de software livre, não trata apenas do objeto, mas da comunidade que o desenvolve. Assim, devem-se especificar as atividades necessárias para promover a integração de seus membros pela igualdade e liberdade de acesso aos artefatos, que, para os objetos de aprendizagem, inclui os modelos definidos pelo LODM. Apenas após o estabelecimento dessas premissas é que se devem definir os processos técnicos – atividades de construção, verificação, validação, dentre outras.

O método LODM define a integração entre modelos que representam diferentes interesses/aspectos dos objetos de aprendizagem. Dessa forma, estabelece-se um compromisso entre a formalidade e completeza daqueles métodos que definem precisamente as sequências das atividades e a abstração e fácil compreensão das linguagens de modelagem educacional. O conceito de desenvolvimento dirigido a modelos, observado em alguns métodos com foco na adaptação de conteúdos, é utilizado no contexto da integração dos modelos, em um aspecto que o LODM aperfeiçoa-se quanto à IMA-CID pela redefinição de alguns modelos, a representação deles por linguagens comuns como a UML (e, portanto, com amplo suporte de ferramentas e curva de aprendizado mais branda) acrescida de perfis específicos, a definição de transformações entre esses modelos e deles para a geração de artefatos "executáveis", tais como apresentações e aplicações para TVD.

6.3.3. Resultados em relação aos requisitos de interação

Em relação aos aspectos interativo e multimídia de objetos de aprendizagem, o método LODM considera a utilização de elementos multimídias e sincronização entre eles. *Statecharts* permitem a resposta a eventos gerados pelo sistema e pelo usuário e isso é considerado pelo modelo de interação do LODM. Além disso, contemplam-se algumas características específicas de televisão digital, seja pelo cuidado durante a geração dos recursos a serem utilizados na aplicação (quanto à apresentação e à estética, por exemplo) ou pela forma com que se interage (ou não interage) com a televisão, permitindo-se a execução automática de uma sequência de atividades caso o usuário não queira interagir com a aplicação, assumindo

o papel de espectador. Assim, atende-se tanto a interação quanto à não interação com a televisão digital, ambos cenários válidos e reais.

O processo LODP estabelece atividades de elicitação (engenharia de requisitos), projeto e implementação, permitindo o desenvolvimento de objetos de aprendizagem interativos de forma sistemática. Embora não seja uma garantia de qualidade, os processos e atividades estabelecem pontos para aferir a qualidade durante o desenvolvimento, contribuindo para processos centrados no usuário e iterativos.

6.3.4. Trabalhos relacionados

Celes e Souza (2007) definem gabaritos (*templates*) a partir de estilos arquiteturais que definem uma atividade de aprendizagem. Um estilo arquitetural define um vocabulário composto por elementos do projeto e um conjunto de conectores e restrições quanto à combinação dos elementos, permitindo o reúso de organizações arquiteturais utilizadas em problemas recorrentes (SHAW; GARLAN, 1996). A especificação de estilos arquiteturais, utilizando linguagens de descrição arquitetural, é utilizada para a geração de um gabarito NCL, restando apenas a definição de elementos de mídia (a sincronização/projeto instrucional já são definidos pelo estilo arquitetural).

O método LODM permite a geração de documentos NCL a partir de uma especificação formal do projeto instrucional. No entanto, ela não se restringe às definições estabelecidas no estilo arquitetural, permitindo a realização de projetos instrucionais diversificados. Além disso, a escolha de elementos de mídia é realizada no modelo, sem a necessidade de um passo posterior para escolhê-las, tal como na abordagem de Celes e Souza (2007). A associação com um modelo conceitual e, consequentemente, com os objetivos de aprendizagem proporcionam rastreabilidade quanto aos requisitos do objeto de aprendizagem, proporcionando uma visão integrada do objeto de aprendizagem.

Não obstante, o uso de estilos arquiteturais para representar as soluções típicas de projeto instrucional, de acordo com um método ou teoria educacional, é um tópico a ser investigado em trabalhos futuros. No contexto dessa tese, realizou-se uma revisão sistemática sobre padrões pedagógicos, que consistem de padrões de projeto para o contexto de ensino e aprendizagem (SILVA, 2007). Padrões pedagógicos e estilos arquiteturais proporcionam o reúso do projeto instrucional e, nesse contexto, poderiam ser utilizados para facilitar a definição de projetos instrucionais. A implementação dos gabaritos pode ser realizada tanto com a proposta de Celes e Souza (2007) quanto com o XTemplates (SANTOS; SAADE, 2011), que compartilham o mesmo conceito de reúso a partir do estilo arquitetural, porém contando com uma proposta mais completa e madura.

Considerando a abordagem de desenvolvimento baseada em modelos, Souza *et al.* (2010) definem linguagens específicas de domínio para a geração de objetos de aprendizagem

personalizados. No entanto, a abordagem deles é centrada em questões de interface gráfica. Embora tais aspectos sejam relevantes, eles não são suficientes para a definição de objetos de aprendizagem relevantes educacionalmente. O método LODM permite representar os elementos de mídia e de interface gráfica, definindo-se perfis adicionais, mas, nesta tese, explorou-se principalmente os elementos associados ao projeto instrucional.

Em relação ao trabalho de Dodero *et al.* (2007), ele está fortemente associado aos padrões IMS (CP, LD e QTI), definindo a abordagem MDLD, composta por transformações que permitem a geração de objetos de aprendizagem em conformidade com os referidos padrões. O LODM, no atual estágio, não considera a geração de objetos em conformidade com esses padrões, embora seja possível a geração automática dos metadados a partir dos elementos definidos nos modelos. Nesse quesito, a integração entre os modelos do LODM, considerando desde o modelo conceitual e a sua geração a partir de outros modelos ou materiais didáticos, configura uma vantagem da abordagem em relação ao MDLD.

6.4. Considerações finais

Os objetos de aprendizagem gerados permitem avaliar a viabilidade da técnica para o desenvolvimento sistemático de objetos de aprendizagem. Foram considerados tanto aspectos multimídia quanto de interação com o usuário, com ênfase na multimídia no primeiro estudo e em interação no segundo. Observa-se que a técnica direciona as atividades, provendo diretivas para a criação de modelos e mecanismos sucintos para a representação de alguns aspectos de multimídia e interação.

Entretanto, destaca-se que a interatividade é ainda um elemento a ser melhor elaborado no âmbito da televisão digital, integrando-se a modelagem dos documentos, visando à programação declarativa do objeto de aprendizagem, com o desenvolvimento de programas em linguagens imperativas (tal como Java e Lua) e a integração entre esses dois ambientes.

Em relação aos demais trabalhos relacionados, a integração entre os modelos conceitual, instrucional e de interação do LODM permite o estabelecimento de um conjunto rico de transformações. Embora no estado atual eles se restrinjam à geração de objetos de aprendizagem no formado de documentos NCL e apresentações multimídia, a representação em outros formatos e em conformidade com padrões estabelecidos no mercado é passível de realização a partir dos dados modelados ou a partir da definição de perfis adicionais para os modelos do LODM.

A definição de um modelo de referência (SPLOD) para a elaboração de processos de desenvolvimento de objetos de aprendizagem também contribui para o estabelecimento de uma cultura de objetos de aprendizagem de qualidade. Em geral, as abordagens ignoram os processos ou os consideram de forma simplificada. Embora seja possível adotar processos

simples para organizações com pouca maturidade, a complexidade do desenvolvimento de objetos de aprendizagem sugere a necessidade por processos elaborados e bem definidos. O modelo de referência contribui para esse propósito ao definir esses elementos e associá-los a padrões reconhecidos para esse fim. O processo definido a partir do SPLOD, o LODP é um exemplo nesse contexto, estabelecendo um processo para o desenvolvimento de objetos de aprendizagem abertos, com vistas às características de interatividade em televisão digital em um contexto social, com alteração e compartilhamento de conteúdos.

Capítulo 7

Conclusões

O desenvolvimento de objetos de aprendizagem multimídias e interativos requer a análise das perspectivas educacionais e computacionais envolvidas, com atenção à interatividade das atividades e das plataformas de execução: a televisão digital. Os objetivos deste trabalho estabelecem que métodos de modelagem adequados devem ser utilizados, integrando essas diversas perspectivas e com o auxílio de ferramentas para automatizar as tarefas. Além disso, deve-se considerar não apenas a questão técnica do desenvolvimento, mas os demais processos associados. A definição de processos-padrão que estabeleçam os processos e atividades e que permita a definição de processos adequados ao perfil dos objetos de aprendizagem para televisão digital também constituiu um dos objetivos desta pesquisa.

Este trabalho oferece uma visão ampla sobre o desenvolvimento de objetos de aprendizagem, compreendendo processos e métodos baseados em diversas técnicas: hipertexto, linguagens de modelagem educacional e modelos formais. Em consonância com tendências educacionais, tal como televisão digital e suas características multimídia e interativa, desenvolveu-se a abordagem LOD, considerando requisitos educacionais, computacionais e de interação.

Conforme estabelecido pelo método de pesquisa escolhido para este trabalho, as atividades deste doutorado compreenderam a primeira etapa, de definição inicial da tecnologia, e a primeira fase da segunda etapa, que corresponde aos estudos de viabilidade da etapa de

refinamento da tecnologia. Na primeira etapa, foram realizadas (1) revisões bibliográficas para a formação da base teórica sobre os métodos para desenvolvimento de objetos de aprendizagem e das características intrínsecas das plataformas multimídia e interativas a serem consideradas na construção dos objetos de aprendizagem, e (2) revisões sistemáticas sobre fatores críticos para educação (SILVA *et al.*, 2011e) e sobre a interatividade em objetos de aprendizagem (SILVA *et al.*, 2011c), abrangendo as formas de interação, os métodos de desenvolvimento e a avaliação da qualidade da interação dos objetos de aprendizagem identificados. Com base nas evidências obtidas dessas revisões, definiram-se os requisitos para processos e métodos de desenvolvimento de objetos de aprendizagem multimídia e interativos para televisão digital, apresentados no Capítulo 4.

A proposta inicial da tecnologia consistiu na elaboração de uma abordagem de desenvolvimento de objetos de aprendizagem, seguindo os moldes de um processo de desenvolvimento de software livre. A abordagem é composta pelo método LODM e do processo LODP, definidos a partir dos requisitos identificados nas perspectivas educacional, computacional e de interação. Foram desenvolvidos diversos objetos de aprendizagem sobre teste de software na forma de apresentações multimídia e interativas (com foco na navegação), inicialmente com a técnica IMA-CID (BARBOSA; MALDONADO, 2011b), estabelecendo-se diretivas e técnicas para auxiliar a construção dos modelos (BARBOSA; SILVA, 2009; MACHADO *et al.*, 2011), o que permitiu identificar pontos críticos da viabilidade da técnica e a incorporação de melhorias. O emprego de uma abordagem orientada a modelos, auxiliada com extração de dados, demonstrou-se promissora para o desenvolvimento sistemático de objetos de aprendizagem nesse contexto (SILVA *et al.*, 2011a), portanto estabelecendo-se como uma contribuição do presente trabalho.

Considerando ainda os requisitos estabelecidos nesta tese, as diretivas elaboradas e a experiência adquirida com a abordagem dirigida a modelos durante o desenvolvimento de objetos de aprendizagem, definiu-se o processo-padrão SPLOD – estabelecido a partir das normas ISO 15288:2008 e 12207:2008 e do processo-padrão SP-DEM (BARBOSA, 2004; BARBOSA; MALDONADO, 2008). Desse modelo de processo, definiu-se o processo LODP com a finalidade de auxiliar a condução do desenvolvimento de objetos de aprendizagem multimídias e interativos em uma concepção aberta (recursos educacionais livres).

O método LODM, nesse contexto, pode ser utilizado para os processos técnicos, constituindo uma solução integrada de desenvolvimento. Esse método contempla o desenvolvimento de objetos de aprendizagem, em especial daqueles voltados para televisão digital, satisfazendo requisitos quanto aos aspectos multimídia e interativo. O protótipo de uma ferramenta para auxiliar nas atividades de transformação dos modelos, denominada LODE, também foi desenvolvido, colaborando significativamente para a viabilidade da aplicação do método LODM e da abordagem LOD.

A segunda etapa do método de pesquisa corresponde ao refinamento da tecnologia. Neste trabalho, executou-se a primeira fase: estudo de viabilidade. Utilizando a abordagem LOD, foram criados dois objetos de aprendizagem, conforme apresentados no Capítulo 6. O primeiro, sobre teste de software com o auxílio da ferramenta JaBUTi, explorou a questão multimídia e valida o conceito de navegação proposto no modelo. O segundo, também sobre teste de software, tratou da interatividade de modo geral em um cenário típico para um objeto de aprendizagem em televisão digital.

Os resultados obtidos pela aplicação da abordagem evidenciam a importância da modelagem e do processo para o desenvolvimento de objetos de aprendizagem de qualidade. A qualidade é definida por construção ao se definir métodos, processos e ferramentas para auxiliar a execução de atividades necessárias para a engenharia de objetos de aprendizagem. Além disso, a qualidade também é resultado do esforço economizado em atividades de definição de projeto instrucional, modelagem e geração dos objetos a partir dos modelos, permitindo ao usuário dedicar-se aos elementos relevantes do objetos de aprendizagem e das características das atividades de aprendizagem.

Considera-se assim que os objetivos estabelecidos para esta tese foram alcançados: determinou-se um conjunto de requisitos considerando as perspectivas educacionais, computacionais e de interação, que serve de subsídio para a criação de abordagens de desenvolvimento de objetos de aprendizagem; definiu-se o método LODM para o desenvolvimento de objetos de aprendizagem e um protótipo do ambiente LODE para auxiliar sua aplicação; estabeleceu-se o modelo de processo SPLOD, considerando versões atualizadas nas principais normas ISO relacionadas e em acordo com os processos e atividades do desenvolvimento de objetos de aprendizagem; delineou-se o processo LODP a partir do SPLOD, contemplando o desenvolvimento de objetos de aprendizagem multimídias, interativos e abertos.

7.1. Publicações

Durante o doutorado, foram produzidas as seguintes publicações relacionadas a este trabalho em eventos arbitrados:

- **Graciotto Silva, M. A.**, Barbosa, E. F., Maldonado, J. C. *Model-driven development of learning objects*. In: Frontiers in Education Conference (FIE 2011), Rapid City, SD, EUA, 2011.
  Este artigo define o método LODM, parte da abordagem LOD, compreendendo uma das principais contribuições deste trabalho.
- **Graciotto Silva, M. A.**, Borges, Borges, V. A., Barbosa, E. F., Maldonado, J. C. *Novas Tendências em Educação de Engenharia de Software: um Estudo de Caso no Domínio de Teste de Software*. In: Fórum de Educação em Engenharia de Software

(FEES 2008), Campinas, SP, Brasil, 2008.
Este trabalho apresenta elementos de motivação para o desenvolvimento de objetos de aprendizagem interativos, em especial para televisão digital e no domínio de teste de software, conforme explorados nesta tese.

- Machado, J. B., **Graciotto Silva, M. A.**, Maldonado, J. C., Barbosa, E. F. *Reengineering of Educational Content: an Experience in the Computer Networks Domain*. In: Simpósio Brasileiro de Informática na Educação (SBIE 2011). Aracaju, SE, Brasil, 2011.
  Trata da aplicação da técnica IMA-CID na reengenharia de conteúdos na elaboração de objetos de aprendizagem.
- Barbosa, E. F., **Graciotto Silva, M. A.**, Corte, Camila K. D., Maldonado, J. C., *Integrated Teaching of Programming Foundations and Software Testing*. In: Frontiers in Education Conference (FIE 2008), Saratoga Springs, NY, EUA, 2008.
  Trata da aplicação da técnica IMA-CID na produção de objetos e ferramentas de aprendizagem no domínio de teste de software.

Além desses trabalhos, uma série de relatórios técnicos foi elaborada, tratando de aspectos específicos do desenvolvimento de objetos de aprendizagem e das revisões sistemáticas conduzidas durante o doutorado:

- **Graciotto Silva, M. A.**, Barbosa, E. F., Maldonado, J. C. *Educação a Distância na PUCRS*. São Carlos, SP, Brasil, 2010.
- **Graciotto Silva, M. A.**, Barbosa, E. F., Maldonado, J. C. *Movie support for PDF using LaTeX*. São Carlos, SP, Brasil, 2011.
- **Graciotto Slva, M. A.**, Barbosa, E. F., Maldonado, J. C. *Revisão sistemática sobre interatividade em objetos de aprendizagem*. São Carlos, SP, Brasil, 2011.
- **Graciotto Silva, M. A.**, Barbosa, E. F., Maldonado, J. C. *Screencast recording*. São Carlos, SP, Brasil, 2011.
- **Graciotto Silva, M. A.**, Barbosa, E. F., Maldonado, J. C. *Systematic review on critical factors for distance learning and e-learning*. São Carlos, SP, Brasil, 2011.

7.2. Limitações

O desenvolvimento de objetos de aprendizagem multimídias e interativos é um processo interdisciplinar e de lenta execução. Embora abordagens orientadas a modelo, como a LOD, atribuam mais agilidade ao processo, apenas após o efetivo uso do objeto de aprendizagem e a avaliação dos aprendizes é que um novo ciclo de desenvolvimento pode se configurar. Observando o método escolhido para esta tese, dever-se-ia, nas próximas etapas de avaliação, considerar as iterações e o envolvimento dos diversos atores do processo – desenvolvedores,

aprendizes e professores – para avaliar com precisão a abordagem e estabelecer medidas corretivas, adaptativas, perfectivas e preventivas.

O desenvolvimento de aplicações interativas está geralmente associado a projetos centrados no usuário e em avaliações de usabilidade. Embora o processo-padrão contenha atividades de validação, o método não considera técnicas dessa natureza. Também se deve considerar a execução de outras atividades para avaliação da qualidade dos objetos de aprendizagem produzidos, devidamente associadas a modelos de qualidade de produto e de processo.

O método LODM, embora tenha considerado diversas teorias educacionais e faça uso de perfis UML para permitir a utilização de outros métodos instrucionais e respetivos elementos instrucionais e itens de informação, foi utilizado principalmente no contexto de aulas no ensino superior e treinamento. A utilização com outras formas de ensino (por exemplo, baseado em casos e projetos) provavelmente exercitará não apenas a extensibilidade do método pela definição de novos perfis, mas também a organização das atividades em estados aninhados. Embora se acredite que a atual abordagem seja capaz de representar tais projetos instrucionais, resta avaliar se o esforço necessário é razoável e se o modelo resultante é satisfatório para entendimento pelos atores envolvidos, em especial àqueles não-técnicos em computação.

Atividades de avaliação são consideradas apenas como elementos instrucionais, sem um tratamento especial. Não era o objetivo deste trabalho, mas se entende que avaliações diagnósticas, formativas e sumativas são importantes para a utilização de objetos de aprendizagem. Em geral, as avaliações estão associadas aos objetivos de aprendizagem, que estão definidos no modelo conceitual. Assim, poder-se-ia definir um modelo adicional para associar, aos conceitos e proposições do modelo conceitual, a classificação taxonômica a ser alcançada quanto aos aspectos cognitivo, afetivo e psicomotor e como cada estado pode satisfazer tais requisitos. Utilizando-se dos mecanismos de propagação de eventos e variáveis nos estados, atividades de avaliação poderiam ser adequadamente representadas.

Quanto ao ambiente de desenvolvimento, atualmente ele auxilia a execução das transformações, mas não oferece uma visão integrada das atividades suportadas ou uma interface amigável ao usuário. Ainda quanto às transformações, elas precisam ser aperfeiçoadas, permitindo a reutilização de padrões de projeto (tal como padrões pedagógicos) e a geração personalizada, de acordo com o perfil de cada usuário.

7.3. Trabalhos futuros

Esta tese permite a execução de pesquisas relacionadas a diversos panoramas de objetos de aprendizagem. No entanto, como trabalhos futuros, definem-se aqueles que mitigam as limitações apresentadas na seção anterior.

- Validação: Em continuação ao método para transferência e adoção de novas tecnologias na indústria, apresentado no Capítulo 1, pretende-se executar as próximas fases da etapa de refinamento de tecnologia, visando ao amplo uso da abordagem em centro de ensino e treinamento. Para esses estudos, deve-se considerar as perspectivas do aluno, do professor e a própria qualidade intrínseca do objeto de aprendizagem. A execução dos estudos experimentais pode ser realizada no contexto da academia e, posteriormente, de centros de treinamento, contemplando as necessidades de grupos locais ou distantes, nas modalidades presencial, híbrida e à distância.
- Avaliação: A definição de um modelo de avaliação do aprendiz, que entrecorte os modelos conceituais e instrucionais e defina objetivos educacionais claros quanto às classificações taxonômicas, contribuiria para a execução das atividades de aprendizagem e para o próprio desenvolvimento do objeto de aprendizagem.
- Processo: O processo de referência SPLOD definido nesta tese define os processos e atividades necessários à engenharia de objetos de aprendizagem. No processo LODP, define-se um processo simplificado, a partir do modelo estabelecido no SPLOD, para o domínio de objetos de aprendizagem multimídia e interativos desenvolvidos de forma aberta. No entanto, ainda é possível especializar o processo considerando a maturidade da organização ou comunidade envolvida, acrescentando-se e substituindo-se atividades por outras que agreguem mais qualidade aos objetos de aprendizagem e à própria vivacidade da comunidade.

  Seguindo a estratégia MDD adotada para o método LODM, pode-se realizar a especialização do SPLOD ou do LODP a partir da maturidade da organização e das características dos objetos de aprendizagem em desenvolvimento, tal como se propõe no contexto de engenharia de software (ALEGRíA *et al.*, 2011). Atualmente o SPLOD e o LODP estão descritos em um modelo UML, representando-se (1) os processos por interfaces no SPLOD e por classes no LODP e (2) as atividades por métodos nas interfaces e classes associadas aos respectivos processos. Atualmente existe o padrão *Software Process Engineering Metamodel* (SPEM) para a descrição de processos (OMG, 2008), que define um perfil UML. A partir de modelos definidos em UML com esse perfil, seria possível gerar modelos especializados, empregando-se mecanismos semelhantes àqueles utilizados nas transformações de modelos da LODM.
- Ambiente de desenvolvimento: O protótipo definido neste trabalho possui limitações quanto à interatividade. Em trabalho correlato, definiu-se um ambiente Web para o desenvolvimento de objetos de aprendizagem com a IMA-CID (BORGES *et al.*, 2011). Considerando os resultados desse trabalho e sob a perspectiva de desenvolvimento aberto de objetos de aprendizagem, com forte enfoque colaborativo, planeja-se a criação de um *forge* para objetos de aprendizagem, tal como os gerenciadores de projetos de software livre existentes, e integrá-los em ambientes de ensino a distância, permitindo o

oferecimento ágil de atividades de aprendizagem, em uma proposta aperfeiçoada de objetos de aprendizagem abertos.

- Qualidade: Considerando-se os modelos formais utilizados no LODM, é possível realizar o teste dos modelos gerados aplicando-se critérios de teste baseados na estrutura e em erros. Nesse contexto, técnicas e ferramentas da família Proteum (MALDONADO *et al.*, 2000), tal como a Proteum-RS/ST (SUGETA, 1999), podem ser aplicados. Quanto à qualidade do objeto de aprendizagem no aspecto educacional, pode-se observar a estética, alinhamento pedagógico, grau de apropriação ao contexto educacional, dentre outros elementos (LEARY *et al.*, 2011; CECHINEL *et al.*, 2011).

Referências

AARRENIEMI-JOKIPELTO, Päivi. Interactive learning environment in digital TV: Results and experiences. In: NALL, Janice; ROBSON, Robby (Ed.). *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*. Washington, DC, EUA: AACE, 2004. p. 1602–1609. ISBN 1-880094-54-1. Disponível em: `http://www.editlib.org/p/11485`.

AARRENIEMI-JOKIPELTO, Paivi. *Modelling and content production of distance learning concept for interactive digital television*. 200 p. Tese (Doutorado) — Helsinki University of Technology, Espoo, Finlândia, dez. 2006. Orientador: Juha Tuominen.

ABNT. *ABNT NBR 15601:2007 Versão Corrigida:2008 – Televisão digital terrestre - Sistema de transmissão*. Rio de Janeiro, RJ, Brasil, abr. 2008. 57 p. Disponível em: `http://www.abntcolecao.com.br/norma.aspx?ID=1606`.

ABNT. *ABNT NBR 15602-3:2007 Versão Corrigida:2008 – Televisão digital terrestre – Codificação de vídeo, áudio e multiplexação – Parte 3: Sistemas de multiplexação de sinais*. Rio de Janeiro, RJ, Brasil, abr. 2008. 17 p. Disponível em: `http://www.abntcolecao.com.br/norma.aspx?ID=1617`.

ABNT. *ABNT NBR 15606-4:2010 – Televisão digital terrestre – Codificação de dados e especificações de transmissão para radiodifusão digital – Parte 4: Ginga-J – Ambiente para a*

execução de aplicações procedurais. Rio de Janeiro, RJ, Brasil, abr. 2010. 91 p. Disponível em: http://www.abntcolecao.com.br/norma.aspx?ID=58131.

ABNT. *ABNT NBR 15606-6:2011 – Televisão digital terrestre – Codificação de dados e especificações de transmissão para radiodifusão digital – Parte 6: Java DTV 1.3*. Rio de Janeiro, RJ, Brasil, jul. 2010. 860 p. Disponível em: http://www.abntcolecao.com.br/norma.aspx?ID=62080.

ABNT. *ABNT NBR 15606-2:2011 – Televisão digital terrestre – Codificação de dados e especificações de transmissão para radiofusão digital – Parte 2: Ginga-NCL para receptores fixos e móveis – Linguagem de aplicação XML para codificação de aplicações*. Rio de Janeiro, RJ, Brasil, maio 2011. 299 p. Disponível em: http://www.abntcolecao.com.br/norma.aspx?ID=87219.

ABNT. *ABNT NBR 15606-8:2011 – Televisão digital terrestre – Codificação de dados e especificações de transmissão para radiodifusão digital – Parte 8: Ginga-J – Diretrizes operacionais para a ABNT NBR 15606-4*. Rio de Janeiro, RJ, Brasil, out. 2011. 47 p. Disponível em: http://www.abntcolecao.com.br/norma.aspx?ID=88638.

ADL. *SCORM 2004*. ago. 2009. Padrão. Disponível em: http://www.adlnet.gov/scorm/. Acesso em: 20 fev 2012.

ADL; AICC. *Project Tin Can*. jan. 2011. Projeto de Pesquisa. Disponível em: http://scorm.com/tincan/.

AEBIL, Hans. *Práticas de ensino: Formas fundamentais de ensino elementar, médio e superior*. 11. ed. São Paulo, SP, Brasil: EPU (Editora Pedagógica e Universitária), 1982. 387 p. Traduzido do original "Grundformen des Lehrens" por Edwino Aluysius Royer. ISBN 85-12-30250-x.

AICC. *AICC/CMI Compliance – Web-Based Computer-Managed Instruction*. set. 1998. Padrão. Disponível em: http://www.aicc.org/docs/AGRs/agr010v1.pdf. Acesso em: 20 fev 2012.

ALBRIGHT, Paul. *Open Educational Resources Final Forum Report*. Paris, França, dez. 2005. Internet Discussion Forum: Open Educational Resources - Open Content for Higher Education. Disponível em: http://www.unesco.org/iiep/virtualuniversity/media/forum/oer_forum_final_report.pdf. Acesso em: 20 fev 2012.

ALEGRíA, Julio A. Hurtado; BASTARRICA, María Cecilia; QUISPE, Alcides; OCHOA, Sergio F. An MDE approach to software process tailoring. In: *2011 International Conference on Software and Systems Process*. Honolulu, HI, United States: ACM, 2011. (ICSSP), p. 43–52. ISBN 978-1-4503-0730-7.

ALLERT, Heidrun; DHRAIEF, Hadhami; NEJDL, Wolfgang. *How are Learning Objects Used in Learning Processes? Instructional Roles of Learning Objects in LOM*. dez. 2001. Artigo online. Learning Lab Lower Saxony, University of Hanover. Disponível em: `http://www.kbs.uni-hannover.de/Arbeiten/Publikationen/2002/Modellierung-LOM.pdf`.

Altova. *UModel*. 2005. Programa de computador. Disponível em: `http://www.altova.com/umodel.html`.

AMARAL, Lucas L. do; GOMES, Thiago de A.; SOUZA, Maria de Fátima C. de; FILHO, José Aires de Castro; PEQUENO, Mauro C. Um aprimoramento do modelo de processo de criação de objetos de aprendizagem do projeto rived. In: *XXVI Congresso da SBC - XII Workshop de Informática na Escola*. Campo Grande, MS, Brasil: SBC, 2006. p. 373–376.

Apache. *OpenNLP*. 2010. Programa de computador. Disponível em: `http://incubator.apache.org/opennlp`.

ARAúJO, Rafael Viana Lopes; CARNEIRO, Caio Alves; SOUSA, Robson Cavalcante; GOMES, Fábio de Jesus Lima. Um objeto de aprendizagem na TV digital para o ensino de astronomia. In: *III Escola Regional de Computação Ceará-Maranhão-Piauí (ERCEMAPI 2009)*. Parnaíba, PI, Brasil: SBC, 2009. p. 1–6. Disponível em: `http://www.ufpi.br/subsiteFiles/ercemapi/arquivos/files/artigos/graduacao/g6.pdf`.

ATKINS, Daniel E.; BROWN, John Seely; HAMMOND, Allen L. *A Review of the Open Educational Resources (OER) Movement: Achievements, Challenges, and New Opportunities*. [S.l.], fev. 2007. 80 p. Disponível em: `http://www.hewlett.org/uploads/files/ReviewoftheOERMovement.pdf`.

ATKINSON, Richard Chatham; WILSON, Harlalee Allen. Computer-assisted instruction. *Science*, AAAS, United States, v. 162, n. 3849, p. 73–77, out. 1968. ISSN 0036-8075, 1095-9203.

ATSC. *A/52:2010: Digital Audio Compression (AC-3) Standard*. EUA: [s.n.], nov. 2010. Padrão. Disponível em: `http://www.atsc.org/cms/index.php/standards/published-standards/48-atsc-a52-standard`. Acesso em: 20 fev 2012.

AUSUBEL, David Paul. *Educational Psychology: a Cognitive View*. 1. ed. New York, NY, EUA: Holt, Rinehart and Winston, 1968. 685 p.

BARBOSA, Ellen Francine. *Uma Contribuição ao Processo de Desenvolvimento e Modelagem de Módulos Educacionais*. 253 p. Tese (Doutorado) — Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo, São Carlos, SP, Brasil, mar. 2004. Orientador: José Carlos Maldonado. Disponível em: `http://www.teses.usp.br/teses/disponiveis/55/55134/tde-24062008-164700/`.

BARBOSA, E. F.; MALDONADO, José Carlos. An integrated content modeling approach for educational modules. In: KUMAR, Deepak; TURNER, Joe (Ed.). *IFIP International Conference on Education for the 21st Century - 19th IFIP World Computer Congress (WCC 2006)*. Santiago, Chile: Springer Boston, 2006. v. 210, p. 17–26.

BARBOSA, E. F.; MALDONADO, J. C. Towards the establishment of a standard process for developing educational modules. In: *36th Annual Frontiers in Education Conference (FIE 2006)*. San Diego, CA, EUA: IEEE, 2006. p. 5–10. CD-ROM.

BARBOSA, Ellen Francine; MALDONADO, José Carlos. Specialization and instantiation aspects of a standard process for developing educational modules. In: MARGARIA, T.; STEFFEN, B. (Ed.). *International Symposium on Leveraging Applications of Formal Methods, Verification and Validation (ISoLA 2008)*. Porto Sani, Grécia: Springer Berlin Heidelberg, 2008. (Communications in Computer and Information Science, v. 17), p. 503–518.

BARBOSA, E. F.; MALDONADO, J. C. Collaborative development of educational modules: a need for lifelong learning. In: MAGOULAS, G. D. (Ed.). *E-Infrastructures and Technologies for Lifelong Learning: Next Generation Environments*. London, UK: IGI Global, 2011. p. 175–211.

BARBOSA, Ellen Francine; MALDONADO, José Carlos. IMA-CID: an integrated modeling approach for developing educational modules. *Journal of the Brazilian Computer Society*, Springer, Heidelberg, Alemanha, v. 17, n. 4, p. 207–239, nov. 2011. ISSN 0104-6500, 1678-4804.

BARBOSA, Ellen Francine; SILVA, Marco Aurélio Graciotto. *Guidelines for the creation of training materials*. São Carlos, SP, Brasil, ago. 2009. Disponível em: `http://www.inct-sec.org/sites/default/files/aim-cid/Guidelines%20-%20AIMCID.pdf`. Acesso em: 20 fev 2012.

BARRETO, Hugo. Aprendizagem por televisão. In: LITTO, Fredric Michael; FORMIGA, Marcos (Ed.). *Educação a distância: o estado da arte*. 1. ed. São Paulo, SP, Brasil: Pearson Education, 2009. v. 1, cap. 60, p. 449–454. ISBN 978-85-7605-197-8.

BARUQUE, Lúcia Blondet; MELO, Rubens Nascimento. Learning theory and instructional design using learning objects. *Journal of Educational Multimedia and Hypermedia*, AACE, Norfolk, VA, EUA, v. 13, n. 4, p. 343–370, out. 2004. ISSN 1055-8896. Disponível em: `http://editlib.org/p/7432`.

BATES, Peter J. Learning through iDTV - results of t-learning study. In: *European Conference on Interactive Television: from Viewers to Actors?* Brighton, Reino Unido: ACM, 2003. (EuroITV), p. 137–138.

BBC. *BBC-i Interactive Television Style Guide*. ago. 2002. 54 p.

BELLONI, Maria Luiza. *Educação a Distância*. 5. ed. São Paulo, Brasil: Autores Associados, 2006. 124 p. (Coleção Educação Contemporânea). ISBN 978-85-85701-77-2.

BELLOTTI, F.; BERTA, R.; GLORIA, A. De; OZOLINA, A. Investigating the added value of interactivity and serious gaming for educational TV. *Computers & Education*, v. 57, n. 1, p. 1137–1148, 2011. ISSN 0360-1315.

BELLOTTI, Francesco; VROCHIDIS, Stefanos; PARISSI, Eirini; LHOAS, Pascal; MATHEVON, Damien; PELLEGRINO, Matteo; BO, Ioannis Kompatsiaris Giancarlo. A t-learning courses development and presentation framework. *IEEE Multidisciplinary Engineering Education Magazine*, IEEE Education Society, v. 3, n. 3, p. 69–76, set. 2008. ISSN 1558-7908.

BELLOTTI, F.; VROCHIDIS, S.; TSAMPOULATIDIS, I.; BO, G.; NAPOLETANO, L. A learning oriented technological framework for iDTV. In: *4th International Conference on Automated Solutions for Cross Media Content and Multi-Channel Distribution*. Florença, Itália: IEEE, 2008. p. 79–86. ISBN 9780769534060.

BLEVIS, Eli; STOLTERMAN, Erik. Transcending disciplinary boundaries in interaction design. *Interactions*, ACM, New York, NY, EUA, v. 16, n. 5, p. 48–51, set.-out. 2009. ISSN 1072-5520.

BLOOM, Benjamin S.; ENGELHART, Max D.; FURST, Edward J.; HILL, Walker H.; KRATHWOHL, David R. *Taxonomy of Educational Objectives: The Classification of Educational Goals (Handbook 1: Cognitive Domain)*. Ann Arbor, Michigan, EUA: Longmans, Green and Co, 1956. 207 p.

BORGES, Karen S.; BARBOSA, Maria Lúcia K.; VARELLA, Fernando; ROESLER, Valter. Educação através da TV digital utilizando metadados. In: *XIX Simpósio Brasileiro de Informática na Educação*. Fortaleza, CE, Brasil: SBC, 2008. (SBIE), p. 1–10. ISBN 857669207-4. Disponível em: `http://www.br-ie.org/pub/index.php/sbie/article/view/709`.

BORGES, Vanessa Araujo; MALDONADO, José Carlos; BARBOSA, Ellen Francine. Towards the establishment of supporting mechanisms for modeling and generating educational content. In: *Symposium On Applied Computing*. TaiChung, Taiwan: ACM, 2011. p. 1202–1207. ISBN 978-1-4503-0113-8.

BOURDA, Yolaine; GAUTHIER, Gilles; REGIL, Rosa-Maria Gomez de; CATTEAU, Olivier. Métadonnées derramar ressources d'apprentissage (mlr) – nouvelle norme ISO de description de ressources pédagogiques. *Sciences et Techniques de l'Information et de la Communication pour l'Éducation et la Formation (STICEF)*, v. 17, p. 1–11, 2010. ISSN 1764-7223. Disponível em: `http://sticef.univ-lemans.fr/num/vol2010/08r-bourda/sticef_2010_bourda_08r.htm`.

BOYD, Robert Dean; APPS, Jerold W. *Redefining the discipline of adult education*. EUA: Proquest Info & Learning, 1980. 219 p. ISBN 0835749339, 978-0835749336.

BOYLE, T. Design principles for authoring dynamic, reusable learning objects. *Australian Journal of Educational Technology*, v. 19, p. 46–58, 2003. Disponível em: `http://www.ascilite.org.au/ajet/ajet19/boyle.html`.

BRA, Paul De; HOUBEN, Geert-Jan; WU, Hongjing. AHAM: a dexter-based reference model for adaptive hypermedia. In: *10th ACM Conference on Hypertext and hypermedia : returning to our diverse roots*. New York, NY, EUA: ACM, 1999. (HYPERTEXT '99), p. 147–156. ISBN 1-58113-064-3.

BRANSON, Robert K.; RAYNER, Gail T.; COX, J. L.; FURMAN, John P.; KING, F. J. *Interservice Procedures for Instructional Systems Development. Executive Summary and Model*. EUA, ago. 1975. Disponível em: `http://handle.dtic.mil/100.2/ADA019486`.

Brasil. *Decreto nº 5.800*. jun. 2006. Decreto.

BULTERMAN, Dick C. A.; CESAR, Pablo; JANSEN, A. J. An architecture for viewer-side enrichment of TV content. In: *Proceedings of the 14th annual ACM international conference on Multimedia*. New York, NY, United States: ACM, 2006. (MULTIMEDIA '06), p. 651–654. ISBN 1-59593-447-2.

BUTCHER, Neil. *A Basic Guide to Open Educational Resources (OER)*. Vancouver,: COL, UNESCO, 2011. 133 p. ISBN 9781894975414.

CAEIRO-RODRíGUEZ, Manuel. PoEML: A separation-of-concerns proposal to instructional design. In: BOTTURI, Luca; STUBBS, Todd (Ed.). *Handbook of Visual Languages for Instructional Design: Theories and Practices*. EUA: IGI Global, 2008. p. 183–207. ISBN 9781599047317, 1599047314.

California State University. *Multimedia Educational Resource for Learning and Online Teaching (MERLOT)*. 1997. Projeto de Pesquisa. Disponível em: `http://www.merlot.org/`. Acesso em: 20 fev 2012.

CANAS, Alberto J.; HILL, Greg; BUNCH, Larry; CARFF, Roger; ESKRIDGE, Thomas; PéREZ, Carlos. KEA: A knowledge exchange architecture based on web services, concept maps and CmapTools. In: CANãS, A. J.; NOVAK, J. D. (Ed.). *International Conference on Concept Mapping*. San José, Costa Rica: University of Malta, 2006. v. 1, p. 304–310.

CANAS, A. J.; HILL, G.; CARFF, R.; SURI, N.; LOTT, J.; ESKRIDGE, T.; GóMEZ, G.; ARROYO, M.; CARVAJAL, R. CmapTools: A knowledge modeling and sharing environment.

In: *International Conference on Concept Mapping*. Pamplona Iruña, Navarra, Espanha: Universidad Pública de Navarra, 2004. v. 1, p. 125–133.

CANAS, Alberto J.; NOVAK, Joseph D. Concept mapping using CmapTools to enhance meaningful learning. In: OKADA, Alexandra; SHUM, Simon Buckingham; SHERBONE, Tony (Ed.). *Knowledge Cartography: Software Tools and Mapping Techniques*. 1. ed. London, Reino Unido: Springer, 2008, (Advanced Information and Knowledge Processing). cap. 2, p. 25–46. ISBN 978-1-84800-148-0.

CAO, Jinwei; CREWS, Janna M.; LIN, Ming; BURGOON, Judee K.; NUNAMAKER JR., Jay F. An empirical investigation of virtual interaction in supporting learning. *SIGMIS Database*, ACM, New York, NY, United States, v. 39, p. 51–68, jul. 2008. ISSN 0095-0033.

CAREY, John. Winky dink to stargazer: Five decades of interactive television. In: *UnivEd Conference on Interactive Television*. Edinburgh, Scotland, Reino Unido: University of Edinburgh, 1996. p. 2–21.

CARMICHAEL, Alex; RICE, Mark; SLOAN, David; GREGOR, Peter. Digital switchover or digital divide: a prognosis for usable and accessible interactive digital television in the UK. *Universal Access in the Information Society*, Springer-Verlag, Berlin, Alemanha, v. 4, n. 4, p. 400–416, maio 2006. ISSN 1615-5289.

CARVALHO, Eduardo Rodrigues de; BARROS, Gil Garcia de; COSTA, Laisa Caroline de Paula; FARIA, Regis Rossi Alves; NUNES, Rogério Pernas; LOPES, Roseli de Deus; ZUFFO, Marcelo Knörich. The brazilian digital television system access device architecture. *Journal of the Brazilian Computer Society*, SBC, Porto Alegre, RS, Brasil, v. 13, n. 1, p. 95–113, 2007. ISSN 0104-6500.

CASANOVA, Marco A.; TUCHERMAN, Luiz; LIMA, Maria Julia D.; NETTO, José L. Rangel; RODRIQUEZ, Noemi; SOARES, Luiz F. G. The nested context model for hyperdocuments. In: *3rd ACM Conference on Hypertext*. San Antonio, Texas, EUA: ACM, 1991. (HYPERTEXT), p. 193–201. ISBN 0-89791-461-9.

CATANIA, A. Charles. *Aprendizagem: Comportamento, Linguagem e Cognição*. 4. ed. Porto Alegre, RS, Brasil: Artes Médicas, 1999. 467 p. Traduçao por Deisy das Graças de Souza e outros. ISBN 85-7307-553-8.

CECHINEL, Cristian; SáNCHEZ-ALONSO, Salvador; GARCíA-BARRIOCANAL, Elena. Statistical profiles of highly-rated learning objects. *Computers & Education*, Elsevier, Oxford, Reino Unido, v. 57, n. 1, p. 1255–1269, ago. 2011. ISSN 0360-1315.

CELES, Clayson Sandro Francisco de Sousa; SOUZA, Cidcley Teixeira de. Estilos arquiteturais de software na construção de objetos de aprendizagem para a TV digital interativa. In: *XIX*

Simpósio Brasileiro de Informática na Educação. São Paulo, SP, Brasil: SBC, 2007. (SBIE), p. 290–299. ISBN 978-85-7669-156-3. Disponível em: `http://www.br-ie.org/pub/index.php/sbie/article/view/578`.

CERVO, Amado Luiz; BERVIAN, Pedro Alcino; SILVA, Roberto da. *Metodologia científica*. 6. ed. São Paulo, SP, Brasil: Pearson Prentice Hall, 2007. 159 p. ISBN 978-85-7605-047-6.

CESAR, Pablo; BULTERMAN, Dick C. A.; OBRENOVIC, Zeljko; DUCRET, Julien; CRUZ-LARA, Samuel. An architecture for non-intrusive user interfaces for interactive digital television. In: *5th European conference on Interactive TV: a shared experience*. Berlin, Alemanha: Springer, 2007. (EuroITV, v. 4471), p. 11–20. ISBN 978-3-540-72558-9.

CESAR, Pablo; CHORIANOPOULOS, Konstantinos. Interactivity and user participation in the television lifecycle: creating, sharing, and controlling content. In: *1st International Conference on Designing Interactive User Experiences for TV and Video*. New York, NY, EUA: ACM, 2008. (UXTV '08), p. 125–128. ISBN 978-1-60558-100-2.

CESAR, Pablo; CHORIANOPOULOS, Konstantinos; JENSEN, Jens F. Social television and user interaction. *Computers in Entertainment*, ACM, New York, NY, United States, v. 6, n. 1, p. 1–10, maio 2008. ISSN 1544-3574.

CHORIANOPOULOS, Konstantinos. Content-enriched communication supporting the social uses of TV. *Journal of the Communications Network*, v. 6, n. 1, p. 23–30, jan.-mar. 2007.

Cisco Systems, Inc. *Reusable Learning Object Strategy: Designing Information and Learning Objects Through Concept, Fact, Procedure, Process and Principle Templates*. nov. 2001. CISCO Systems. Disponível em: `http://www.cisco.com/warp/public/779/ibs/solutions/learning/whitepapers/el_cisco_rio.pdf`.

COFFEY, John W.; HOFFMAN, Robert; CANAS, Alberto. Concept map-based knowledge modeling: perspectives from information and knowledge visualization. *Information Visualization*, v. 5, p. 192–201, jun. 2006. ISSN 1473-8716.

COFFEY, J. W.; HOFFMAN, R.; CANAS, A. J.; FORD, K. M. A concept map-based knowledge modeling approach to expert knowledge sharing. In: *IASTED International Conference on Information and Knowledge Sharing (IKS)*. Virgin Islands, EUA: ACTA Press, 2002. p. 212–217.

COSTAGLIOLA, Gennaro; FERRUCCI, Filomena; FUCCELLA, Vittorio. SCORM run-time environment as a service. In: *International Conference on Web Engineering*. New York, NY, EUA: ACM, 2006. p. 103–110. ISBN 1-59593-352-2.

CUNNINGHAM, Ward. Wiki design principles. 2005. Disponível em: `http://c2.com/cgi/wiki?WikiDesignPrinciples`. Acesso em: 20 fev 2012.

DAMASEVICIUS, Robertas; STUIKYS, Vytautas. On the technological aspects of generative learning object development. In: MITTERMEIR, Roland; SYSLO, Maciej (Ed.). *International Conference on Informatics in Secondary Schools – Evolution and Perspectives (ISSEP)*. Torun, Poland: Springer, 2008. (Lecture Notes in Computer Science, v. 5090), p. 337–348. ISBN 978-3-540-69923-1.

DAMáSIO, Manuel José. Uses of interactive television on educational settings: Evaluating the media impact. In: *European Conference on Interactive Television: from Viewers to Actors?* Brighton, Reino Unido: ACM, 2003. (EuroITV), p. 117 – 119.

DAVE, Ravindrakumar H. Psychomotor levels. In: ARMSTRONG, Robert. J.; OTHERS (Ed.). Tucson, Arizona, USA: Educational Innovators, 1968. p. 33–34. Também conhecido como "A Systematic approach to developing and writing behavioral objectives: a handbook designed to increase the communication of laymen and educators".

DCMI. *Dublin Core Metadata Element Set*. out. 2010. Disponível em: `http://dublincore.org/documents/dces/`. Acesso em: 20 fev 2012.

DEFUDE, Bruno; FARHAT, Ramzi. A framework to design quality-based learning objects. In: *Fifth IEEE International Conference on Advanced Learning Technologies*. Washington, DC, EUA: IEEE Computer Society, 2005. (International Conference on Advanced Learning Technologies), p. 23–27. ISBN 0-7695-2338-2.

DEMILLO, R. A. Mutation analysis as a tool for software quality assurance. In: BRYAN, William; CHADBOURNE, Christopher; SIEGEL, Stan (Ed.). *COMPSAC*. Chicago, IL, EUA: IEEE, 1980. p. 390–393.

DICK, Walter; CAREY, Lou; CAREY, James O. *The Systematic Design of Instruction*. 6. ed. EUA: Pearson, 2005. 376 p.

DIX, Alan J.; FINLAY, Janet E.; ABOWD, Gregory D.; BEALE, Russel. *Human-Computer Interaction*. 2. ed. Inglaterra: Prentice Hall, 1998. 638 p. ISBN 0-13-239864.

DODERO, Juan; TATTERSALL, Colin; BURGOS, Daniel; KOPER, Rob. Transformational techniques for model-driven authoring of learning designs. In: LEUNG, Howard; LI, Frederick; LAU, Rynson; LI, Qing (Ed.). *International Conference on Advances in Web Based Learning*. Edinburgh, Reino Unido: Springer, 2007. (Lecture Notes in Computer Science, v. 4823), p. 230–241. ISBN 978-3-540-78138-7.

DODERO, J. M.; DIEZ, D. Model-driven instructional engineering to generate adaptable learning materials. In: *6th International Conference on Advanced Learning Technologies (ICALT 2006)*. Kerkrade, Países Baixos: IEEE Computer Society, 2006. p. 1188 – 1189. ISBN 0-7695-2632-2.

DOUGLAS, Ian. Instructional design based on reusable learning objects: applying lessons of object-oriented software engineering to learning systems design. In: *Frontiers in Education Conference*. Reno, Nevada, EUA: IEEE, 2001. v. 3, p. F4E–1 – F4E–5. ISBN 0-7803-6669-7.

DOWNES, Stephen. Learning objects: Resources for distance education worldwide. *International Review of Research in Open and Distance Learning*, v. 2, n. 1, 2001. ISSN 1492-3831. Disponível em: `http://www.irrodl.org/index.php/irrodl/article/view/32/378`.

DVB. *DVD Document A152: Digital Video Broadcasting (DVB); Globally Executable MHP (GEM) Specification 1.3 (including OTT and hybrid broadcast/broadband)*. mar. 2011. 860 p. Disponível em: `http://www.mhp.org/specs/A153_GEM_v1_3.pdf`. Acesso em: 20 fev 2012.

DYBA, T.; DINGSOYR, T.; HANSSEN, G.K. Applying systematic reviews to diverse study types: An experience report. In: *International Symposium on Empirical Software Engineering and Measurement*. Washington, DC, EUA: IEEE Computer Society, 2007. p. 225 –234. ISSN 1938-6451.

ERONEN, Leena. Combining quantitative and qualitative data in user research on digital television. In: *Panhellenic Conference with International Participation on Human-Computer Interaction*. Patras, Grécia: Typorama Publications, 2001. p. 51–56.

ERTELT, Anna; RENKL, Alexander; SPADA, Hans. Making a difference: exploiting the full potential of instructionally designed on-screen videos. In: *International Conference on Learning Sciences*. Bloomington, Indiana, EUA: International Society of the Learning Sciences, 2006. p. 154–160. ISBN 0-8058-6174-2.

ES, René Van; KOPER, Rob. Testing the pedagogical expressiveness of ims ld. *Journal of Educational Technology & Society*, International Forum of Educational Technology & Society, v. 9, n. 1, p. 229–249, jan. 2006. ISSN 1436-4522.

FARIA, Wilson de. *Aprendizagem e Planejamento de Ensino*. 1. ed. São Paulo, SP, Brasil: Ática, 1989. 84 p.

FILATRO, Andrea. As teorias pedagógicas fundamentais em EAD. In: LITTO, Fredric Michael; FORMIGA, Marcos (Ed.). *Educação a distância: o estado da arte*. 1. ed. São Paulo, SP, Brasil: Pearson Education, 2009. v. 1, cap. 14, p. 96–104. ISBN 978-85-7605-197-8.

FILHO, Guido Lemos de Souza; LEITE, Luiz Eduardo Cunha; BATISTA, Carlos Eduardo Coelho Freire. Ginga-J: the procedural middleware for the Brazilian digital TV system. *Journal of the Brazilian Computer Society*, v. 12, n. 4, p. 47–56, mar. 2007. ISSN 0104-6500.

FLóREZ, Julián; GARCíA, Igor; AIZPURUA, Iker; PALOC, Céline; UGARTE, Alejandro; JAINAGA, Igor; COLET, Jesús; ZUBIAUR, Xabier. SEITV – interactive multimedia leisure/educational services for digital TV in MHP. In: RAUTERBERG, Matthias (Ed.). *Third International Conference on Entertainment Computing*. Cambridge, Reino Unido: Springer, 2004. (Lecture Notes in Computer Science, v. 3166), p. 109–119. ISBN 978-3-540-22947-6.

FORD, Kenneth; CANAS, Alberto; JONES, Jeremy; STAHL, Howard; NOVAK, Joseph; ADAMS-WEBBER, Jack. ICONKAT: an integrated constructivist knowledge acquisition tool. *Knowledge Acquisition*, v. 3, n. 2, p. 215 – 236, jun. 1991. ISSN 1042-8143.

FORTE, Eddy N.; FORTE, Maria H. K. Wentland; DUVAL, Erik. The ariadne project (part 1): Knowledge pools for computer-based and telematics-supported classical, open and distance education. *European Journal of Engineering Education*, v. 22, n. 1, p. 61–74, 1997. ISSN 0304-3797, 1469-5898.

FRANTZI, M.; MOUMOUTZIS, N.; CHRISTODOULAKIS, S. A methodology for the integration of SCORM with TV-Anytime for achieving interoperable digital TV and e-learning applications. In: *IEEE International Conference on Advanced Learning Technologies*. Joensuu, Finlândia: IEEE, 2004. (International Conference on Advanced Learning Technologies), p. 636–638.

FREIRE, Paulo Reglus Neves. *Pedagogia da autonomia: saberes necessárias à prática educativa*. 39. ed. São Paulo, SP, Brasil: Paz e Terra, 1996. 148 p. (Coleção Leitura). ISBN 978-85-7753-015-1.

FREIRE, Paulo Reglus Neves; GUIMARÃES, Sérgio. *Sobre Educação (Diálogos)*. 3. ed. São Paulo, SP, Brazil: Paz e Terra, 2003. 196 p. ISBN 85-219-0682-X.

GAINES, B. R.; SHAW, M. L. G. Concept maps as hypermedia components. *International Journal of Human-Computer Studies*, Elsevier, Duluth, MN, United States, v. 43, n. 3, p. 323–361, set. 1995. ISSN 1071-5819.

GARZOTTO, Franca; PAOLINI, Paolo; SCHWABE, Daniel. HDM: a model-based approach to hypertext application design. *ACM Transactions on Information Systems*, ACM Press, New York, NY, EUA, v. 11, n. 1, p. 1 – 26, jan. 1993.

GASPAR, Alberto. A educação formal e a educação informal em ciências. In: MASSARANI, Luisa; MOREIRA, Lideu de Castro; BRITO, Fatima (Ed.). 1. ed. Rio de Janeiro, RJ, Brazil:

Casa da Ciência – Centro Cultural de Ciência e Tecnologia da Universidade Federal do Rio de Janeiro, 2002, (Série Terra Incógnita, v. 1). cap. A educação formal e a educação informal em ciências, p. 171–183. ISBN 85-89229-01-7.

GIMENES, Itana M. S.; BARROCA, Leonor; FELTRIM, Valéria D. Tendências na educação a distância e educação aberta na computação. In: *Congresso da Sociedade Brasileira da Computação (CSBC) – XXXI Jornadas de Atualização em Informática (JAI)*. [S.l.: s.n.], 2012. cap. 1, p. 1–41.

HALASZ, Frank; SCHWARTZ, Mayer. The Dexter hypertext reference model. *Communications of the ACM*, ACM, New York, NY, United States, v. 37, n. 2, p. 30–39, fev. 1994. ISSN 0001-0782.

HAREL, D. Statecharts: A visual formalism for complex systems. *Science of Computer Programming*, v. 8, n. 3, p. 231–274, jun. 1987.

HASSENZAHL, M.; TRACTINSKY, N. User experience – a research agenda. *Behaviour & Information Technology*, v. 25, n. 2, p. 91–97, 2006.

HOFFMAN, R.R.; ZIEBELL, D.; FIORE, S.M.; BECERRA-FERNANDEZ, I. Knowledge management revisited. *IEEE Intelligent Systems*, IEEE, v. 23, n. 3, p. 84 –88, maio-jun. 2008. ISSN 1541-1672.

HULSEN, P; KIM, J. G.; LEE, H. K; KANG, K. O. Delivering T-learning with TV-anytime through packaging. In: *International Symposium on Consumer Electronics*. Reading, Reino Unido: IEEE, 2004. p. 614–619. ISBN 0-7803-8527-6.

HUMPHREY, Watts S. Boston, MA, United States: Addison-Wesley, 1989. 512 p. ISBN 978-0201180954, 0201180952.

IBGE. *Síntese de Indicadores Sociais: Uma Análise das Condições de Vida da População Brasileira*. Rio de Janeiro, RJ, Brasil, 2009. Disponível em: `http://biblioteca.ibge.gov.br/visualizacao/monografias/GEBIS%20-%20RJ/sintese_indic/indic_sociais2009.pdf`. Acesso em: 20 fev 2012.

IEEE. Draft Standard, *IEEE Standard for Learning Object Metadata (IEEE 1484.12.1)*. jun. 2002.

IEEE. Draft Standard, *IEEE Standard for Learning Technology – Extensible Markup Language (XML) Schema Definition Language Binding for Learning Object Metadata (IEEE 1484.12.3)*. nov. 2005.

IMS. *Learning Design Specification*. jan. 2003. Disponível em: `http://www.imsglobal.org/learningdesign/`. Acesso em: 20 fev 2012.

IMS. *Simple Sequencing Specification*. mar. 2003. Disponível em: http://www.imsglobal.org/simplesequencing/. Acesso em: 20 fev 2012.

IMS. *Content Package Specification*. mar. 2007. Disponível em: http://www.imsglobal.org/content/packaging/. Acesso em: 20 fev 2012.

IMS. *Common Cartridge*. out. 2011. Disponível em: http://www.imsglobal.org/cc/. Acesso em: 20 fev 2012.

ISAKOWITZ, Tomás; STOHR, Edward A.; BALASUBRAMANIAN, P. RMM: A methodology for structured hypermedia design. *Communications of the ACM*, ACM, New York, NY, United States, v. 38, n. 8, p. 34 – 44, ago. 1995. ISSN 0001-0782.

ISO/IEC. *ISO/IEC 11172-3:1993 - Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s – Part 3: Audio*. 1993. Padrão.

ISO/IEC. *ISO/IEC 12207 – Standard for Information Technology - Software life cycle processes*. aug 1995.

ISO/IEC. *ISO/IEC 13818-2:2000 - Information technology – Generic coding of moving pictures and associated audio information: Video*. 2000. 208 p.

ISO/IEC. *ISO/IEC 14496-2:2004 – Information technology – Coding of audio-visual objects – Part 2: Visual*. 2004. 706 p.

ISO/IEC. *ISO/IEC 13818-7:2006 - Information technology – Generic coding of moving pictures and associated audio information – Part 7: Advanced Audio Coding (AAC)*. 2006. 194 p.

ISO/IEC. *ISO/IEC 13818-1:2007 - Information technology – Generic coding of moving pictures and associated audio information: Systems*. 2007. 174 p.

ISO/IEC. *ISO/IEC 12207 – Standard for Information Technology - Software life cycle processes*. 2008. 123 p. Disponível em: http://www.iso.org/iso/catalogue_detail?csnumber=43447.

ISO/IEC. *ISO/IEC 15288 – Systems and software engineering – System life cycle processes*. 2008. 123 p. Disponível em: http://www.iso.org/iso/catalogue_detail?csnumber=43447.

ISO/IEC. *ISO/IEC 14496-3:2009 – Information technology – Coding of audio-visual objects – Part 3: Audio*. 2009. 1381 p.

ISO/IEC. *ISO/IEC 14496-10:2010 – Information technology – Coding of audio-visual objects – Part 10: Advanced Video Coding*. 2010. 693 p.

ITU. *J.200: Worldwide common core – Application environment for digital interactive television services*. abr. 2010. Disponível em: `http://www.itu.int/rec/T-REC-J.200-201004-I/en`. Acesso em: 20 fev 2012.

ITU. *J.202: Harmonization of procedural content formats for interactive TV applications*. ago. 2010. Disponível em: `http://www.itu.int/rec/T-REC-J.202-201008-I/en`. Acesso em: 20 fev 2012.

JENSEN, Jens F. Interactivity: Tracing a new concept in media and communication studies. *Nordicom Review*, v. 19, p. 185–204, jul. 1998.

JENSEN, Jens F. Interactive television: new genres, new format, new content. In: *2nd Australasian Conference on Interactive Entertainment*. Sydney, Austrália: Creativity & Cognition Studios, 2005. (IE), p. 89–96. ISBN 0-9751533-2-3.

JONASSEN, David. Objectivism versus constructivism: Do we need a new philosophical paradigm? *Educational Technology Research and Development*, Springer Boston, v. 39, n. 3, p. 5–14, 1991. ISSN 1042-1629. 10.1007/BF02296434.

JONASSEN, David H. Thinking technology: Toward a constructivist design model. *Educational Technology*, v. 34, n. 4, p. 34–37, abr. 1994. ISSN 0013-1962.

JONES, M. K.; LI, Z.; MERRILL, M. D. Rapid prototyping in automated instructional design. *Education Technology Research and Development*, v. 40, n. 4, p. 95–100, 1992.

JSR. *JSR 219: Foundation Profile 1.1*. dez. 2005. Java Specification Request (JSR). Disponível em: `http://jcp.org/en/jsr/detail?id=219`. Acesso em: 20 fev 2012.

JSR. *JSR 216: Personal Profile 1.1*. ago. 2006. Java Specification Request (JSR). Disponível em: `http://jcp.org/en/jsr/detail?id=216`. Acesso em: 20 fev 2012.

JSR. *JSR 217: Personal Basis Profile 1.1*. ago. 2006. Java Specification Request (JSR). Disponível em: `http://jcp.org/en/jsr/detail?id=217`. Acesso em: 20 fev 2012.

JSR. *JSR 218: Connected Device Configuration (CDC) 1.1*. ago. 2006. Java Specification Request (JSR). Substituído por (JSR, 2006c). Disponível em: `http://jcp.org/en/jsr/detail?id=218`. Acesso em: 20 fev 2012.

JSR. *JSR 927: Java TV API 1.1*. ago. 2008. Java Specification Request (JSR). Disponível em: `http://jcp.org/en/jsr/detail?id=927`. Acesso em: 20 fev 2012.

KAWASAKI, Evelise Izumi. *Modelo e metodologia para projeto de cursos hipermídia*. Dissertação (Mestrado) — Instituto Tecnológico de Aeronáutica (ITA), São José dos Campos, SP, Brasil, nov. 1996. Orientador: Clovis Torres Fernandes. Disponível em: `http://www.bd.bibl.ita.br/tesesdigitais/000373683.pdf`.

KAWASAKI, Evelise Izumi; FERNANDES, Clovis Torres. Modelo para projeto de cursos hipermídia. In: *VII Simpósio Brasileiro de Informática na Educação (SBIE 96)*. Belo Horizonte, MG, Brasil: SBC, 1996. p. 227–240.

KHAN, Salman; OTHERS. *Khan Academy*. set. 2006. Projeto de Pesquisa. Disponível em: `http://www.khanacademy.org`. Acesso em: 20 fev 2012.

KITCHENHAM, Barbara. *Procedures for Performing Systematic Reviews*. Keele, Reino Unido, jul. 2004. Disponível em: `http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.122.3308&rep=rep1&type=pdf`.

KOPER, R. *Modeling Units of Study from a Pedagogical Perspective: The Pedagogical Meta-Model behind EML*. jun. 2001. White paper (Open University of The Netherlands). Disponível em: `http://hdl.handle.net/1820/36`.

KRATHWOHL, David R.; BLOOM, Benjamin S.; MASIA, Bertram B. *Taxonomy of Educational Objectives: The Classification of Educational Goals (Handbook 2: Affective Domain)*. EUA: Longmans, Green and Co, 1964.

LAVONEN, Jari; MEISALO, Veijo; LATTU, Matti; LEINONEN, Liisa; WILUSZ, Tadeusz. Using computers in science and technology education. *ACM SIGCSE Bulletin*, v. 33, n. 2, p. 127–135, 2001.

LEARY, Heather; RECKER, Mimi; WALKER, Andrew; WETZLER, Philipp; SUMNER, Tamara; MARTIN, James. Automating open educational resources assessments: a machine learning generalization study. In: NEWTON, Glen; WRIGHT, Michael; CASSEL, Lillian N. (Ed.). *International Joint Conference on Digital Libraries*. New York, NY, United States: ACM, IEEE, 2011. (JCDL), p. 283–286. ISBN 978-1-4503-0744-4.

LEITE, Luiz Eduardo Cunha; BATISTA, Carlos Eduardo Coelho Freire; FILHO, Guido Lemos de Souza; KULESZA, Raoni; ALVES, Luiz Gustavo Pacola; BRESSAN, Graça; RODRIGUES, Rogerio Ferreira; SOARES, Luiz Fernando Gomes. FlexTV – uma proposta de arquitetura de middleware para o sistema brasileiro de TV digital. *Revista de Engenharia de Computação e Sistemas Digitais*, PCS-POLI-USP, São Paulo, SP, Brasil, n. 2, p. 29–50, nov. 2005. ISSN 1678-8435. Disponível em: `http://www.pcs.poli.usp.br/revista/n2/r002a003.pdf`.

LEIVA, Willie Dresler. *Um Modelo de Hipertexto para Apoio ao Ensino Mediado pela Web*. Tese (Doutorado) — ICMC-USP, São Carlos, SP, Brasil, jul. 2003. Disponível em: `http://www.teses.usp.br/teses/disponiveis/55/55134/tde-17112003-071043/`.

LEIVA, W. D.; OLIVEIRA, M. C. F.; MASIERO, P. C.; FORTES, R. P. M. MDE: A hypertext model for online courses. In: PIMENTEL, Maria da Graça Campos; RIOS, Riverson (Ed.). *Simpósio Brasileiro de Multimídia e Sistemas Hipermídia – Workshop de Ferramentas e Aplicações*. Fortaleza: BNB, 2002. v. 1, p. 333–340. ISBN 858844238-8.

LEVESON, Nancy G.; TURNER, Clark S. An investigation of the Therac-25 accidents. *Computer*, IEEE Computer Society, United States, v. 26, n. 7, p. 18–41, jul. 1993. ISSN 0018-9162.

LIN, H. W.; CHANG, Wen-Chih; YEE, George; SHIH, Thimothy K.; WANG, Chun-Chia; YANG, Hsuan-Che. Applying petri nets to model scorm learning sequence specification in collaborative learning. In: *International Conference on Advanced Information Networking and Applications*. Washington, DC, EUA: IEEE, 2005. v. 1, p. 203–208. ISBN 0-7695-2249-1. ISSN 1550-445X.

LIN, H. W.; SHIH, Timothy K.; CHANG, Wen-Chih; YANG, Chao-Hsun; WANG, Chun-Chia. A petri nets-based approach to modeling scorm sequence. In: *IEEE International Conference on Multimedia and Expo (ICME)*. Taipei, Taiwan: IEEE, 2004. p. 1247–1250. ISBN 0-7803-8603-5.

LITTO, Fredric Michael. O atual cenário internacional da EAD. In: LITTO, Fredric Michael; FORMIGA, Marcos (Ed.). *Educação a distância: o estado da arte*. 1. ed. São Paulo, SP, Brasil: Pearson Education, 2009. v. 1, cap. 3, p. 14–20. ISBN 978-85-7605-197-8.

LóPEZ-NORES, M.; BLANCO-FERNáNDEZ, Y.; PAZOS-ARIAS, J. J. Architecting multimedia-rich collaborative learning services over interactive digital TV. In: *5th Iberian Conference on Information Systems and Technologies*. Santiago de Compostela, Espanha: IEEE, 2010. p. 1–6. ISBN 9789899624733.

LóPEZ-NORES, Martín; PAZOS-ARIAS, José Juan; GARCíA-DUQUE, Jorge; BLANCO-FERNáNDEZ, Yolanda; GIL-SOLLA, Alberto. A core of standards to support t-learning. In: *Sixth IEEE International Conference on Advanced Learning Technologies*. Washington, DC, EUA: IEEE, 2006. ISBN 0-7695-2632-2.

MACHADO, Joice B.; SILVA, Marco Aurélio Graciotto; MALDONADO, José C.; BARBOSA, Ellen F. Reengineering of educational content: an experience in the computer networks domain. In: *XXII Simpósio Brasileiro de Informática na Educação*. Aracaju, SE, Brasil:

SBC, 2011. (SBIE), p. 361–370. ISSN 2176-4301. Disponível em: http://www.br-ie.org/sbie-wie2011/SBIE-Trilha2/93202_1.pdf.

MAFRA, Sômulo Nogueira; BARCELOS, Rafael Ferreira; TRAVASSOS, Guilherme Horta. Aplicando uma metodologia baseada em evidência na definição de novas tecnologias de software. In: MASIERO, Paulo Cesar (Ed.). *20th Brazilian Symposium on Software Engineering*. [S.l.], 2006. (Brazilian Symposium on Software Engineering), p. 239–254. ISBN 85-7669-079-9.

MALDONADO, José Carlos; DELAMARO, Márcio Eduardo; FABBRI, Sandra C. P. F.; SIMãO, Adenilso da Silva; SUGETA, Tatiana; VINCENZI, Auri Marcelo Rizzo; MASIERO, Paulo Cesar. Proteum: A family of tools to support specification and program testing based on mutation. In: WONG, W. Eric (Ed.). *MUTATION 2000: A Symposium on Mutation Testing for the new Century*. [S.l.]: Kluwer Academic Publishers, 2000. (Advances in Database Systems), p. 113–116. ISBN 0-7923-7323-5.

MARTINS, Diogo S.; OLIVEIRA, Lílian S.; PIMENTEL, Maria da Graça C. Designing the user experience in iTV-based interactive learning objects. In: ANACLETO, Junia C.; FORTES, Renata Pontin de M.; COSTA, Carlos J. (Ed.). *International Conference on Design of Communication*. New York, NY, EUA: ACM, 2010. (SIGDOC '10), p. 243–250. ISBN 978-1-4503-0403-0.

MAYER, Richard E.; MORENO, Roxana. Nine ways to reduce cognitive load in multimedia learning. *Educational Psychologist*, Lawrence Erlbaum, v. 38, n. 1, p. 43–52, 2003.

MCGREAL, Rory; ANDERSON, Terry; BABIN, Gilbert; DOWNES, Stephen; FRIESEN, Norm; HARRIGAN, Kevin; HATALA, Marek; MACLEOD, Doug; MATTSON, Mike; PAQUETTE, Gilbert; RICHARDS, Griff; ROBERTS, Toni; SCHAFER, Steve. Edusource: Canada's learning object repository network. *International Journal of Instructional Technology and Distance Learning*, p. 1–16, mar. 2004. ISSN 1550-6908. Disponível em: http://www.itdl.org/Journal/Mar_04/article01.htm.

MEISZNER, Andreas. *The Why and How of Open Education – With lessons from the openSE and openED projects*. Países Baixos, out. 2011. 107 p. Version 1.5.

MERRILL, M. D. Component display theory. In: REIGELUTH, Charles M. (Ed.). *Instructional Design Theories and Models: An Overview of their Current States*. Hillsdale, NJ: Lawrence Erlbaum, 1983. v. 1, p. 487. ISBN 0898592755. Também conhecido como 'Green Book'.

MERRILL, M. David. Components of instruction: toward a theoretical tool for instructional design. *Instructional Science*, v. 29, n. 4-5, p. 291–310, 2001.

MEYER, Marek; RENSING, Christoph; STEINMETZ, Ralf. Multigranularity reuse of learning resources. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, ACM, New York, NY, EUA, v. 7, n. 1, p. 1–23, jan. 2011. ISSN 1551-6857.

MICHENER, Edwina Rissland. Understanding understanding mathematics. *Cognitive Science*, v. 2, n. 4, p. 361–383, jan. 1978.

MONTEIRO, Bruno de Souza; PROTA, Thiago Monteiro; SOUZA, Fernando da Fonseca de; GOMES, Alex Sandro. Desenvolvimento de objetos de aprendizagem para TVDi. In: *XIX Simpósio Brasileiro de Informática na Educação*. Fortaleza, CE, Brasil: SBC, 2008. (SBIE), p. 1–10. ISBN 857669207-4.

MONTEIRO, Bruno de Sousa; SOUZA, Fernando da Fonseca de; GOMES, Alex Sandro. Amadeus-tv: Portal educacional na tv digital integrado a um sistema de gestão de aprendizado. In: *XX Simpósio Brasileiro de Informática na Educação*. Florianópolis, SC, Brasil: SBC, 2009. (SBIE), p. 1–10. ISSN 2176-4301.

MOORE, Michael Graham. Three types of interaction. *The American Journal of Distance Education*, Routledge, v. 3, n. 2, p. 1–6, 1989. ISSN 0892-3647, 1538-9286.

MOORE, Michael G. Recent contributions to the theory of distance education. *Open Learning*, Wadsworth, v. 5, n. 3, p. 10–15, 1990.

MORAN, José Manuel. *O que é educação a distância*. 2002. Artigo. Disponível em: `http://www.eca.usp.br/prof/moran/dist.htm`. Acesso em: 20 fev 2012.

MORRISON, Gary R.; ROSS, Steven M.; KEMP, Jerrold E.; KALMAN, Howard. *Designing effective instruction*. 6. ed. Hoboken, NJ, EUA: John Wiley & Sons, 2010. 491 p. ISBN 0470522828, 978-0470522820.

NASCIMENTO, A.; MORGADO, E. *Um projeto de colaboração internacional na América Latina*. 2003. Projeto de pesquisa. Disponível em: `http://www.rived.mec.gov.br/artigos/rived.pdf`. Acesso em: 20 apr 2009.

NATHAN, Mukesh; HARRISON, Chris; YAROSH, Svetlana; TERVEEN, Loren; STEAD, Larry; AMENTO, Brian. CollaboraTV: making television viewing social again. In: *1st International Conference on Designing Interactive User Experiences for TV and Video*. Silicon Valley, California, United States: ACM, 2008. (UXTV), p. 85–94. ISBN 978-1-60558-100-2.

NEWBY, T. J.; STEPICH, D. A.; LEHMAN, J. D.; RUSSELL, J. D. Instructional technology for teaching and earning-designing instruction, integrating computers. In: STOLLENWERK,

Debra A. (Ed.). *Using media theory into application*. New Jersey, EUA: Prentice-Hall, 1996. p. 24–43.

NOVAK, Joseph Donald. *A theory of education*. Ithaca, NY, EUA: Cornell University Press, 1977. 295 p. ISBN 0801411041, 9780801411045.

NOVAK, Joseph D. Concept mapping: A useful tool for science education. *Journal of Research in Science Teaching*, v. 27, n. 10, p. 937–949, 1990. ISSN 0022-4308.

NOVAK, Joseph D. *Learning, Creating, and Using Knowledge: Concept Maps As Facilitative Tools in Schools and Corporations*. EUA: Lawrence Erlbaum Associates, 1998. 251 p. ISBN 978-0805826265.

NOVAK, Joseph D.; GOWIN, D. Bob. *Learning How to Learn*. New York, NY, EUA: Cambridge University, 1984. 199 p. ISBN 978-0-521-26507-2, 978-0-521-31926-3. Disponível em: `http:/www.cambridge.org/9780521265072`.

OCW Consortium. *OpenCourseWare*. 2002. Projeto de Pesquisa. Disponível em: `http://www.ocwconsortium.org/en/aboutus/whatisocw`. Acesso em: 20 fev 2012.

OLIVIER, Bill; TATTERSALL, Colin. The learning design specification. In: KOPER, Rob; TATTERSALL, Colin (Ed.). *Learning Design*. 1. ed. Países Baixos: Springer, 2005. cap. 2, p. 21–40. ISBN 3-540-22814-4.

OMG. *Software & Systems Process Engineering Metamodel (SPEM)*. abr. 2008. Disponível em: `http://www.omg.org/spec/SPEM/2.0`.

PANSANATO, L. T. E.; NUNES, M. G. V. Autoria de aplicações hipermídia para ensino. *Revista Brasileira de Informática na Educação*, v. 1, n. 5, p. 103–124, set. 1999.

PAPERT, Seymour. Interviews with seymour papert. *Computers in Entertainment*, ACM, New York, NY, EUA, v. 2, p. 9–9, jan. 2004. ISSN 1544-3574.

PAQUETTE, Gilbert. Telelearning systems engineering – towards a new ISD model. *Journal of Structural Learning*, n. 14, p. 1–35, 2001.

PAQUETTE, Gilbert. Educational modelling languages from an instructional engineering perspective. In: MCGREAL, Rory (Ed.). *Online education using learning objects*. 1. ed. EUA: Routledge, 2004. cap. 26, p. 299–311. ISBN 0415335124, 978-0415335126.

PAQUETTE, Gilbert; LéONARD, Michel; LUNDGREN-CAYROL, Karin. The MOT+ visual language for knowledge-based instructional design. In: *Instructional Design: Concepts, Methodologies, Tools and Applications*. Londres, Reino Unido: IGI Global, 2011. cap. 3.12, p. 697–717.

PAQUETTE, Gilbert; TEJA, Ileana de la; LEóNARD, Michel; LUNDGREN-CAYROL, Karin; MARINO, Olga. An instructional engineering method and tool for the design of units of learning. In: KOPER, Rob; TATTERSALL, Colin (Ed.). *Learning Design*. 1. ed. Países Baixos: Springer, 2005. cap. 9, p. 161–184. ISBN 3-540-22814-4.

PASCUAL-NIETO, Ismael; MARíN, Diana Rosario Pérez; RODRíGUEZ, Pilar; O'DONNELL, Mick. Using automatically generated students' clickable conceptual models for e-tutoring. In: EKLUND, Peter; HAEMMERTé, Ollivier (Ed.). *16th International Conference on Conceptual Structures – Supllementary proceedings*. Toulouse, França: Springer Verlag, 2008. p. 1–8. Disponível em: `http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-354/p10.pdf`.

PAULOVICH, Fernando V.; MINGHIM, Rosane. Text map explorer: a tool to create and explore document maps. In: BANISSI, Ebad; BURKHARD, Remo Aslak; URSYN, Anna; ZHANG, Jian J; BANNATYNE, Mark; MAPLE, Carsten; COWELL, Andrew J.; TIAN, Gui Yun; ; HOU, Ming (Ed.). *Tenth International Conference on Information Visualization*. Los Alamitos, CA, EUA: IEEE Computer Society, 2006. p. 245–251. ISBN 0-7695-2602-0. ISSN 1550-6037.

PAZOS-ARIAS, José J.; LóPEZ-NORES, Martín; GARCíA-DUQUE, Jorge; DíAZ-REDONDO, Rebeca P.; BLANCO-FERNáNDEZ, Yolanda; RAMOS-CABRER, Manuel; GIL-SOLLA, Alberto; FERNáNDEZ-VILAS, Ana. Provision of distance learning services over interactive digital TV with MHP. *Computers & Education*, Elsevier, Oxford, Reino Unido, v. 50, n. 3, p. 927–949, abr. 2008. ISSN 0360-1315.

PEDROSA, Diogo; MARTINS JR., José Augusto C.; MELO, Erick L.; TEIXEIRA, Cesar A. C. A multimodal interaction component for digital television. In: *ACM Symposium on Applied Computing*. New York, NY, EUA: ACM, 2011. (SAC '11), p. 1253–1258. ISBN 978-1-4503-0113-8.

PERRATON, Hilary. *Open and Distance Learning in the Developing World*. 2. ed. United States: Routledge, 2006. 256 p. ISBN 978-0-415-39398-0, 978-0-415-39397-3.

PESSOA, Marcello de Castro; BENITTI, Fabiane Barreto Vavassori. Proposta de um processo para produção de objetos de aprendizagem. *Hífen*, Uruguaiana, RS, Brasil, v. 32, n. 62, p. 172–180, 2008. ISSN 1983-6511.

PETRI, Carl Adam. *Kommunikation mit Automaten*. Alemanha, jun. 1962. Orientador: A. Walther. Disponível em: `http://epub.sub.uni-hamburg.de/informatik/volltexte/2011/160/`.

PICKMAN, Charles H.; LIU, Dar-Biau. A low cost common educational framework. *Journal of Computing Sciences in Colleges*, v. 24, n. 4, p. 238–244, 2009.

PéREZ-MARíN, D.; ALFONSECA, E.; RODRíGUEZ, P.; PASCUAL-NIETO, I. Automatic generation of students' conceptual models from answers in plain text. In: CONATI, Cristina; MCCOY, Kathleen; PALIOURAS, Georgios (Ed.). *International Conference on User Modeling*. Corfu, Grécia: Springer, 2007. (Lecture Notes in Computer Science, v. 4511), p. 329–333. ISSN 0302-9743.

PéREZ-MARíN, Diana; PASCUAL-NIETO, Ismael. Showing automatically generated students' conceptual models to students and teachers. *International Journal of Artificial Intelligence in Education*, IOS Press, Amsterdam, Países Baixos, v. 20, n. 1, p. 47–72, jan. 2010. ISSN 1560-4292. Disponível em: `http://iaied.org/pub/1305/`.

QUADROS, Teresinha; MARTINS, Joberto S. B. A prática interdisciplinar em programas de educação a distância num cenário de novas tecnologias da informação e comunicação. In: *XVI Simpósio Brasileiro de Informática na Educação*. Juiz de Fora, MG, Brasil: SBC, 2005. (SBIE), p. 351–361. ISSN 2176-4301. Disponível em: `http://br-ie.org/pub/index.php/sbie/article/view/419`.

RAWLINGS, Adrian; ROSMALEN, Peter van; KOPER, Rob; RODRíGUEZ-ARTACHO, Miguel; LEFRERE, Paul. Survey of educational modelling languages (EMLs). In: *CEN/ISSS Workshop on Learning Technologies (WS/LT)*. [s.n.], 2002. Disponível em: `http://www.cenorm.be/cenorm/businessdomains/businessdomains/isss/activity/emlsurveyv1.pdf`.

REDEKER, Giselher H. J. An educational taxonomy for learning objects. In: *International Conference on Advanced Learning Technologies*. Athens, Grécia: IEEE, 2003. p. 250 – 251. ISBN 0-7695-1967-9.

REIGELUTH, Charles M. Instrucional design: What is it and why is it? In: REIGELUTH, Charles M. (Ed.). *Instructional Design Theories and Models: An Overview of their Current Status*. Hillsdale, NJ, EUA: Erlbaum, 1983. v. 1, cap. 1, p. 3–36. ISBN 0898592755. Também conhecido como 'Green Book'.

REIGELUTH, Charles M.; CARR-CHELLMAN, Alison A. Understanding instructional theory. In: REIGELUTH, Charles M.; CARR-CHELLMAN, Alison A. (Ed.). *Instructional Design Theories and Models: Building a Common Knowledge Base*. New York, NY, EUA: Routledge, 2009. v. 3, cap. 1, p. 3–26. ISBN 987-0-8058-6456-4. Também conhecido como 'Green Book III'.

ROBBINS, Jason Elliot. *Cognitive Support Features for Software Development Tools*. Tese (Doutorado) — University of California, Irvine, EUA, 1999. Orientador: David F. Redmiles.

ROMISZOWSKI, Alexander J.; ROMISZOWSKI, Hermelina P. *Dicionário de Terminologia de Educação a Distância*. Rio de Janeiro, RJ, Brasil: Fundação Roberto Marinho, 1998.

ROWNTREE, Derek. *Exploring Open and Distance Learning*. Padstow, Reino Unido: Routledge, 1992. 299 p. ISBN 074940813-8, 978-0749408138.

SANTOS, Cristina Mamédio da Costa; PIMENTA, Cibele Andrucioli de Mattos; NOBRE, Moacyr Roberto Cuce. A estratégia pico para a construção da pergunta de pesquisa e busca de evidências. *Revista Latino-Americana de Enfermagem*, v. 15, n. 3, p. 508–511, maio-jun. 2007.

SANTOS, Davi Trindade dos; VALE, Douglas Terêncio do; MELONI, Luís Geraldo Pedroso. Digital TV and distance learning: Potentials and limitations. In: *Frontiers in Education Conference*. San Diego, CA, EUA: IEEE, 2006. p. 1 –6. ISSN 0190-5848.

SANTOS, Joel André Ferreira dos; SAADE, Débora Christina Muchaluat. Xtemplate 3.0: spatio-temporal semantics and structure reuse for hypermedia compositions. *Multimedia Tools and Applications*, Springer Netherlands, p. 1–29, 2011. ISSN 1380-7501.

SCHMIDT, Douglas C. Model-driven engineering. *IEEE Software*, IEEE, v. 39, n. 2, p. 25–31, fev. 2006. ISSN 0018-9162.

SCHWABE, Daniel; ROSSI, Gustavo. Building hypermedia applications as navigational views of information models. In: *28th Hawaii International Conference on Information Systems*. Maui, Havaí: IEEE, 1995. v. 3, p. 231–240. ISBN 0-8186-6940-3.

SGRIGNOLI, Gary. History of ATSC digital television transmission system. In: *International Conference on Consumer Electronics (ICCE)*. Las Vegas, NV, EUA: IEEE, 2007. p. 1–2. ISBN 1-4244-0763-X, 1-4244-0763-X.

SHAW, Mary; GARLAN, David. *Software architecture: perspectives on an emerging discipline*. 1. ed. Upper Saddle River, NJ, United States: Prentice-Hall, 1996. 242 p. ISBN 978-0131829572.

SHULL, F.; CARVER, J.; TRAVASSOS, G. H. An empirical methodology for introducing software processes. In: *Joint 8th European Software Engineering Conference (ESEC) and 9th ACM SIGSOFT Symposium on the Foundations of Software Engineering (FSE-9)*. [S.l.]: ACM, 2001. p. 288–296. ISBN 1-58113-390-1.

SILVA, Lincoln David Nery e; TAVARES, Tatiana Aires; FILHO, Guido Lemos de Souza. Desenvolvimento de programas de TVDI explorando as funções inovadoras do Ginga-J. In: FILHO, José Gonçalves Pereira; GOMES, Roberta Lima; GEROSA, Marco Aurélio; GUIZZARDI, Renata Silva Souza (Ed.). *14th Brazilian Symposium on Multimedia and the Web*. [S.l.]: ACM, 2008. (WebMedia), p. 75–82. ISBN 978-85-7669-199-0.

SILVA, Marco Aurélio Graciotto. *Uma ferramenta Web colaborativa para apoiar a engenharia de requisitos em software livre*. 1-12 p. Dissertação (Mestrado) — Universidade de São Paulo, São Carlos, SP, Brasil, nov 2005. Orientadora: Renata Pontin de Mattos Fortes. Disponível em: `http://www.teses.usp.br/teses/disponiveis/55/55134/tde-28042006-080550/publico/dissertacao-v2.pdf`.

SILVA, Marco Aurélio Graciotto. *Revisão sistemática sobre padrões pedagógicos*. São Carlos, SP, Brasil, nov. 2007. 25 p. Disponível em: `http://www.labes.icmc.usp.br/~magsilva/publications/GraciottoSilva-SysRev-PPP.pdf`.

SILVA, Marco Aurélio Graciotto; BARBOSA, Ellen Francine; MALDONADO, José Carlos. *Educação a Distância na PUCRS*. São Carlos, SP, Brasil, jun. 2010. 38 p. Disponível em: `http://www.icmc.usp.br/~magsilva/publications/GraciottoSilva-PUCRS-2010.pdf`.

SILVA, Marco Aurélio Graciotto; BARBOSA, Ellen Francine; MALDONADO, José Carlos. Model-driven development of learning objects. In: *41st ASEE/IEEE Frontiers in Education Conference*. [S.l.]: IEEE, 2011. (FIE), p. F4E–1 – F4E6. ISBN 978-1-61284-467-1. ISSN 0190-5848.

SILVA, Marco Aurélio Graciotto; BARBOSA, Ellen Francine; MALDONADO, José Carlos. *Movie support for PDF using LaTeX*. São Carlos, SP, Brasil, nov. 2011. 31 p. Disponível em: `http://www.labes.icmc.usp.br/~magsilva/publications/GraciottoSilva-LatexMovie.pdf`.

SILVA, Marco Aurélio Graciotto; BARBOSA, Ellen Francine; MALDONADO, José Carlos. *Revisão sistemática sobre interatividade em objetos de aprendizagem*. São Carlos, SP, Brasil, maio 2011. Disponível em: `http://www.icmc.usp.br/~magsilva/publications/GraciottoSilva-SysRev-ILO-2011.pdf`.

SILVA, Marco Aurélio Graciotto; BARBOSA, Ellen Francine; MALDONADO, José Carlos. *Screencast recording*. São Carlos, SP, Brasil, 2011. Disponível em: `http://www.labes.icmc.usp.br/~magsilva/publications/GraciottoSilva-ScreencastRecording.pdf`.

SILVA, Marco Aurélio Graciotto; BARBOSA, Ellen Francine; MALDONADO, José Carlos. *Systematic review on critical factors for distance learning and e-learning (distance e-learning)*.

São Carlos, SP, Brasil, nov. 2011. 25 p. Disponível em: http://www.labes.icmc.usp.br/~magsilva/publications/GraciottoSilva-SysRev-ElearningCriticalFactors.pdf.

SIMPSON, Elizabeth Jane. *The Classification of Educational Objectives, Psychomotor Domain*. Urbana, Illinois, EUA, 1966. 35 p.

SIMS, Roderick C.; JONES, D. Where practice informs theory: Reshaping instructional design for academic communities of practice in online teaching and learning. *Information Technology, Education and Society*, James Nicholas, v. 4, n. 1, p. 3–20, 2003. ISSN 1037-616X.

SKINNER, B. F. The science of learning and the art of teaching. *Harvard Educational Review*, Harvard Education, v. 24, p. 86–97, 1954. ISSN 1943-5045, 0017-8055.

SMITH, Mark K. *Introducing informal education*. 2011. Encyclopaedia of informal education. Disponível em: http://www.infed.org/i-intro.htm. Acesso em: 20 fev 2012.

SOARES, Luiz Fernando Gomes; OTHERS. MAESTRO: The declarative middleware proposal for the SBTVD. In: *EuroITV 2006 - Beyond Usability, Broadcast, and TV*. Athens, Grécia: ACM, 2006. (EuroITV), p. 1–4.

SOARES, Luiz Fernando Gomes; RODRIGUES, Rogério Ferreira; MORENO, Marcio Ferreira. Ginga-NCL: the declarative environment of the Brazilian digital TV system. *Journal of the Brazilian Computer Society*, SBC, Porto Alegre, RS, Brasil, v. 12, n. 4, p. 37–46, mar. 2007. ISSN 0104-6500.

SOUZA, Maria de Fátima C. de; CASTRO-FILHO, José A.; ANDRADE, Rossana M. C. Model-driven development in the production of customizable learning objects. In: IEEE COMPUTER SOCIETY. *International Conference on Advanced Learning Technologies*. Washington, DC, EUA, 2010. p. 701 –702.

SOUZA, Maria de Fátima C. de; ANDRADE, Rossana M. C.; FILHO, José Aires de Castro. Aplicando engenharia de software orientado a modelos ao desenvolvimento de objetos de aprendizagem. In: *Workshop de Teses e Dissertações em Engenharia de Software – Simpósio Brasileiro de Engenharia de Software*. Campinas, SP, Brasil: SBC, 2008. p. 62–69.

SOUZA, Maria de Fátima Costa de; CASTRO-FILHO, Jose Aires; ANDRADE, Rossana Maria de Castro. Applying model-driven development for building customizable learning objects. *Multidisciplinary Engineering Education Magazine*, IEEE Education Society Students Activities Committee (EdSocSAC), United States, v. 6, n. 1, p. 22–30, mar. 2011. ISSN 1558-7908. This journal has been renamed to 'Technology and Engineering Education'. Disponível em: http://www.ewh.ieee.org/soc/e/sac/meem/index.php/meem/article/view/135/156.

SOUZA, Maria de Fátima Costa de; GOMES, Danielo G.; BARROSO, Giovanni Cordeiro; SOUZA, Cidcley Teixeira de; FILHO, José Aires de Castro; PEQUENO, Mauro Cavalcante; ANDRADE, Rossana. LOCPN: Redes de Petri coloridas na produção de objetos de aprendizagem. *Revista Brasileira de Informática na Educação*, CEIE-SBC, v. 15, n. 3, p. 39–52, set. 2007. ISSN 1414-5685. Disponível em: `http://bibliotecadigital.sbc.org.br/download.php?paper=1024`.

SOUZA, Maria Fátima C. de; FILHO, J. Aires de Castro; ANDRADE, Rossana M. C. Expertdsl: um perfil uml para o suporte à definição de escopo pedagógico em um processo de desenvolvimento de objetos de aprendizagem orientados a modelo. In: *Simpósio Brasileiro de Informática na Educação*. Fortaleza, CE, Brasil: SBC, 2008. p. 1–4. Poster.

STAHL, Thomas; VöLTER, Markus. *Model-Driven Software Development*. 1. ed. Great Britain: Wiley, 2006. 428 p. Título original: Modellgetriebene Softwareentwicklung. Traduzido por Bettina von Stockfleth. ISBN 978-0-470-02570-3.

STOTTS, P. David; FURUTA, Richard. Petri-net-based hypertext: document structure with browsing semantics. *ACM Transactions on Information Systems*, ACM, New York, NY, EUA, v. 7, n. 1, p. 3–29, jan. 1989. ISSN 1046-8188.

SU, Jun-Ming; TSENG, Shian-Shyong; CHEN, Chia-Yu; WENG, Jui-Feng; TSAI, Wen-Nung. Constructing SCORM compliant course based on high-level Petri nets. *Computer Standards & Interfaces*, Elsevier, Amsterdam, Países Baixos, v. 28, n. 3, p. 336–355, jan. 2006. ISSN 0920-5489.

SUGETA, Tatiana. *Proteum-RS/ST : Uma Ferramenta para Apoiar a Validação de Especificações Statecharts Baseada na Análise de Mutantes*. Dissertação (Mestrado) — ICMC-USP, São Carlos, SP, Brasil, nov. 1999. Orientador: José Carlos Maldonado.

TANTAU, Till; FEUERSAENGER, Christian; OTHERS. *PGF and TikZ*. ago. 2003. Programa de computador. Disponível em: `http://sourceforge.net/projects/pgf/`. Acesso em: 20 fev 2012.

TANTAU, Till; WRIGHT, Joseph; MILETIç, Vedran; OTHERS. *LaTeX Beamer*. 2003. Programa de computador. Disponível em: `http://bitbucket.org/rivanvx/beamer/wiki/Home`. Acesso em: 20 fev 2012.

TERRELL, Robert L.; CAUDILL, Jason G. OpenCourseWare: open sharing of course content and design. *Journal of Computing Sciences in Colleges*, Consortium for Computing Sciences in Colleges, EUA, v. 27, n. 3, p. 38–42, jan. 2012. ISSN 1937-4771.

TING, T. C.; JENSEN, A. P. An on-line interactive audiographic learning system. In: *ACM Annual Conference*. New York, NY, United States: ACM, 1974. p. 239–244.

TURINE, Marcelo Augusto Santos; OLIVEIRA, Maria Cristina Ferreira de; MASIERO, Paulo Cesar. A navigation-oriented hypertext model based on statecharts. In: *ACM Conference on Hypertext*. New York, NY, EUA: ACM, 1997. p. 241–256. ISBN 0-89791-866-5.

Final report of the Forum on the Impact of Open Courseware for Higher Education in Developing Countries. 28 p.

VEGA-OLIVEROS, Didier Augusto; PEDROSA, Diogo de Carvalho; PIMENTEL, Maria da Graça Campos; FORTES, Renata Pontin de Mattos. An approach based on multiple text input modes for interactive digital TV applications. In: ANACLETO, Junia C.; FORTES, Renata Pontin de M.; COSTA, Carlos J. (Ed.). *28th International Conference on Design of Communication*. São Carlos, SP, Brasil: ACM, 2010. (SIGDOC), p. 191–198. ISBN 978-1-4503-0403-0.

VINCENZI, Auri; DELAMARO, Márcio; HöHN, Erika; MALDONADO, José Carlos. Functional, control and data flow, and mutation testing: Theory and practice. In: BORBA, Paulo; CAVALCANTI, Ana; SAMPAIO, Augusto; WOODCOOK, Jim (Ed.). *Testing Techniques in Software Engineering*. Berlin, Alemanha: Springer, 2010, (Lecture Notes in Computer Science, v. 6153). cap. 2, p. 18–58. ISBN 978-3-642-14334-2.

VINCENZI, Auri Marcelo Rizzo; DELAMARO, Márcio Eduardo; SILVA, Marco Aurélio Graciotto; MALDONADO, José Carlos. *JaBUTi – Java Bytecode Understanding and Testing – User's manual*. São Carlos, SP, Brasil, set 2010. 107 p. Disponível em: `http://www.labes.icmc.usp.br/~magsilva/publications/GraciottoSilva-JaBUTi.pdf`.

VINCENZI, A. M. R.; WONG, W. E.; DELAMARO, M. E.; MALDONADO, J. C. JaBUTi: A coverage analysis tool for java programs. In: *Sessão de Ferramentas do Simpósio Brasileiro de Engenharia de Software (SBES)*. Manaus, AM, Brasil: SBC, 2003. p. 79–84. Disponível em: `http://ccsl.icmc.usp.br/projects/jabuti-0`.

WAGNER, Ellen D. Steps to creating a content strategy for your organization. *Learning Solutions e-Magazine*, eLearning Guild, p. 1–9, out. 2002.

WILEY, David. Openness as catalyst for an educational reformation. *EDUCAUSE Review*, EDUCAUSE, United States, v. 45, n. 4, p. 14–20, jul.-ago. 2010. ISSN 1945-709X, 1527-6619.

WILEY, David A. Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In: WILEY, David A. (Ed.). *The Instructional Use of Learning Objects*. 1. ed. United States: AIT/AECT, 2000. cap. 1.1, p. 1–35. ISBN 0-7842-0892-1, 978-0784208922. Disponível em: `http://reusability.org/read/chapters/wiley.doc`.

YU, Lu; CHEN, Sijia; WANG, Jianpeng. Overview of AVS-video coding standards. *Image Communication*, Elsevier, New York, NY, EUA, v. 24, n. 4, p. 247–262, abr. 2009. ISSN 0923-5965.

ZANTING, Anneke; VERLOOP, Nico; VERMUNT, Jan D. Using interviews and concept maps to access mentor teachers' practical knowledge. *Higher education*, Kluwer Academic, v. 46, n. 2, p. 195–214, 2003.

ZENG, Fanping; MAO, Liangliang; CHEN, Zhide; CAO, Qing. Mutation-based testing of integer overflow vulnerabilities. In: *5th International Conference on Wireless Communications, Networking and Mobile Computing*. Beijing, China: IEEE Communications Society, 2009. p. 1–4.