

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

FREDERICO MENINE SCHAF

**ARQUITETURA MODULAR PARA
AMBIENTES VIRTUAIS DE ENSINO DE
AUTOMAÇÃO COM SUPORTE À
REALIDADE MISTA E COLABORAÇÃO**

Porto Alegre
2011

FREDERICO MENINE SCHAF

**ARQUITETURA MODULAR PARA
AMBIENTES VIRTUAIS DE ENSINO DE
AUTOMAÇÃO COM SUPORTE À
REALIDADE MISTA E COLABORAÇÃO**

Tese de doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Rio Grande do Sul como parte dos requisitos para a obtenção do título de Doutor em Engenharia Elétrica.

Área de concentração: Controle e Automação

ORIENTADOR: Prof. Dr. Carlos Eduardo Pereira

Porto Alegre
2011

FREDERICO MENINE SCHAF

**ARQUITETURA MODULAR PARA
AMBIENTES VIRTUAIS DE ENSINO DE
AUTOMAÇÃO COM SUPORTE À
REALIDADE MISTA E COLABORAÇÃO**

Esta tese foi julgada adequada para a obtenção do título de Doutor em Engenharia Elétrica e aprovada em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____
Prof. Dr. Carlos Eduardo Pereira, UFRGS
Doutor pela Universidade de Stuttgart – Alemanha

Banca Examinadora:

Prof. Dr. José Manoel Ferreira Martins, FEUP, Portugal
Doutor pela Universidade do Porto, Portugal

Prof. Dr. Alexandre de Assis Mota, PPGEE - PUC Campinas
Doutor pela Universidade Estadual de Campinas

Prof. Dr. Maria Cristina Villanova Biazus, PPGIE - UFRGS
Doutora pela Universidade Federal do Rio Grande do Sul

Prof. Dr. Walter Fetter Lages, PPGEE - UFRGS
Doutor pelo Instituto Tecnológico de Aeronáutica

Prof. Dr. Renato Ventura Bayan Henriques, PPGEE - UFRGS
Doutor pela Universidade Federal de Minas Gerais

Coordenador do PPGEE: _____
Prof. Dr. Alexandre Sanfelice Bazanella

Porto Alegre, Março de 2011.

DEDICATÓRIA

Dedico aos meus pais, à minha esposa, aos futuros filhos e aos educadores que porventura possam empregar algumas idéias e conceitos apresentados neste trabalho para aprimorar o ensino e/ou treinamento.

AGRADECIMENTOS

À minha esposa e família que sempre me apoiaram.

Ao meu pai que serviu de exemplo e sempre incentivou o aperfeiçoamento acadêmico.

Aos meus pais pela revisão ortográfica deste manuscrito e aconselhamento mesmo que na visão leiga do trabalho.

Ao Programa de Pós-Graduação em Engenharia Elétrica, PPGEE, pela oportunidade de realização de trabalhos em minha área de pesquisa.

Agradeço à CAPES pelo provimento da bolsa de estudos de doutorado no país e no exterior e também ao CNPq e ao DAAD alemão por financiamentos parciais eventuais.

Agradeço também à parceria frutífera dos parceiros do consórcio internacional Europa-América Latina RExNet financiado pela Comunidade Européia e ao SENAI por financiar um projeto no início de meus estudos.

Agradeço pessoalmente: ao meu orientador Carlos Eduardo Pereira; ao pesquisador alemão que idealizou uma parceria mais próxima entre Brasil e Alemanha, Prof. Dr. Heinz Erbe; e aos orientadores alemães Dr. Dieter Müller, e Prof. Dr. Wilhelm Bruns do *ArtecLab* da Universidade de Bremen.

Aos colegas do PPGEE e bolsistas de iniciação científica pelo seu auxílio nas tarefas e projetos desenvolvidos durante o doutorado.

Aos parceiros em projetos de pesquisa afins que adicionaram tanto aplicações práticas quanto diversas implementações secundárias em harmonia com este trabalho.

Aos professores componentes das diversas bancas: de seminários, proposta de tese, e outros, onde parte deste trabalho foi apresentado e devidamente debatido.

RESUMO

Este trabalho apresentará estudos referentes a uma proposta de arquitetura para ambientes computacionais de suporte à colaboração para auxiliar o ensino e o treinamento em áreas multidisciplinares de engenharia de controle e automação. A motivação provém de evoluções de tecnologias da *Web*, redes sociais, ambientes virtuais, e mundos virtuais 3D que proporcionam sucessivas possibilidades inovadoras de emprego de tecnologia no ensino. Estas tecnologias podem ser usadas no ensino à distância ou de forma complementar (ensino híbrido) ao ensino-aprendizagem tradicional. O emprego de ambientes computacionais virtuais como incubadores para o desenvolvimento e criação de uma infinidade de materiais educacionais e didáticos torna-se crescentemente frequente na comunidade científica ao passo que instituições de ensino percebem o potencial desta iniciativa. A colaboração virtual entre estudantes dispersos geograficamente e a cooperação entre instituições de ensino diversas são indicadores do avanço destas metodologias. À medida que ambientes virtuais evoluem, podem-se empregar cada vez mais funcionalidades e tecnologias a este complemento ou ferramenta de ensino. É sabido que a prática laboratorial é de suma importância para aprendizagem de áreas tecnológicas, como a engenharia elétrica. Maneiras de expandir esta prática aliada a outras técnicas altamente defendidas na comunidade científica serão abordadas neste estudo, que sucede a um trabalho anterior do autor, no qual é proposto a conjunção de componentes de automação simulados com reais de forma a expandir cenários didáticos. A automação destes ambientes voltados ao ensino passa a se tornar real através de técnicas computacionais relacionadas a gerenciamento, monitoração e adaptação de conteúdo. Dentre as tecnologias e características englobadas pela arquitetura proposta estão: ambientes sociais, ambientes imersivos com visualização tridimensional (*metaversos*), tutoriamento virtual autônomo baseado em coleta de informações de interação, auxílio à colaboração de usuários, experimentos com componentes intercambiáveis de realidade mista e outras funcionalidades afins. De forma a propor uma implementação viável para este nicho de pesquisa foram desenvolvidos vários estudos de caso e um protótipo, chamado de *3D AutoSysLab*, um ambiente imersivo virtual otimizado com ligações a experimentos de realidade mista, visualização de materiais educacionais e mídias interativas, auxílio à colaboração, e suporte à tutoriamento autônomo. Tanto testes do protótipo quanto a utilização dos estudos de caso serviram de validação experimental do trabalho. A validação da utilização ainda não se encontra completa embora resultados preliminares apontem a queda do índice de reprovação de estudantes da Engenharia Elétrica na UFRGS em uma determinada unidade curricular. O emprego de técnicas de acesso remoto além de proporcionar que instituições de ensino cooperem compartilhando recursos materiais/equipamentos também abre horizontes para a colaboração de estudantes entre instituições. Isso alavanca uma melhoria na qualidade do conjunto ensino-aprendizagem e também no emprego de recursos tecnológicos, o que pode, inclusive, representar uma alternativa econômica viável para a prática laboratorial.

Palavras-chave: Tecnologias da Educação, Ambientes Virtuais de Aprendizagem, Aprendizado/Trabalho Colaborativo/Cooperativo Assistido por Computador (CSCL/CSCW), Ensino Aberto ou Híbrido, Experimentação Remota, Realidade Mista, Sistemas Multiagente, Sistemas de Tutoriamento Autônomos.

ABSTRACT

This work will present studies towards a proposal of architecture for computer supported collaborative environments applied to enhance learning and training of multidisciplinary areas linked to automation and control engineering. Recent evolutions of Web technologies, social networks, virtual environments, and 3D virtual worlds motivate the adoption of new technologies in education, opening successive innovative possibilities. These technologies or tools can be used in distance education scenarios and/or can also enhance traditional learning-teaching (blended or hybrid learning scenario). The use of virtual environments as breeding grounds for creation and display of learning and didactical materials is a trend among educational institutes since they realized the potential of this endeavor. Physically dispersed and geographically distant students can collaborate in virtual learning environments as well as educational institutions, proving the advance of learning methodologies involving technology. New functionalities to this teaching and learning complementary tool can be added as virtual environments continually evolve. It is known and a wide advocated issue that laboratorial practice is essential to technical education, foremost in engineering. Expanding and obtaining the optimal results from this practice allied to other tools can be seen as the natural continuation of previous work from the author, in which is described a strategy to use mixed reality interchangeable components applied in didactic experiments to broaden learning scenarios. The automation of virtual learning environments is beginning to mature through computational techniques related to management, monitoring and content adaptation. Incorporated to the architecture proposal are the topics/concepts of: social networks, immersive interfaces with tridimensional representation (metaverses), autonomous virtual tutoring based on data mining from user interactions logs, active collaboration support of students, experimentation with mixed reality and others. In order to develop a feasible implementation to this research area, several case studies and a prototype were elaborated, the last called *3D AutoSysLab*, in which a metaverse is used as interface, experiments are linked to virtual objects representations, learning objects are displayed in a simple way with interactive medias and simple collaboration and feedback is given through an autonomous tutoring system. The experimental validation was based on the use and evaluation of the prototype and his separated modules (case studies). Preliminary results showed that the failure rate, in a determined curriculum unit from the Electrical Engineering of UFRGS, has dropped and the student's acceptance was very high in a blended learning scenario. The remote experimentation brings up the possibility to narrow different learning institutions in cooperation, this way sharing material resources (equipments) and closing up students. From cooperation and collaboration is expected a leveling and optimization of quality of the pair teaching-learning, as well as a better employment of economical resources.

Keywords: Educational Technologies, Virtual Learning Environments (VLEs), CSCL, CSCW, Hybrid/Blended Learning, Remote Experiments, Multi-Agent Systems (MASs), Intelligent Tutoring Systems (ITSs).

SUMÁRIO

LISTA DE ILUSTRAÇÕES	13
LISTA DE TABELAS	17
LISTA DE ABREVIATURAS	19
1 INTRODUÇÃO	21
1.1 Motivação	21
1.2 Contextualização	23
1.3 Objetivos	25
1.4 Contribuições da Tese	25
1.5 Organização do Manuscrito	26
2 EMBASAMENTO TEÓRICO	27
2.1 Introdução	27
2.2 Ensino e Aprendizagem	27
2.3 Tecnologias de Educação	30
2.4 Ferramentas de Suporte à Colaboração	32
2.4.1 Classificação	33
2.5 Simulação e Modelo Virtuais	34
2.6 Realidade Mista	35
2.7 Ambiente e Espaço Virtuais	37
2.7.1 Colaboração e/ou Cooperação em Ambientes Virtuais	38
2.7.2 Ambientes Virtuais de Aprendizagem	38
2.7.3 Ambientes Virtuais de Trabalho	39
2.7.4 Ambientes Imersivos	40
2.8 Experimentação	41
2.9 Redes Sociais	44
2.10 Entretenimento <i>Online</i> (MMOGs)	44
2.11 <i>Serious Games</i>	45
2.12 Mundos Virtuais Tridimensionais	45
2.13 Sistemas Multiagente	45
3 ANÁLISE DO ESTADO DA ARTE	49
3.1 Introdução	49
3.2 Tecnologias na Educação	51
3.3 Ambientes Virtuais de Aprendizagem	53
3.4 Experimentação Virtual/Remota	54

3.4.1	<i>VCLab</i>	54
3.4.2	<i>SimQuest</i>	54
3.4.3	<i>Automatic Control Telelab</i>	56
3.4.4	<i>CoLab e ModellingSpace</i>	57
3.4.5	<i>HTI Solar Energy Laboratory</i>	59
3.4.6	<i>Mixed Reality Mechatronics Lab - DeriveSERVER e variantes</i>	60
3.5	Ambientes/Mundos Virtuais Imersivos	62
3.5.1	<i>Active Worlds</i>	63
3.5.2	<i>Open Wonderland</i>	64
3.5.3	<i>Second Life</i>	65
3.5.4	<i>OpenSim</i>	68
3.5.5	<i>RealXtend</i>	69
3.5.6	<i>3DXplorer</i>	71
3.5.7	Outros	72
3.6	<i>Serious Games</i>	72
3.7	Tecnologias de Colaboração	73
3.8	Tutoriameto Autônomo	74
3.9	Tecnologias de Rede e Distribuição de Recursos	75
3.10	Integração e Interoperabilidade de Ferramentas de <i>Software</i>	76
3.11	Análise e Comparações	78
4	PROPOSTA DE ARQUITETURA	83
4.1	Introdução	83
4.2	Conceitos e Características Relevantes Associados aos AVCs	86
4.3	Tecnologias e Ferramentas	88
4.4	Módulos	90
4.4.1	Módulo de Espaço Compartilhado de Mídias Educacionais Teóricas	90
4.4.2	Módulo de Interface Social 3D	91
4.4.3	Módulos Associados à Experimentação	93
4.4.4	Módulos Associados à Adaptação de Conteúdo e de <i>Feedback</i> ao Usuário	95
4.4.5	Repositório de Informações	100
4.4.6	<i>Middleware</i>	101
4.5	Arquitetura para AVCEAs	103
5	IMPLEMENTAÇÃO	105
5.1	Sinopse	105
5.2	Panorama das Ferramentas e Tecnologias	105
5.2.1	Ferramentas para Sistemas de Aprendizagem	106
5.2.2	Ferramentas para Componentes Intercambiáveis	107
5.2.3	Ferramentas para Sistemas Multiagente	108
5.2.4	Ferramentas para Adaptação de Conteúdo	109
5.2.5	Ferramentas para Tutoriameto	111
5.2.6	Ferramentas para Suporte à Colaboração	111
5.2.7	Interfaces Sociais Tridimensionais e Ambientes Imersivos	112
5.3	Estudos de Caso	113
5.3.1	Monitorador de Agendamento	113
5.3.2	<i>Metaversos</i> como AVCEAs	116
5.3.3	<i>Hardware-in-the-Loop</i> de Baixo Custo para Testes de Realidade Mista	119
5.3.4	Componentes intercambiáveis com <i>Processing</i> e <i>Arduino</i>	122

5.3.5	Sistema Tutor Autônomo Simples	124
5.3.6	Colaboração no AVCEA	128
5.4	Protótipo 3DAutoSysLab	129
6	VALIDAÇÃO	133
6.1	Avaliação da Conformidade dos Objetivos do Trabalho	133
6.2	Avaliação do Protótipo	134
6.3	Avaliação dos Usuários	136
7	CONCLUSÕES	139
	REFERÊNCIAS	143

LISTA DE ILUSTRAÇÕES

Figura 1:	Diagrama do Aprendizado em Ação - habilidades e ações.	29
Figura 2:	Modelagem e simulação - diagrama.	34
Figura 3:	Realidade mista - espectro (<i>continuum</i>).	36
Figura 4:	Realidade mista - exemplo de interconexão de processos reais e virtuais.	36
Figura 5:	Modelo 3Cs - diagrama de relações e classificação de sistemas colaborativos.	38
Figura 6:	AVAs - ferramentas mais comuns.	39
Figura 7:	<i>Porta-Person</i> e <i>Virtual Workplace</i> da Sun.	40
Figura 8:	Ambiente imersivo - exemplo com percepção visual e tátil.	41
Figura 9:	Experimentação remota - diagrama de estrutura.	43
Figura 10:	Colaboratórios - diagrama de estrutura.	43
Figura 11:	Sistemas multiagente - diagrama de estrutura.	47
Figura 12:	<i>VCLab</i> - modelos <i>VRML</i> de experimentos diversos.	55
Figura 13:	<i>SimQuest</i> - interface com exercício didático proposto ao estudante.	55
Figura 14:	<i>ACT</i> - interface e experimentos diversos.	57
Figura 15:	<i>CoLab</i> - interface do ambiente.	58
Figura 16:	<i>ModellingSpace</i> - interface do ambiente.	59
Figura 17:	<i>HTI Solar Energy Laboratory</i> - experimento e diagrama didático de funcionamento.	60
Figura 18:	<i>DeriveServer</i> - interface e bancada montada no <i>ArtecLab</i> e no SENAI de Caxias do Sul.	61
Figura 19:	<i>Hyper-bond</i> do <i>deriveServer</i> - diagrama de interconexão, <i>hardware</i> e <i>software</i>	61
Figura 20:	<i>CAVE</i> - exemplo de colaboração na experimentação e configuração multi- <i>CAVE</i> de espaços de trabalho distribuídos.	62
Figura 21:	<i>AWEDU</i> - interface do usuário no mundo virtual <i>iGarden</i>	63
Figura 22:	<i>3D Physics Lab</i> - uso do <i>Open Wonderland</i> como ambiente educacional.	65
Figura 23:	<i>SLOODLE</i> - encontro no <i>VirtuALBA</i> e <i>SLOODLE Moot</i>	66
Figura 24:	<i>SLOODLE</i> - diagrama do funcionamento e de trocas de informações.	67
Figura 25:	<i>Magee Campus</i> no <i>SL</i> - exemplos de simulações disponíveis.	67
Figura 26:	<i>Wiimote</i> e experimento híbrido de máquina de lavar.	68
Figura 27:	<i>Second Lab</i> da Universidade de Deusto - interface e experimento.	68
Figura 28:	<i>RealXtend</i> - interface com gerador de <i>avatar</i> , importação de <i>meshes</i> e o <i>Naali viewer</i>	70
Figura 29:	<i>3DXplorer</i> - interface.	71
Figura 30:	<i>JDoc</i> - interface do jogo.	72

Figura 31:	<i>ECOSPACE</i> - diagrama organizacional.	74
Figura 32:	<i>ITS</i> simples - emprego do <i>Framework</i> para SMA <i>JADE</i> em associação com o AVA <i>MOODLE</i>	75
Figura 33:	<i>Access Grid</i> - nodo e distribuição dos nodos.	76
Figura 34:	<i>CSGrid</i> - interface do espaço de trabalho e arquitetura.	76
Figura 35:	<i>Gaia</i> - espaço ativo experimental e arquitetura.	78
Figura 36:	Cronologia de trabalhos relacionados a AVCs.	79
Figura 37:	Ambientes imersivos - exemplo de comparação com ambiente tradicional.	87
Figura 38:	Grupos de características identificadas para AVCs.	89
Figura 39:	Módulo de espaço compartilhado de mídias educacionais teóricas - diagrama.	91
Figura 40:	Módulo de interface tridimensional social - diagrama de idealização.	92
Figura 41:	Componentes intercambiáveis - diagrama de cenários possíveis.	93
Figura 42:	Módulos do grupo de experimentação - diagrama de idealização.	95
Figura 43:	Adaptação de conteúdo - exemplo para quatro usuários distintos.	96
Figura 44:	Módulo dos monitores de interações - exemplo da coleta de informações e organização de perfis para adaptação de conteúdos.	98
Figura 45:	Módulo dos tutores autônomos - diagrama sequência <i>UML</i> para as diversas tarefas e agentes envolvidos.	99
Figura 46:	Módulos do grupo de adaptação de conteúdo e de <i>feedback</i> aos usuários - diagrama e interações.	100
Figura 47:	Repositório de Informações - exemplo de distribuição e interconexão.	101
Figura 48:	<i>Middleware</i> da arquitetura - exemplo de funcionamento com conexões.	101
Figura 49:	<i>Middleware</i> de conexão - exemplo de interações diretas e indiretas.	102
Figura 50:	Proposta de arquitetura para AVCEAs - diagrama de módulos e conectividade.	104
Figura 51:	<i>MOODLE</i> - sistemas de aprendizado em uso na UFRGS.	107
Figura 52:	Implementação dos componentes intercambiáveis - diagrama de comunicação.	108
Figura 53:	<i>JADE</i> - <i>GUI</i> e funcionalidade <i>sniffer</i>	109
Figura 54:	<i>JADE</i> - arquitetura de comunicação básica.	110
Figura 55:	Sistema de agendamento - interface integrada no <i>MOODLE</i>	114
Figura 56:	Estudo de caso: monitorador de agendamento de experimento - diagrama de implementação.	115
Figura 57:	Estudo de caso: monitorador de agendamento de experimento - <i>GUI</i> do <i>JADE</i> e interação de agentes.	117
Figura 58:	Estudo de caso: interface social 3D.	117
Figura 59:	<i>OpenSim</i> - ferramenta de edição de objetos do <i>metaverso</i>	118
Figura 60:	Pacote <i>SLOODLE</i> - teste de uso do no <i>OpenSim</i>	120
Figura 61:	Planta de envase virtual - interface no <i>RealXtend Viewer</i>	121
Figura 62:	Estudo de caso: diagrama de <i>HiL</i> com o <i>Arduino</i>	122
Figura 63:	Estudo de caso: <i>HiL</i> com o <i>Arduino</i> e a planta de envase virtual no <i>OpenSim</i>	123
Figura 64:	Estudo de caso: <i>HiL</i> com o <i>Arduino</i> e a planta de envase virtual no <i>Processing</i>	124
Figura 65:	Estudo de caso: diagrama do sistema tutor autônomo genérico.	125

Figura 66:	Estudo de caso: bloco de tutoriamento integrado ao <i>MOODLE</i>	127
Figura 67:	Estudo de caso: representação do sistema tutor (<i>bot</i>) no <i>OpenSim</i>	128
Figura 68:	Protótipo - arquitetura de implementação com ferramentas e interfaces de comunicação.	129
Figura 69:	Protótipo - diagrama de implementação.	130
Figura 70:	Protótipo <i>3DAutoSysLab</i> - experimentos, colaboração, visualização, salas virtuais, etc.	130
Figura 71:	Protótipo <i>3DAutoSysLab</i> - interações de usuários.	137

LISTA DE TABELAS

Tabela 1:	Classificação de laboratórios.	41
Tabela 2:	Comparação dos AVCs apresentados sem pretensões educacionais explícitas.	80
Tabela 3:	Comparação dos AVCs apresentados com pretensões educacionais explícitas.	81
Tabela 4:	Informações dos sensores e atuadores da planta de envase virtual. . .	121
Tabela 5:	Resultados da avaliação TICESE.	135
Tabela 6:	Comparação de implementações de ambientes virtuais voltados ao ensino-aprendizagem com a arquitetura e o protótipo proposto. . . .	136
Tabela 7:	Resultado da pesquisa de opinião dos estudantes do GCAR-EAD. . .	138

LISTA DE ABREVIATURAS

API	<i>Application Programming Interface</i>
AVA	Ambiente Virtual de Aprendizagem
AVC	Ambiente Virtual Colaborativo
AVCEA	Ambiente Virtual Colaborativo de Ensino-Aprendizagem
BSD	<i>Berkeley Software Distribution</i>
CBL	<i>Computer Based Learning</i>
CAEE	<i>Computer Aided Engineering Education</i>
CAS	<i>Collaboration Aware System</i>
CBT	<i>Computer Based Training</i>
CMC	<i>Computer Mediated Communication</i>
CSC	<i>Computer Supported Collaboration/Cooperation</i>
CSCE	<i>Computer Supported Collaborative Environment</i>
CSCL	<i>Computer Support Collaborative Learning</i>
CSCW	<i>Computer Support Cooperative Work</i>
EaD	Educação à Distância
FOSS	<i>Free Open Source Software</i>
GCAR	Grupo de Controle, Automação e Robótica
GPL	<i>General (or GNU) Public License</i>
GUI	<i>Graphical User Interface</i>
HiL	<i>Hardware-in-the-Loop</i>
HTML	<i>Hypertext Mark-up Language</i>
HTTP	<i>Hypertext Transfer Protocol</i>
IBL	<i>Inquiry-Based Learning</i>
IMS	<i>Instructional Management System</i>
ITS	<i>Intelligent Tutoring Systems</i>
JVM	<i>Java Virtual Machine</i>

JRE	<i>Java Runtime Engine</i>
LO	<i>Learning Object</i>
LOM	<i>Learning Object Metadata</i>
MAS	<i>Multi-Agent System</i>
MMOG	<i>Massive Multiplayer Online Game</i>
MRCSE	<i>Mixed Reality Collaborative Social Environment</i>
MUVE	<i>Multi-User Virtual Environment</i>
NTIC	Novas Tecnologias da Informação e Comunicação
OLE	<i>Object Linking and Embedding</i>
OO	<i>Object Oriented</i>
OPC	<i>OLE Process Control</i>
OS	<i>Operating System</i>
P2P	<i>Peer-to-Peer</i>
PBL	<i>Problem-Based Learning</i>
PHP	<i>Perl Hypertext Pre-processor</i>
PID	<i>Proportional-Integral-Derivative</i>
PPGEE	Programa de Pós-Graduação em Engenharia Elétrica
RPC	<i>Remote Procedure Call</i>
RMI	<i>Remote Method Invocation</i>
RTSP	<i>Real-Time Streaming Protocol</i>
SBBT	<i>Second Best of Being There</i>
SCADA	<i>Supervisory Control and Data Acquisition</i>
SCORM	<i>Sharable Content Object Reference Model</i>
SMA	Sistemas Multiagente
SiL	<i>Software-in-the-Loop</i>
SL	<i>Second Life</i>
TIC	Tecnologias de Informação e Comunicação
TICESE	Técnica de Inspeção Conformidade Ergonômica de <i>Software</i> Educacional
UI	<i>User Interface</i>
UML	<i>Unified Modeling Language</i>
URL	<i>Uniform Resource Locator</i>
VRML	<i>Virtual Model Modeling Language</i>
WS	<i>Web Service</i>
WWW	<i>World Wide Web</i>
XML	<i>Extensible Markup Language</i>

1 INTRODUÇÃO

1.1 Motivação

Há diversos desafios na educação tecnológica com relação ao ensino-aprendizagem. Entre eles podem-se citar: a acelerada evolução tecnológica, a necessidade do estreitamento entre teoria e prática, a motivação dos estudantes, e o desenvolvimento de habilidades não técnicas. Outros desafios ainda referentes a demandas sociais, como: diminuição e corte de postos de trabalho, risco tecnológico, e ainda reflexos dos problemas do ensino médio e fundamental. Assim como outras áreas, o ensino de engenharia de controle e automação se depara igualmente com estes mesmos obstáculos, somados ainda alguns inerentes à formação, tais como: a dificuldade na integração multidisciplinar; uma área de atuação cujas bases epistemológicas são voltadas para a abstração; e a necessidade de uma sólida base matemática para o desenvolvimento de metodologias.

Ainda que algumas destas dificuldades possam ser atenuadas pelo uso de tecnologia na educação, este ainda é um tema onde há muita controvérsia entre os educadores. Hoje em dia é absolutamente imprescindível a utilização de computadores em qualquer área de pesquisa, sobretudo na informática e nas engenharias. Linhas de pesquisa como controle e automação se baseiam principalmente em ferramentas computacionais para o desenvolvimento de sistemas e novas e melhores tecnologias, ferramentas, conceitos e metodologias.

Entre as diversas ferramentas de NTICs (Novas Tecnologias da Informação e Comunicação) que podem ser aplicadas na educação ou no ensino-aprendizagem de engenharia elétrica destacam-se os ambientes computacionais virtuais (digitais) e a experimentação (prática) através de computadores, justamente por estes suprirem algumas soluções para problemas sempre frequentes em instituições de ensino que não dispõem de grande infraestrutura laboratorial e de assistência à colaboração entre estudantes. O emprego de ambientes virtuais, em evolução em diversas instituições, tem sido encorajado tanto pelas universidades e outras instituições educacionais e de treinamento, quanto pelo próprio Ministério da Educação (MEC) brasileiro através da Secretaria de Ensino à Distância (SEED). Estes, quando empregados ao ensino, são chamados de ambientes virtuais de aprendizagem (AVAs).

A consolidação da Internet, disponível crescentemente a mais estudantes e com sua capacidade de interconectar nodos tecnológicos dispersos geograficamente, trouxe consigo novos paradigmas que promoveram bases tecnológicas para a solução de um antigo problema - inerente não só ao ensino tradicional, mas também da Educação à Distância (EaD) - a experimentação, através do emprego de práticas laboratoriais remotas. Estas práticas são alternativas (ou complementos) para atender às demandas crescentes por laboratórios (COOPER, 2000) e por vagas em instituições de ensino superior. Experimentos

remotos, baseados na *Web*, possuem diversos aspectos positivos para a implantação, pois utilizam frequentemente equipamentos industriais de alto custo comercial que, por sua vez, requerem uma área específica (muitas vezes grande) para a instalação. Este tipo de experimento é acessível a um maior número de estudantes, pois possui tanto flexibilidade espacial (estudantes podem estar fora da instituição de ensino), quanto temporal (os experimentos podem ser acessados 24 horas por dia, nos 7 dias na semana).

A discussão das bases do ensino laboratorial na área de engenharias torna imperativa uma análise de impacto e das consequências dos ambientes auxiliados por computador, bem como os modernos ambientes digitais de simulação, com seus desdobramentos não somente sobre a metodologia de projeto e análise dos sistemas de controle e automação, mas também sobre os procedimentos pedagógicos atuais. Não se concebe mais o aprendizado tecnológico sem a interveniência crescente dos ambientes digitais, em suas múltiplas variantes. Sua utilização como alternativa ou mesmo complemento aos laboratórios físicos, geralmente caros e com manutenção complexa, integra obrigatoriamente os currículos modernos, principalmente pela capacidade de disponibilizar ferramentas de visualização e animação singulares, que auxiliam na representação e apreensão dos abstratos conceitos subjacentes ao ensino do controle. Esta discussão está intimamente ligada ao termo treinamento auxiliado por computador (*CBT - Computer Based Training*¹) desde que computadores estão presentes nas instituições de ensino.

O uso de computadores propicia a possibilidade de se usarem modelos computacionais que ilustrem e simulem práticas reais para os estudantes. Apesar da simulação de sistemas ser de extrema importância em fases iniciais do aprendizado, ela não substitui equipamentos físicos, pois é uma necessidade imperativa que os estudantes sejam expostos aos “caprichos e perversidades” do mundo real (KHEIR et al., 1996). Estas características essenciais à formação plena dos engenheiros, e que sistemas físicos apresentam, incorporam fatores como o fato de constituírem estruturas não ideais, com a ocorrência de fenômenos de histerese e fricção; a presença de ruídos e não linearidades com diferentes níveis de complexidade; a existência de limitações físicas, como por exemplo, faixa limitada de deslocamento; a importância de uma montagem física correta, mesmo para elementos simples como cabos e conectores; a necessidade de proteção contra possíveis defeitos de funcionamento e, finalmente, a imposição de especificar interfaces apropriadas para os diferentes componentes garantindo um sistema operacionalmente correto (KOKU; KAYNAK, 2001).

De forma a atingir níveis mais altos de interação entre estudantes (ou usuários) conceitos de colaboração são necessários. A colaboração busca o trabalho em grupo focando um objetivo comum, podendo os membros estar em tempos e locais diferentes (flexibilidade temporal e espacial), em diferentes institutos de ensino/treinamento (relações inter instituições/ companhias) e em diferentes funções (trabalho multidisciplinar). Os princípios são: oferecer meios de colaboração; suportar participação flexível de grupos (compartilhando ideias); focar em processos, comunicação e relacionamentos utilizando a tecnologia. Embora *softwares* possam ser desenvolvidos para alcançar maior colaboração entre usuários também se observa que usuários destes *softwares* necessitam criar relações sociais antes de colaborarem ou terem trabalhos em co-autoria. Desta maneira, para se atingir a colaboração também é necessário oferecer suporte para interações sociais. A maioria dos AVAs usados em cursos de ensino à distância empregam conceitos de colaboração, como: ensino em grupo e distribuído - *team learning* e *distributed learning* (AUER

¹Existem diversas variantes do *CBT* e muitos outros termos semelhantes, como: *WBT - Web-based Training*, *IBT - Internet-based Training* e *e-Learning*.

et al., 2003); e ensino autodidático (ou ativo). A aplicação e desenvolvimento apropriado de ferramentas e materiais de ensino que suporte o ensino tanto remoto quanto presencial torna possível e praticável o casamento perfeito dessas tecnologias educacionalmente.

Embora tecnologias na educação sejam difíceis de ser empregadas naturalmente por educadores sem treinamento, é importante frisar que a dinâmica lenta da evolução de técnicas educacionais (NEALE; CARROLL; ROSSON, 2004) não impacta na crescente evolução de ferramentas educacionais com emprego de tecnologias do estado da arte. Esta tendência pode ser atestada neste trabalho onde pesquisas e desenvolvimentos da comunidade científica estão presentes em todo o mundo. Apesar do aparente retardo evolutivo do ensino brasileiro - alguns países europeus e os EUA são mais propensos a mudanças para novas metodologias e conceitos educacionais - este trabalho incentiva tanto a utilização quanto apresenta a tendência contextual da existência inevitável de tecnologia nas diversas esferas organizacionais da sociedade, inclusive no ensino. Este estado de “marasmo” dificulta a avaliação e análise de resultados de utilização da proposta. Mesmo assim, diversos estudos já apontam para a necessidade de uma melhora e padronização de tal avaliação tanto quanto as causas da estagnação evolutiva (NEALE; CARROLL; ROSSON, 2004).

1.2 Contextualização

Esta tese pode ser vista como evolução natural que sucede a pesquisa e o trabalho anterior do autor (dissertação de mestrado) na mesma área de aplicação, onde foi investigada uma abordagem para experimentos de realidade mista associados a AVAs (SCHAF, 2006).

Há atualmente diversas implementações de instituições de ensino que usam a prática remota com experimentos acessíveis via Internet para enriquecer o EaD e/ou ensino tradicional. Segundo (ATKAN et al., 1996), esta opção pode ser chamada de “segunda melhor alternativa de estar lá” (*SBBT - Second Best of Being There*) e se tornou uma solução economicamente atrativa para experiências práticas empregadas didaticamente em instituições de ensino. Seguindo esta tendência, muitas e mais variadas áreas empregam experimentos remotos, entre elas pode-se citar: controle de processos digitais (RAMAKRISHNAN et al., 2000), aplicações aeroespaciais, controle *PID* (BATUR et al., 2000; ZEILMANN et al., 2003), controle preditivo, sistemas embarcados de comunicação (SCHMID, 2000), controle supervisorio (LEE; HSU, 2003), teleoperação de robôs e outros sistemas (HUIJUN; AIGUO, 2007), e aplicações tempo-real para vídeo e áudio. A maioria destes experimentos utiliza dispositivos/equipamentos e *softwares* personalizados para criar experimentos remotos didáticos de pequena escala.

Experiências anteriores mostraram que somente a disponibilização de experimentos remotos não é condição suficiente para garantir um melhor processo de aprendizado (SCHAF, 2006). Oferecer experimentos de laboratórios remotos sem associá-los a materiais didáticos em fase pré-laboratorial apropriados (explicando os tópicos educacionais que devem ser observados no experimento), normalmente leva estudantes a utilizar a estratégia de tentativa e erro (*trial and error strategy*), que tem um impacto educacional menor do que o inicialmente desejado para experimentação. Outro fator não presente em laboratórios tradicionais que se diferenciam dos laboratórios remotos é a disponibilidade, os anteriores estão disponíveis em tempo integral o que normalmente demanda um maior número de tutores que são necessários para guiar e providenciar material de ensino adequado.

De forma a harmonizar a tecnologia com o ensino tradicional muitas das soluções empregadas pelas instituições de ensino se baseiam no chamado *blended learning*, que é uma mescla do ensino tradicional com o apoio de ferramentas de *e-Learning*, tais como, AVAs com material de apoio didático associado a experimentos remotos. Esta “solução” possibilita não somente a manutenção dos cargos dos professores como pode oferecer enriquecimento tanto do conteúdo da disciplina quanto maior prática e uso de tecnologias atuais em prol do ensino. Para áreas técnicas como engenharias isto também apresenta uma vantagem motivacional aos estudantes que se deparam com ferramentas atuais de comunicação e de projetos colaborativos, muitas vezes, inclusive adotados pelos estudantes independente da instituição de ensino, por conta própria, para maior interação com estudantes e o mundo científico extrainstitucional.

O desenvolvimento de AVAs, acessíveis através da Internet, também possibilita que cursos atendam e estejam disponíveis a um número maior de estudantes em modalidades de *blended learning* e EaD. Os professores ligados a estes cursos nem sempre estão disponíveis, gerando a necessidade de desenvolvimento de ambientes virtuais de aprendizagem com certo grau de autonomia.

O emprego de tecnologias de colaboração, tanto para o ensino, quanto para o treinamento e o trabalho é pesquisado há bastante tempo na comunidade científica, e estes avanços são direcionados a áreas conhecidas como *CSCW* (*Computer Supported Cooperative Work*), que é o trabalho cooperativo/colaborativo assistido por computador(es), e *CSCL* (*Computer Supported Collaborative Learning*), que conta com a mesma assistência ou suporte voltada para o ensino (ou aprendizado).

O *CSCW* emergiu como um campo de pesquisa multidisciplinar envolvendo tecnólogos, economistas, psicólogos sociais, antropologistas, teóricos organizacionais, educadores, e qualquer um que pudesse desfrutar de atividades de grupo (GRUDIN, 1988, 1994). É de entendimento geral na área que usuários devem poder personalizar suas presenças e operações para poderem ser representados como indivíduos únicos e selecionar o meio e a maneira com que acessam e participam do ambiente (de ensino ou de trabalho). A nova “organização” deve ser descrita como uma série de objetivos compartilhados comuns. O objetivo de sistemas *CSCL* é proporcionar uma maneira de ensino efetiva entre estudantes que estudam “juntos”. Sistemas *CSCL* proporcionam troca de ideias e informações entre os “estudantes” que acessam informações e documentos colaborativamente. Instrutores ou tutores respondem a atividades facilitando a dinâmica e os processos de grupo de maneiras não possibilitadas por seu equivalente presencial.

A integração de laboratórios, tanto presenciais quanto remotos, no currículo de disciplinas técnicas traz diversos benefícios para o estudante. (FEISEL; ROSA, 2005) cita treze benefícios desta integração: i. familiarização com a instrumentação; ii. identificação, abstração e comparação de modelos; iii. experimentação; iv. análise de dados; v. desenvolvimento de projetos; vi. aprendizado por erros; vii. criatividade; viii. habilidades psicomotoras; ix. cuidados com segurança; x. comunicação com outros; xi. trabalho/dinâmica de grupo; xii. ética no laboratório; xiii. uso dos sentidos.

Na atualidade, há um amplo desenvolvimento de NTICs (Novas Tecnologias da Informação e Comunicação) voltadas à educação e fala-se inclusive em *e-Learning 2.0* em comparação com a *Web 2.0*, que usa caráter colaborativo e cooperativo para compartilhamento e co-autoria de mídias digitais (neste caso seriam objetos de aprendizagem) na rede. Também há diversas cooperações e consórcios para desenvolvimento de ambientes educacionais usados em conjunto por diversas instituições de ensino espalhadas pelo mundo.

Assim como em todas as áreas em que se necessita a infraestrutura de rede e distribuição de recursos, há avanços em tecnologias e protocolos de comunicação que permitem melhoras na qualidade de trocas de dados. Com uma maior banda de rede disponível fazem-se possíveis melhoras nas visualizações em conjunto com técnicas gráficas tridimensionais.

Estas últimas, associadas a interfaces lúdicas, usam caráter de jogos eletrônicos para prender a atenção de estudantes enquanto apresentam formas mais flexíveis (e menos tradicionais) de aprendizado. Em suma, há uma tendência à flexibilização e evolução do ensino tradicional para uma nova metodologia baseada na teoria de construtivismo social (*social constructivism theory*) para atrair novos estudantes a áreas técnicas tão cruciais para o desenvolvimento de cada nação.

1.3 Objetivos

Sendo a aplicação deste trabalho o par ensino-aprendizagem, os intentos são a melhoria dos mesmos através do emprego da arquitetura, da metodologia de pesquisa e implementação, das ferramentas tecnológicas ou das ideias aqui mencionadas. Também é intenção motivar futuros e atuais educadores no uso da tecnologia em prol de um melhor aprendizado dos estudantes.

Como não poderia ser diferente em um trabalho de cunho técnico da engenharia, necessita-se explicitar e descrever as diversas pesquisas que influenciaram e guiaram o resultado desta tese que propõe uma arquitetura, que pode ser usada como referência, para o desenvolvimento de sistemas que possibilitem o acesso a ambientes virtuais com suporte social, colaborativo, e a experimentação. Os objetivos principais desta arquitetura são integrar:

- materiais didáticos organizados e gerenciados por AVAs;
- suporte à colaboração e à cooperação no aprendizado e em tarefas didáticas;
- suporte à interação social;
- interfaces imersivas com representações tridimensionais;
- características pseudolúdicas (*serious gaming*);
- experimentação remota;
- características de realidade mista;
- suporte a tutores e facilitadores de colaboração autônomos;
- características modulares;
- características de interoperabilidade.

Caracterizam a linha de pesquisa deste trabalho tanto os princípios de modularidade e interoperabilidade quanto a autonomia do sistema, pois estes são especificações desejáveis em qualquer sistema de automação. Juntamente com a proposta de arquitetura também serão apresentados estudos de caso e um protótipo de implementação. Este servirá de validação técnica da arquitetura e será futuramente empregado em cursos de engenharia elétrica e de automação para validação pedagógico-experimental.

1.4 Contribuições da Tese

Apesar deste trabalho se valer de um apanhado de pesquisas e recentes desenvolvimentos técnicos presentes na comunidade científica, vale lembrar que não há um estudo de arquitetura de referência para ambientes virtuais que englobe tantas características

quanto as aqui apresentadas (vide objetivos na Seção 1.3). A maioria das propostas presentes na literatura baseiam-se em sistemas desenvolvidos para suprir a necessidade de cursos específicos da instituição e, muitas vezes, estes ambientes não incluem características modulares e nem se baseiam em uma arquitetura singular, fazendo com que estes sistemas sejam praticamente incompatíveis com outras implementações.

A arquitetura proposta é modular, isto é, construída em blocos que podem ser adicionados ou retirados do sistema de acordo com o desejo do projetista do ambiente. Isto visa dar maior flexibilidade à proposta que pode também integrar outros módulos com funcionalidades diferentes dos aqui propostos. A interoperabilidade entre módulos é garantida pelo uso de protocolos e interfaces de comunicação padronizados e de ferramentas consagradas e livres de licenças comerciais para o desenvolvimento dos mesmos.

A pesquisa aqui apresentada também se torna única do ponto de vista de propiciar diversas interfaces para a interação de estudantes tanto nos ambientes quanto no espaço experimental. Para permitir maior autonomia do ambiente são propostos também soluções de integração com sistemas multiagentes (SMAs) que podem ser usados tanto como facilitadores (mediadores) de colaboração como para tutores autônomos. A base de conhecimento destes agentes se fundamenta tanto em modelos de estudantes (*student models* (GREER; MCCALLA, 1994; ZHOU; EVENS, 1999)) ou somente através de *data-mining* (rastreamento de dados/registros) das interações dos usuários com as diversas interfaces de comunicação do ambiente e suas ramificações.

O protótipo proposto é único no ponto de vista das diversas ramificações e por se basear em tecnologias abertas e de amplo uso na comunidade científica.

1.5 Organização do Manuscrito

Esta tese se baseia em pesquisas da comunidade nacional e internacional (escrita em inglês) e existem muitos acrônimos que muitas vezes possuem tradução imprecisa. Por este motivo muitos destes acrônimos serão descritos na primeira utilização no texto e, se utilizados no decorrer do trabalho, serão muitas vezes apresentados apenas com a abreviatura relacionada ao termo em língua inglesa. Uma lista completa dos acrônimos mais frequentes neste trabalho está presente no início deste trabalho na Seção Lista de Abreviaturas.

Importante destacar que como se trata de um tema atual com diversas referências *online*, algumas implementações e ferramentas serão citadas com o endereço do *Website* atual e sua respectivo último acesso descrito.

Para um melhor entendimento dos leitores foi preparado no Capítulo 2 um embasamento teórico que apresenta algumas noções fundamentais para a compreensão do restante do trabalho e da proposta desta tese. No Capítulo 3 serão descritos e analisados alguns trabalhos da comunidade científica considerados essenciais tanto para a motivação quanto para a contextualização desta tese. Seguindo, o Capítulo 4 delineia a proposta desta tese que apresenta uma arquitetura para ambientes virtuais voltados ao ensino. Para dar suporte à arquitetura proposta são apresentados estudos de caso no Capítulo 5 e, logo após, uma validação para um protótipo é demonstrada no Capítulo 6. Finalmente, no Capítulo 7 serão feitas considerações finais deste trabalho, seguida, por último, das referências bibliográficas citadas.

2 EMBASAMENTO TEÓRICO

2.1 Introdução

Para uma melhor compreensão deste trabalho alguns conceitos e definições serão apresentados neste capítulo. Por estarem relacionadas e inseridas em áreas multidisciplinares, algumas definições são muito amplas e por isso serão simplificadas e focadas para o escopo deste trabalho. Estas definições e explicações servirão para familiarizar o leitor com termos aqui empregados.

Inicialmente uma separação das definições em diferentes subseções será apresentada e organizada de tal maneira a estruturar os diversos conceitos e instrumentos (de tecnologia) empregados. Em cada subseção um direcionamento e uma breve e sucinta explicação serão apresentados. Notoriamente pode-se integrar e mesclar os diversos conceitos e instrumentos apresentados e, visando isso, uma caracterização teórica de cada um é necessária.

Assim, é importante contextualizar o trabalho na área de automação, pois neste também são empregados os princípios e a definição de automação que são: as técnicas, os métodos, os sistemas de controle ou de operação, ou ainda a conjunção destes para controlar processos de maneira a reduzir ao mínimo a intervenção humana (controle manual). Embora não seja muito adequado para este fim o “processo”, o qual se deseja automatizar ou tornar autônomo neste trabalho é o acompanhamento e a monitoria das tarefas dos estudantes, que, por sua vez, utilizam o “sistema” para o aprendizado e a interação entre eles e com os demais educadores.

Por arquitetura, que é a proposta e título desta tese, entende-se o conjunto de ferramentas e conceitos interconectados de maneira tal a formar uma base para proporcionar a construção ou desenvolvimento de sistemas complexos de diversas naturezas e aplicações. Como não poderia deixar de ser, a arquitetura proposta também propiciará uma base ou até um modelo de referência para ambientes virtuais que tenham como finalidade o ensino-aprendizagem em áreas multidisciplinares ligadas à automação.

2.2 Ensino e Aprendizagem

O fenômeno humano da aprendizagem, através do processamento e incorporação de novas informações e vivências (experiências), tem feito parte das preocupações de pedagogos, filósofos, educadores, e cientistas desde quando estas últimas atividades não se viam no que parecem hoje (equivocadamente) pólos distintos e separados (TIMM, 2005). Prova da antiguidade dessa preocupação pedagógica são os chamados métodos socráticos de conscientização do próprio conhecimento, descritos por Platão em obras célebres, como o diálogo Menon (TIMM, 2005; PLATÃO, 2001).

O ensino por sua vez é um processo sistemático onde o objetivo é a transmissão de conhecimentos, ou seja, um processo de instrução. Existem diversas teorias, que muitas vezes são chamadas de paradigmas de ensino-aprendizagem. Nestes paradigmas estabelecem-se diversas relações funcionais entre os elementos do ensino e da aprendizagem. Dentre estes paradigmas iremos descrever três que sumarizam diferentes vertentes: o behaviorismo, o cognitivismo e o construtivismo social.

A teoria comportamental, behaviorismo (do inglês *behaviorism*) é bastante antiga e data do início do século XX (WATSON, 1913) baseada em experimentos de psicólogos em chimpanzés para a elaboração mais básica de seu conceito experimental (SKINNER, 1986). Nela a aprendizagem é uma aquisição de comportamentos através de relações entre o ambiente e o comportamento (ou transformação deste), ocorridas numa história de contingências, estabelecendo uma relação funcional entre ambiente-comportamento. O paradigma behaviorista pode ser resumido como:

- ideia básica: resposta ao estímulo (*stimulus-response*) - todo comportamento é causado por estímulos externos (condicionamento do instrutor/professor). Todo o comportamento pode ser explicado sem considerar estados mentais internos ou de consciência;
- o indivíduo (estudante) é **passivo** durante todo o processo (de aprendizagem);
- o comportamento resultante do processo de aprendizagem pode resultar em reforço (repetição) ou em punição caso o comportamento não seja adquirido (“aprendido”);
- técnicas de ensino: exercícios de **repetição**, ensino individualizado, demonstrações para imitação, **memorização**.

Já o cognitivismo, ou ciência do cognitivismo, provocou uma mudança profunda na visão educacional com sua revolução iniciada em meados de 1960. Nesta nova visão há ainda influências do *framework* empírico da teoria comportamental ainda que um novo paradigma tenha começado. O cognitivismo leva em conta o funcionamento da memória humana no processo de aprendizagem. Nos anos 70 foram desenvolvidos modelos teóricos de memória que impulsionaram a teoria do cognitivismo na educação. Também é importante considerar que o avanço da computação teve grande influência nesta teoria. Conceitos cognitivos como memória de trabalho (anteriormente conhecida como memória de curto prazo) e memória em longo prazo facilitaram as pesquisas e alavancaram novas metodologias na área da ciência da computação. Esta área também possui um de seus ícones em Noam Chomsky linguista, filósofo e ativista político globalmente respeitável (CHOMSKY, 2006). O paradigma cognitivista pode ser resumido como:

- ideia básica: Funções mentais (memória) pode ser compreendida;
- o indivíduo (estudante) é **ativo** durante todo o processo (de aprendizagem) e é visto como um processador de informações;
- o cognitivismo foca nas atividades mentais internas. É necessário determinar como processos como o pensamento, memória, conhecimento, e solução de problemas ocorrem. Pessoas não são “animais programados” que respondem a estímulos do ambiente e sim seres racionais nos quais a ação é consequência do pensamento;
- metáfora da mente como computador: informação entra, é processada e leva a certos resultados/ações;
- técnicas de ensino: **ensino pela descoberta**, apresentação dos **objetivos**, questionários orientados para **compreensão**, esquemas, **debates**, **discussões**, estudos de caso.

Sobre os objetivos cognitivistas do aprendizado é unânime a contribuição de Benjamin Bloom (BLOOM, 1956) para uma taxonomia que expressa qualitativamente diferentes “tipos” de raciocínio e domínios para o aprendizado em ação. Desta forma pode-se organizar habilidades de maneira crescente aumentando o grau e relacionando a ações. Normalmente a representação desta taxonomia se dá pela “rosa de Bloom”, mas aqui será apresentado uma forma simplificada. A Fig. 1 ilustra a taxonomia de Bloom simplificada, à esquerda, e à direita, uma nova proposta de organização segundo POHL (1999).

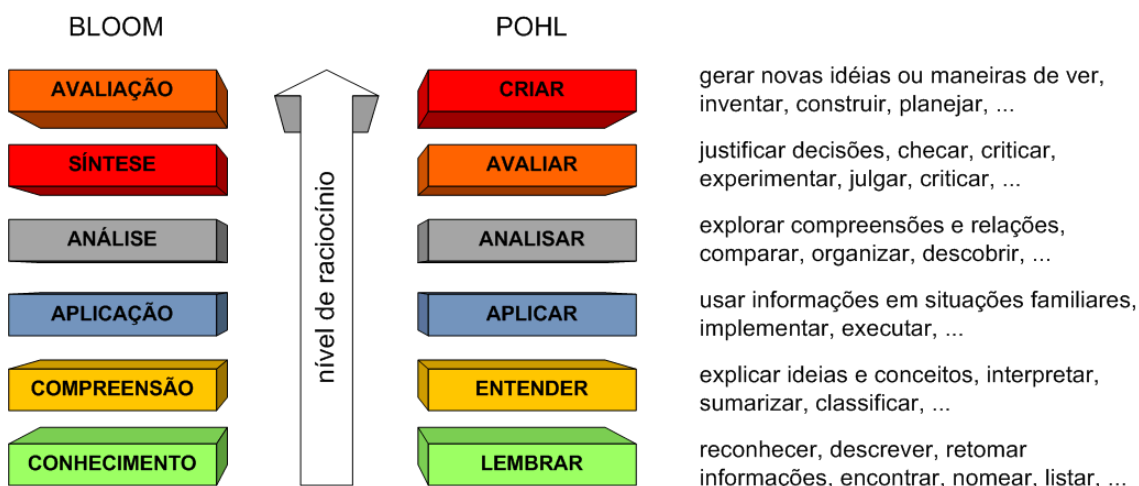


Figura 1: Diagrama do Aprendizado em Ação - habilidades e ações.

Fontes: (BLOOM, 1956) e (POHL, 1999).

O construtivismo, por sua vez, é uma teoria de aprendizado ou filosofia de ensino empregada inicialmente por educadores na década de 90. Um dos pilares desta filosofia é o conceito de que estudantes podem construir o aprendizado à sua própria maneira a partir de novas informações interagindo com o ambiente educacional ou com outros estudantes que tenham diferentes perspectivas. Ambientes de ensino construtivistas devem proporcionar que estudantes utilizem seus conhecimentos e experiências prévias para formular novos, relacionados e/ou conceitos adaptativos na aprendizagem. Através deste ‘pilar’ o papel do professor se transforma em facilitador, guiando os estudantes para que estes construam seu próprio conhecimento. Assim, é de fundamental importância que tanto experiências como conhecimentos prévios estejam em harmonia e devidamente fixados para serem relacionados com os conceitos a serem “ensinados”. Esta área também possui seus ícones em John Dewey (DEWEY, 1938), Jean Piaget (PIAGET, 1972), Jerome Bruner (BRUNER, 1956), Lev Vygotsky (VYGOTSKY, 1978). O paradigma do construtivismo social pode ser resumido como:

- ideia básica: o aprendizado é um processo ativo e construtivo;
- o indivíduo (estudante) é ativo durante todo o processo (de aprendizagem) e é visto como um **construtor** de informações;
- pessoas constroem e criam ativamente suas representações subjetivas da realidade. Novas informações estão conectadas a conhecimento anterior, assim representações são subjetivas;
- técnicas de ensino: imitação, **debates**, **discussões**, jogos de papéis (*role playing*).

Outro paradigma educacional relacionado à psicologia humana é o humanismo. Com seus proponentes em Abraham Maslow (MASLOW, 1943) e Carl Rogers (ROGERS, 1970), entre outros. Este paradigma pode ser resumido como:

- ideia básica: o aprendizado uma ação pessoal para atingir um potencial;
- o indivíduo (estudante) é visto como um ser com necessidades cognitivas e afetivas;
- ênfase na liberdade, dignidade e potencial de humanos;
- o aprendizado é **centrado no estudante** e pessoal, facilitado por professores, com o objetivo de desenvolver pessoas autoatualizáveis (autossuficientes no aprendizado) num ambiente cooperativo e assistido;
- técnicas de ensino: ensino individualizado, debates, discussões, jogos de papéis, **painéis**, simulações, **resolução de problemas**.

Recentemente, com a consolidação da Internet, surge uma teoria da era digital chamada sugestivamente de conectivismo¹ (ou conexãoismo) baseada na análise e na limitação das teorias comportamentais, cognitivistas e construtivistas, mas procura estudar os efeitos da tecnologia no modo de vida, na comunicação e no aprendizado humano. A teoria ainda se encontra em fase de formulação.

Nota-se que há evolução nítida entre estes paradigmas e que conforme há um maior “conhecimento” da psicologia humana, maior é a adaptação ao usuário, isto é, centralização no estudante.

Também há uma clara definição entre o aprendizado passivo e o ativo. No passivo, estudantes são limitados a estímulos de áudio e vídeo do professor/instrutor ou de materiais didáticos, ou seja, ouvir e ver ou ler sobre assuntos em questão. No ativo há a quebra desta barreira, ou seja, estudantes participam, simulam, debatem, interagem, colaboram, cooperam, projetam, e assim atingem níveis mais altos de raciocínio durante o aprendizado com o professor/instrutor.

A tese de doutorado de TIMM (2005) faz uma análise do modelo *e-Learning-by-doing* para validar a elaboração de projetos como estratégia pedagógica para o ensino de engenharia. Desta forma são abordados paradigmas de pesquisa do ensino científico e tecnológico e descrito um perfil cognitivo dos engenheiros. Este perfil usa um modelo baseado nas teorias de mente modular e computacional. Este modelo é facilmente justificável pela prática presente nos problemas e desafios associados à engenharia, que são lógicos e estruturados.

2.3 Tecnologias de Educação

A educação engloba o ensino e a aprendizagem, ou seja, educar implica nos processos de ensinar e aprender. Apesar do nome, tecnologias de educação² é a área de pesquisa dedicada ao estudo e à prática de aumentar o rendimento e a facilitar a aprendizagem através da criação, uso e gerenciamento de processos e recursos tecnológicos apropriados (RICHEY, 2008). Em outras palavras refere-se às metodologias que possam ser usadas com o uso da tecnologia para o ensino. O termo está associado à teoria instrucional e de aprendizagem (ensino). Há atualmente diversas metodologias deste tipo que envolvem sobretudo computadores (ou semelhantes), *hardware* e *software*, e a *Web/Internet* como meio de divulgação e acesso. Os computadores, por sua vez, utilizam ferramentas de *software* para guiar e/ou controlar os estudantes durante o processo de ensino-aprendizagem.

As tecnologias de ensino podem ser definidas como um conjunto de ferramentas que auxilia o aprendizado (ou a aprendizagem) dos estudantes. O termo se baseia no conceito

¹do inglês *connectivism*.

²os termos em inglês associados são *Educational Technology* ou também *Learning Technology*.

abrangente de tecnologia que se refere a objetos materiais como, por exemplo, máquinas, mas também incluem sistemas, métodos e técnicas de aplicação.

As tecnologias de ensino são influenciadas e se baseiam nas “escolas” teóricas, ou também filosóficas, comportamentais (*behaviorism*), do cognitivismo e do construtivismo. Elas possuem grandes características psicológicas e pedagógicas que serão levemente abordadas neste trabalho, pois o foco é mais prático.

Dentre as várias tecnologias de educação podem-se citar a aprendizagem baseada em problemas/projetos (*PBL - Problem-/Project-Based Learning*) e a aprendizagem baseada na pesquisa e no questionamento (*Inquiry-Based Learning*), entre outras (KIRSCHNER; SWELLER; CLARK, 2006). Essas são voltadas ao ensino-aprendizagem ativo usadas para facilitar o aprendizado (MARTYN, 2007). Tecnologias que incluem ciências de processo e físicas podem ser incorporadas em ensino baseado em problemáticas, projetos, ou investigativas já que todas as três possuem filosofias similares. Todas estas são centradas no estudante e envolvem cenários do mundo real no qual estudantes estão engajados ativamente em atividades de pensamento crítico e reflexivo. O processo em que os estudantes são encorajados a participar pode ser considerado uma tecnologia contanto que seja baseado em tarefas empíricas.

Dentre os benefícios do emprego de tecnologia de ensino podem-se citar:

- facilitação do acesso a materiais de ensino - através da disponibilização online dos mesmos;
- motivação dos estudantes - o *CBT* possibilita interação e *feedback* instantâneos na solução e resposta a perguntas. Ainda, as interfaces computacionais são “pacientes” e “imparciais” (não julgam) o que muitas vezes incentiva o estudante a continuar o estudo;
- ampliação da participação - o material desenvolvido pode ser acessado tanto por estudantes presenciais quanto de EaD;
- melhoramento da expressão dos estudantes - através da interação as habilidades escritas dos estudantes são aprimoradas;
- facilitação da aprendizagem - simulações didáticas e outras técnicas interativas e ilustrativas podem ser usadas com mídias eletrônicas;
- estruturação da qualidade do ensino - monitoração e acompanhamento podem ser melhor conduzidos usando ferramentas de *software* e também facilitam a modificação/adaptação de conteúdos para melhor atender as necessidades de cada estudante/grupo.

Obviamente, pode-se também apontar um revés inerente a qualquer método de aplicação: os professores/instrutores necessitam adaptar-se e familiarizar-se com as ferramentas disponíveis e evoluir em conjunto com as tecnologias e tendências atuais. Apesar de constituir um custo adicional inicial o uso de tecnologia no ensino está cada vez mais se consolidando.

Do ponto de vista técnico geralmente sistemas assim como “tecnologias” evoluem em diversas direções (vertentes/dimensões): do analógico/tradicional para o digital (digitalização); do físico (real) para o virtual (virtualização); do massivo para o individual (personalização³) e do individual para o compartilhado (colaboração/co-autoria); do centralizado para o distribuído (distribuição); e também do estático para o dinâmico/flexível/adaptável (dinamização/flexibilização/adaptabilidade) e também do estático para o móvel (mobilidade). Assim, não é difícil imaginar que tais “dimensões tecnológicas” também

³também aportunizado customização.

sejam aos poucos empregadas no ensino-aprendizagem, justamente quando as bases de desenvolvimento se encontram devidamente evoluídas. Mais precisamente, estas bases se encontram presentes hoje em dia no paradigma conceitual de *Web 2.0* e *e-learning 2.0*. Um estudo de ferramentas mais comuns está presente na seção seguinte.

Ainda podem-se referenciar como tecnologia de ensino os conceitos de aprendizado híbrido ou mesclado (*Hybrid Learning / Blended Learning*) que propõem o uso de “tecnologia”, seja ela qual for, em conjunção ou em complementação com o ensino-aprendizado tradicional (presencial).

2.4 Ferramentas de Suporte à Colaboração

Qualquer ferramenta que suporta ou possibilita comunicação síncrona (ambos os usuários presentes num mesmo ambiente de comunicação no mesmo instante de tempo) ou assíncrona (em instantes diferentes de tempo) entre usuários distintos pode ser classificada como “colaborativa”, embora a colaboração necessariamente nem sempre ocorra. Para garantir a colaboração, usuários têm de estar cientes das funcionalidades da ferramenta e usá-la corretamente para interagir colaborativamente. Em situações corriqueiras reais, pessoas estão instintivamente cientes das “regras” (funcionalidades) do “ambiente” no qual estão presentes. Em ambientes virtuais essas “regras” nem sempre são explícitas e consequentemente usuários têm de ser encorajados a colaborar usando os meios de comunicação presentes.

As pesquisas em colaboração assistida por computador (*CSC - Computer Supported Collaboration/Cooperation*) focam, em sua maioria, nas “tecnologias” que afetam grupos, organizações, comunidades e sociedades. Este nicho de pesquisa desenvolveu-se a partir de estudos que assistiam atividades de trabalho (*CSCW*) e de relacionamentos. Após isso, mais e mais ferramentas e metodologias foram desenvolvidas para suportar uma grande variedade de atividades sociais e recreacionais, e mercados consumidores expandiram suas bases devido ao crescimento de usuários com acesso à Internet. Esta demanda criou o que pode-se chamar de “mundo com suporte colaborativo/cooperativo”. Este “mundo” compreende contextos em que centenas de ferramentas de *software* são empregadas para mediar atividades humanas como comunicação, coordenação, cooperação, competição, entretenimento, jogos, arte, música e muitos outros.

A subárea de pesquisa, comunicação mediada por computadores (*CMC - Computer Mediated Communication*), investiga especificamente as interações entre humanos e computadores (ou mídias digitais) para formar, suportar e manter relações com outros (aspecto social), regular/coordenar o fluxo de informações (aspecto instrucional), e “tomar decisões”. Ela não foca nos “produtos” da colaboração, mas sim na colaboração (junção de usuários) em si. Em contraste, a *CSC* é focada no produto da colaboração e não no caráter ou consequências emocionais das junções ou relações dos usuários. Esta é a essência da diferença entre comunicação e colaboração.

O termo *CSC* surgiu para substituir termos de computação em grupo (*workgroup computing*, que foca na tecnologia que suporta o trabalho); *groupware* (que virou palavra comercial para descrever diversos sistemas mal desenvolvidos de atividades em grupo); e *CSCW* (que suporta grupos de pessoas trabalhando em conjunto em um projeto em comum).

Existem basicamente dois tipos diferentes, mas não opostos, de implementações de *CSCs*: *softwares* sociais (*socialwares*), que têm como objetivo principal proporcionar laços sociais; e *softwares* colaborativos, que “produzem” resultados práticos.

Tecnologias básicas como *netnews*, *e-mail*, *chat* e *Wiki* (acrônimo para *What I Know Is*) podem ser classificadas como ambos os tipos de *CSC*. Enquanto *socialwares* focam no uso de “comunidades virtuais” os *softwares* colaborativos possuem preocupações com gerenciamento de conteúdo e no “produto”. Embora *softwares* sejam desenvolvidos para proporcionar laços sociais ou produzir resultados práticos, é difícil suportar a colaboração sem também permitir que relações entre usuários se formem, e suportar interações sociais sem que algum trabalho produzido seja de co- ou autoria compartilhada. Analogamente a diferenciação entre *software* social ou colaborativo pode ser comparada ao “brincar/jogar” ou trabalhar. Alguns autores defendem a ideia de que a ética lúdica deve ser empregada no trabalho para que a atividade de usar computadores se torne uma experiência mais confortável.

2.4.1 Classificação

Ferramentas colaborativas são comumente divididas ou classificadas em três categorias, segundo seu propósito: de comunicação, de conferência e de gerenciamento colaborativo (coordenação). As “tecnologias” envolvidas nestas ferramentas são: colaboração síncrona, assíncrona ou até mesclada (ambos).

Uma das muitas interpretações do significado de comunicação é a troca não estruturada de informação usando diversos métodos (ou canais). Exemplos desta categoria são: ligações telefônicas ou discussões por *chats* ou mensagens instantâneas (*IM - Instant Messaging*). Conferência se refere ao trabalho interativo com o objetivo compartilhado comum (entre os participantes). Exemplos são: votações, discussões (*brainstorming*), etc. Coordenação refere-se ao complexo trabalho interdependente buscando um objetivo compartilhado comum. Por exemplo, a metáfora que descreve um time esportivo, onde todos têm de contribuir para a jogada certa na hora certa e se ajustar para as situações do jogo apesar de cada um estar fazendo uma coisa (tarefa) diferente (funções e papéis distintos), para que a equipe “ganhe”.

Com o avanço das mídias eletrônicas e também da largura de banda oferecida para a maioria dos usuários da Internet surgiu o conceito de *Web 2.0* que seria a evolução natural da *Web* com diversas implementações de compartilhamento e co-autoria de vídeos, textos, áudios, ou seja, uma *Web* mais colaborativa. Diversas ferramentas de colaboração fazem uso do termo *Web 2.0* para denotar a capacidade de compartilhamento de informações via Internet.

Em BAFOUTSOU; MENTZAS (2002) há uma completa classificação e avaliação de diversas ferramentas para colaboração e implementações existentes dos mais diversos tipos e suas funcionalidades. A seguir, uma categorização será apresentada baseada em SERCE; YILDIRIM (2006), onde serão descritas ferramentas e conceitos com alguns exemplos.

2.4.1.1 Ferramentas de Comunicação Eletrônica

Ferramentas de comunicação eletrônica assistem ou suportam usuários de maneira que estes possam trocar mensagens, arquivos, dados, ou documentos entre eles facilitando o processo de compartilhamento de informação através da colaboração. Exemplos são:

- *E-mail*;
- Fax;
- Correio de voz;
- Wikis;
- Publicações na *Web* (ex. *blogs*);
- Conferências síncronas simples (*chat*);
- Mensagens *SMS* (*Short Message Service*) e similares.

2.4.1.2 Ferramentas de Conferência Eletrônica

Ferramentas de conferência eletrônica facilitam o compartilhamento de informações de uma maneira mais interativa. Exemplos são:

- Fóruns da Internet (também conhecidos como quadros de mensagens/discussão);
- *Chat online* - plataforma virtual que facilita a troca de mensagens de texto online;
- Mensagens instantâneas (*IM*);
- Telefonia (*VoIP - Voice over Internet Protocol*);
- Conferência de vídeo - computadores em rede áudio e vídeo;
- Conferência de dados - computadores em rede compartilhando um “quadro branco” comum que cada usuário pode modificar;
- Compartilhamento de aplicações - usuários acessam documentos ou aplicações de seus respectivos computadores em tempo real (embora nenhuma técnica de tempo real seja empregada esta designação é usada para simultâneo via rede);
- Sistemas de reunião eletrônica (*EMS - Electronic Meeting Systems*) - geralmente implementados em grandes salas de conferência onde um projetor de vídeo ligado a um computador que pode ligar diversas salas em rede.

Nota-se que nesta categoria *chats* também se encaixam, pois são ferramentas tanto de comunicação quanto podem ser incluídas também em conferências (normalmente em conjunção com outra funcionalidade como vídeo, por exemplo).

2.4.1.3 Ferramentas de Gerenciamento de Colaboração

Esta categoria de ferramentas colaborativas facilita o gerenciamento de atividades em grupo. Exemplos são:

- Calendários eletrônicos (ou *softwares* gerenciamento de tempo);
- Sistemas de gerenciamento de projetos;
- Sistemas de fluxo de trabalho;
- Mercados de predição;
- *Softwares* de sistemas sociais;
- Sistemas de gerenciamento de conhecimento (*KMS - Knowledge Management Systems*);
- Sistemas extranet e sistemas hospedados em intranet;
- Planilhas *online*.

2.5 Simulação e Modelo Virtuais

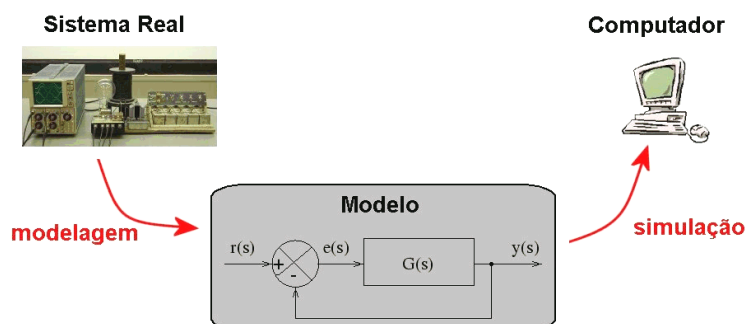


Figura 2: Modelagem e simulação - diagrama.

Um modelo virtual é uma cópia de algo físico ou abstrato, ou ainda um molde para manipulação virtual. Os modelos podem se basear: na forma, modelo tridimensional; no comportamento, modelo comportamental; nas propriedades dinâmicas (físicas), modelo matemático; e em muitas outras características que forem pertinentes para a modelagem. A simulação de algo sempre se baseia em modelos e a maioria deles é simulada computacionalmente, isto é, por computadores que processam dados através de regras matemáticas (vide Fig. 2) ou algorítmicas.

Simuladores são ferramentas que reproduzem condições de sistemas baseados em modelos computacionais através da mudança de parâmetros. Hoje em dia há simuladores para qualquer tipo de processo físico, e os modelos estão cada vez mais aprimorados. Simuladores computacionais são aqueles desenvolvidos em *software*.

Simulações usadas para treinamento de pessoal podem ser categorizadas em FRANK; HELMS; VOOR (2000):

- **simulações “vivas”** (*live simulation*) - pessoas reais usam elementos simulados (também chamados de *dummies*) no mundo real;
- **simulações virtuais** - pessoas reais usam elementos simulados em mundos também simulados, chamados de ambientes virtuais;
- **simulações construtivas** - pessoas simuladas usam elementos simulados em ambientes virtuais.

Simulações construtivas também são conhecidas como *wargaming* devido à semelhança deste cenário com jogos de guerra. Esta classificação é chamada também de *LVC* (*Live, Virtual and Constructive*) e é bastante usada no treinamento militar. Conforme FRANK; HELMS; VOOR (2000), as questões de custo e repetitividade (se a simulação pode ser novamente executada com as mesmas condições, *stimuli* e parâmetros de simulação, obtendo-se a mesma resposta, resultado) também estão presentes na classificação, assim: simulações vivas são de custo alto (desenvolvimento e recursos humanos e materiais) e de baixa repetitividade; simulações virtuais de médio custo e repetitividade mediana; e simulações construtivas de baixo custo e alta repetitividade. Em contraste às características anteriores, a fidelidade da simulação, como era de se esperar, é mais alta na simulação viva decrescendo para menor fidelidade na construtiva.

2.6 Realidade Mista

Realidade mista (*mixed reality*), como o próprio nome sugere, é a mescla ou mistura de simulações (elementos virtuais) com a realidade (elementos físicos). O termo foi definido por MILGRAM; KISHINO (1994) como “a mescla de mundos reais” e virtuais em algum ponto da linha da virtualidade (do inglês *virtuality continuum*) que conecta completamente a realidade com a virtualidade. O conceito de realidade mista engloba: virtualidade, virtualidade aumentada (do inglês *augmented virtuality*), realidade aumentada (do inglês *augmented reality*) e realidade (MILGRAM et al., 1995) (vide Fig. 3). Isso indica que a realidade mista pode ser considerada como uma especialização de realidade aumentada e da mídia interativa. Segundo BILLINGHURST; KATO (1999), realidade mista é comumente entendida como uma integração transparente (do inglês *seamless*) ao usuário entre os mundos real e virtual.

A Figura 3 apresenta quatro situações em que o conteúdo real e virtual varia desde um sistema completamente real até um sistema completamente virtual. Descrevendo os quadros da esquerda para a direita, tem-se um operador (pessoa) interagindo com um

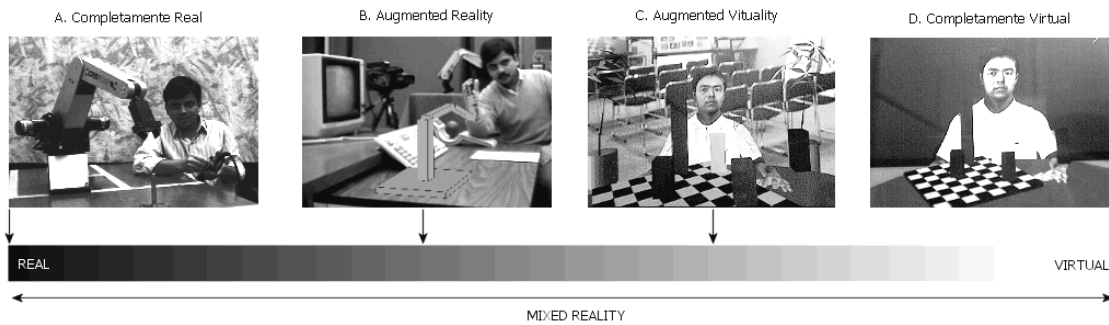


Figura 3: Realidade mista - espectro (*continuum*).

Fonte: (MILGRAM et al., 1995).

robô real, na sequência tem-se um sistema no qual o robô real é substituído por um sistema virtual (por exemplo, tal situação poderia ser utilizada em treinamentos de operadores/programadores de robôs, sem termos o risco de danificarmos os equipamentos). Este cenário apresenta o conceito de *augmented reality*, no qual a realidade é “aumentada” com um componente virtual que incrementa o “ambiente”. No terceiro quadro tem-se um cenário real (com sala, cadeiras e corpo do estudante reais) com um tabuleiro de xadrez virtual assim como o braço do manipulador. Este último representa o cenário de *augmented virtuality*, onde a simulação é enriquecida com elementos reais, como o fundo da sala e a vídeo do operador. Por último, uma situação completamente virtual.

A implementação de um sistema de realidade mista se dá da mesma forma que os experimentos reais remotos com funcionalidades virtuais (por exemplo, instrumentos de medição virtuais). Uma vez que experimentos de realidade mista exigem a integração entre dispositivos reais e virtuais, propostas de integração entre sinais físicos reais e sinais virtuais (isto é, informação) são necessárias.

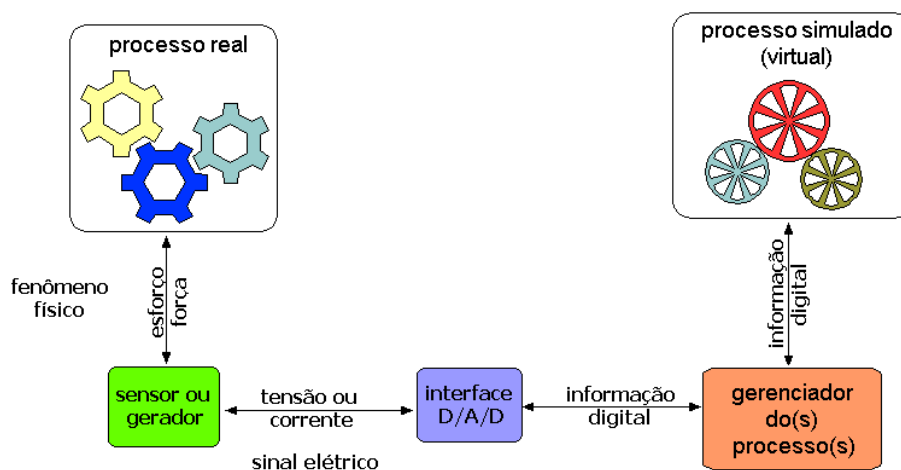


Figura 4: Realidade mista - exemplo de interconexão de processos reais e virtuais.

A Fig. 4 ilustra como se pode interligar processos reais com virtuais através da interligação de sinais elétricos com informações (sinais) digitais. Esta não é a única forma de interligação, mas é largamente a mais usual. O gerenciador do(s) processo(s) é a peça chave desta interligação, pois é o elemento responsável pelo direcionamento e conexão entre os sinais. O sistema de aquisição de dados (*DAS - Data Acquisition System*) é fundamental para a extração de informações necessárias do processo real.

Um conceito de interfaceamento para integração entre sistemas (processos) reais e virtuais são os *hyper-bonds* (BRUNS, 2005). O conceito usa as chamadas ‘hiperconexões’ (*hardware* e *software* para ‘converter’ os sinais para representações lógicas digitais) e teoria de *Bond Graphs* (PAYNTER, 1960; KARNOPP; ROSENBERG, 1990) para modelagem matemática da ligação. Pode-se dividir a implementação básica de *hyper-bonds* em duas partes: i. a parte de aquisição de dados, com *hardware* que desempenha o papel de transdutor e condicionador de sinais; e ii. a parte da comunicação com o computador gerenciador, com o *software* que manipula e é responsável pela interconexão dos diferentes sinais.

No caso da Fig. 4, o conceito de *hyper-bond* pode ser implementado usando a ideia de gráficos de Bond no gerenciador de processos. Gráficos de Bond não necessitam ser usados em casos onde há necessidade de somente uma simples conexão unidirecional de sinais digitais, embora a ideia perpetue na interconexão de sinais. Quando a aquisição de dados é indireta, dados de processos reais se confundem com dados de experimentos virtuais, e, assim, tem-se um *hyper-bond* transparente para o usuário que pode não diferenciar sinais/variáveis reais de virtuais.

A realidade mista combina vantagens da virtualidade com elementos da realidade. Vantagens notórias dos simuladores computacionais - como a manipulação do tempo de simulação e, por consequência, execução passo a passo (análise estática de instantes de tempo para visualização de eventos) - são de grande valia para ilustrar fenômenos físicos e características específicas dos modelos que, por sua vez, imitam componentes reais. Outra vantagem da qual pode se tirar proveito em elementos virtuais é a capacidade de instanciação de diversos modelos, o que é chamado de replicação, que pode inclusive diminuir custos e aumentar a acessibilidade.

É crescente o número de pesquisas que envolvem realidade mista encontradas na literatura, em aplicações nas áreas de educação (MÜLLER; FERREIRA, 2004), entretenimento (jogos eletrônicos, filmes, etc.) (FAUST; YOO, 2006), cultural (HALL et al., 2001), médica (BOCKHOLT et al., 2003), teleoperação com *force-feedback* (*haptics*) (YOO; BRUNS, 2004), manutenção inteligente (ESPINDOLA et al., 2010), projetos da indústria automotiva e aeroespacial (REGENBRECHT; BARATOFF; WILKE, 2005), entre outras.

Cenários diversos podem empregar a realidade mista, entre as áreas mais comuns de aplicação estão: entretenimento (jogos eletrônicos, produções cinematográficas, etc); educação (EaD, experimentos didáticos, etc), comércio (projetos arquitetônicos, assistência ao turismo, etc), indústria (simuladores, prototipação rápida, *Hardware-* e *Software-in-the-Loop*, manutenção, etc), teleoperação voltada para as mais diferentes áreas, etc.

2.7 Ambiente e Espaço Virtuais

Embora dificilmente possamos distinguir a diferença conceitual entre ambiente e espaço, neste trabalho usaremos o termo ambiente virtual para denotar todo ‘espaço físico’ simulado em ambiente computacional onde usuários podem interagir através de conexões de rede de computadores.

A seguir alguns conceitos e temas relacionados a ambientes virtuais serão apresentados e descritos em subseções.

2.7.1 Colaboração e/ou Cooperação em Ambientes Virtuais

A colaboração envolve comunicação, coordenação e cooperação em qualquer meio. A comunicação se realiza através da troca de “mensagens”; a coordenação se realiza através do “gerenciamento de pessoas”, atividades e recursos; e a cooperação se realiza através de operações num espaço compartilhado para a execução das tarefas. Com base nestas premissas foi proposto em 1991 um modelo chamado de 3Cs, originalmente desenvolvido por ELLIS; GIBBS; REIN (1991) e realimentado por PIMENTEL et al. (2006). Este modelo é frequentemente referenciado na comunidade científica para classificar os sistemas colaborativos assim como diferentes ambientes virtuais que possuem propósitos distintos com ferramentas de colaboração. A Fig. 5 ilustra os pilares do modelo e algumas divisões (classificação) de ferramentas.



Figura 5: Modelo 3Cs - diagrama de relações e classificação de sistemas colaborativos.
Fonte: (PIMENTEL et al., 2006).

Cooperação e colaboração podem ser diferenciadas da seguinte maneira: o primeiro sendo uma espécie de protocolo para evitar conflitos e proporcionar sincronização de tarefas orientadas ao trabalho (modelo de linha de produção); enquanto que a colaboração sendo a busca por novos produtos, ferramentas, processos e a qualidade geral da tarefa (modelo de grupo de trabalho criativo) (CARROLL et al., 2003).

O acrônimo AVC⁴ (Ambientes Virtuais Colaborativos) será empregado durante todo trabalho e se refere a sistemas computacionais que proporcionam ambientes virtuais onde é possível a colaboração entre usuários.

2.7.2 Ambientes Virtuais de Aprendizagem

AVAs são ambientes computacionais desenvolvidos com o intuito de facilitar o ensino-aprendizagem em redes de computadores e, mais especificamente, usado para EaD. Em geral, diferentes implementações de AVAs possuem capacidade de gerenciar e hospedar material de ensino, cursos, mídias eletrônicas, tutoriais, etc. Até por esse fato muitos AVAs também podem ser classificados como sistemas de gerenciamento de curso ou de

⁴o autor e alguns pesquisadores utilizam CSCE - *Computer Supported Collaborative Environment* - em suas publicações em língua inglesa.

material didático⁵. Diferentemente de uma página comum da *Web* onde todo material está disponível abertamente (estilo comumente visto em tutoriais e *how to's*), AVAs oferecem suporte para controle e administração de usuários. Normalmente também oferecem ferramentas simples de edição de materiais didáticos e outras mídias eletrônicas baseadas na *Web*, como editores de HTML e ferramentas sociais como alguns módulos de identificação, comunicação e colaboração entre usuários; como, por exemplo, perfis de usuários, fóruns (comunicação assíncrona de mensagens), *chats* (comunicação síncrona de mensagens), troca de arquivos, *Wiki's*, etc (vide Fig. 6).

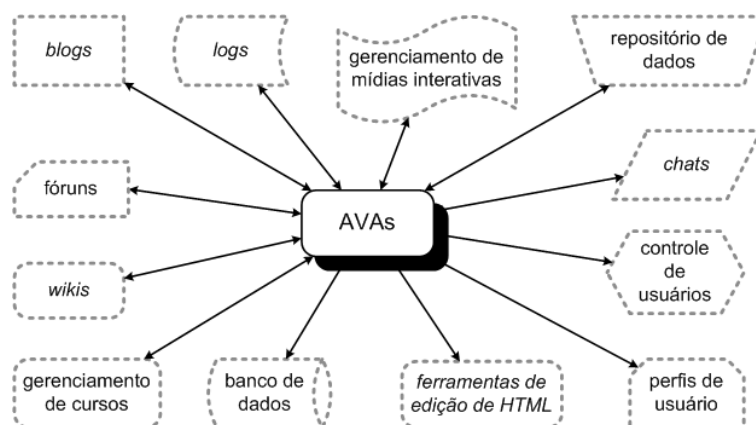


Figura 6: AVAs - ferramentas mais comuns.

Há atualmente uma grande oferta de implementações de AVAs com os mais variados focos e filosofias de interação/colaboração entre usuários. Entre as filosofias abrangidas na comunidade científica destaca-se o construtivismo social (*social constructivism*) que defende a colaboração entre estudantes pregando a descentralização da informação e do aprendizado. Desta forma, professores de disciplinas ou cursos são vistos como facilitadores do conhecimento e não somente detentores do conhecimento e únicos transmissores. A troca de informações e reflexão dos estudantes é encorajada para a construção do conhecimento (*knowledge building*). Esta metodologia é especialmente focada para o EaD onde a figura do professor “perde” um pouco o “poder” original para a “sociedade de aprendizado”.

Algumas implementações de AVAs existentes na comunidade científica, que foram estudadas neste trabalho, estão descritas na Seção 3.3.

2.7.3 Ambientes Virtuais de Trabalho

Estes ambientes estão geralmente ligados ao termo *CSCW* que foi criado nos anos 80 para direcionar pesquisas ligadas ao efetivo uso de tecnologia para auxiliar pessoas nas suas tarefas, em especial quando estas estão desenvolvendo atividades de colaboração com restrições de coordenação. Pesquisadores reconhecem que este não é somente um desafio tecnológico, mas também sócio-organizacional (GRUDIN, 1988; BOEDKER, 1991; KAPTELININ, 1995), mais ainda, não é somente restrito a usuários distribuídos e representação visual (FJELD et al., 2002). Ainda assim, somente recentemente houve uma mudança radical na diferenciação entre colaboração e cooperação no trabalho (vide Seção 2.7.1).

⁵as siglas de AVA em inglês associadas são: *Virtual Learning Environment - VLE*, *Course Management System - CMS*, e *Learning Management System - LMS*.

Ambientes *CSCW* são compostos geralmente por trabalhadores orientados/habilitados ao conhecimento (*knowledge enabled workers*), ambientes colaborativos virtualizados, espaços de trabalho compartilhados, comunidades virtuais e ambientes sensíveis a interações (*responsive environments*) (LASO-BALLESTEROS; KARLSSON, 2006).



Figura 7: *Porta-Person* e *Virtual Workplace* da Sun.

Fonte: (YANKELOVICH et al., 2007).

Existem ainda pesquisas de *CSCW* que incorporam conceitos de realidade mista nos espaços de trabalho (BILLINGHURST; KATO, 1999). Há também um exemplo de “produto” que foi desenvolvido pela Oracle (antiga Sun Microsystems) para serem usados com estes fins (vide Fig. 7), chamado de *Porta-Person* (YANKELOVICH et al., 2007; ORACLE, 2011a).

2.7.4 Ambientes Imersivos

Pode-se retratar imersão ou o sentimento de imersão (*feeling of immersion*) como o estado de consciência no qual o usuário (imerso) tem a noção física do seu corpo atenuada (ou até perdida) pelo envolvimento em um ambiente muitas vezes artificial. Assim, um ambiente imersivo é aquele que leva o usuário a crer que se encontra em um meio no qual ele conhece as “leis naturais” e conscientemente se comporta de maneira natural. Há, desta maneira, uma profunda relação entre o usuário e o ambiente que o envolve e com o qual ele se identifica. Sistemas computacionais que simulam mundos reais podem ser vistos como ambientes imersivos já que o usuário é representado, através de um *avatar*⁶ e se encontra, mesmo que virtualmente, imerso no ambiente em questão.

Este termo é muito usado na área de realidade virtual e nela são considerados diversos tipos de imersão: sensorial e motora; cognitiva; emocional; espacial; e psicológica. Sabe-se que ambientes são crescentemente mais convincentes de acordo com a aproximação da realidade e de respostas do ambiente. Assim, a percepção pode ser estimulada pelos cinco sentidos (visão, audição, tato, olfato e paladar) e a interação por diferentes meios de comunicação: gestos, reconhecimento de voz, interface cérebro-computador, etc.

A Fig. 8 ilustra um ambiente imersivo, usado para jogos eletrônicos com percepção visual 3D através de um *HMD* (*Head Mounted Display*) e mecanismos de *force-feedback* (*haptic*) (GUTIERREZ et al., 2004).

⁶O termo *avatar* provem do hinduísmo e tem o significado de “projeção mental”.

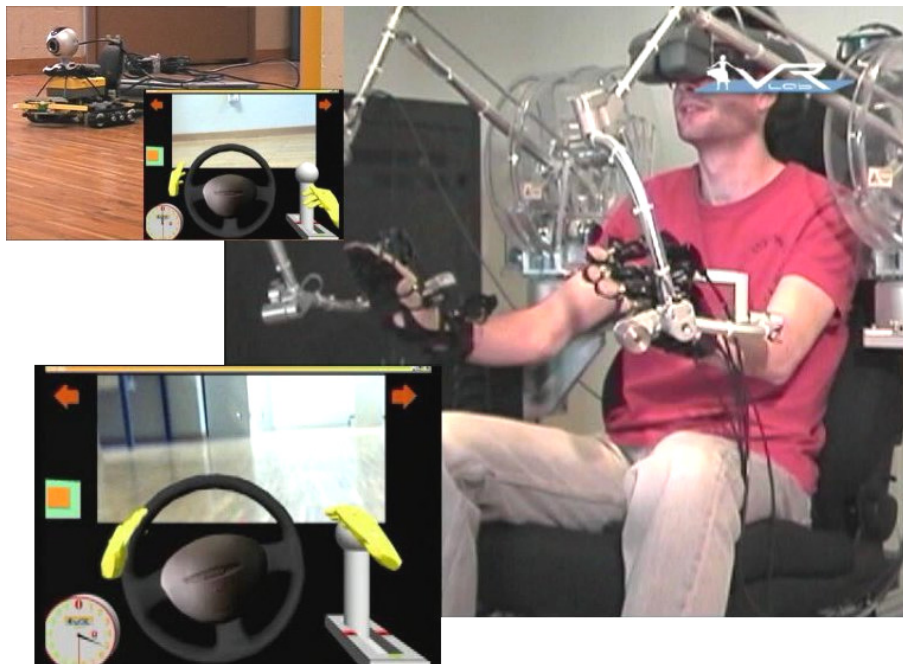


Figura 8: Ambiente imersivo - exemplo com percepção visual e tátil.
Fonte: (GUTIERREZ et al., 2004).

2.8 Experimentação

Tida como parte fundamental e prática de qualquer disciplina, a experimentação quase sempre se dá em laboratórios. Estes laboratórios podem ser basicamente classificados, segundo AUER et al. (2003), quanto à localização dos estudantes e a natureza do experimento, conforme mostra a Tabela 1. Analisando esta tabela, nota-se um confronto evidente entre virtualidade e realidade. Esta taxonomia não apresenta uma nomenclatura adequada para experimentos reais que são acessados remotamente, pois “laboratórios remotos” não exclui a possibilidade de estes serem também virtuais e “laboratórios virtuais” não exclui a possibilidade deles serem locais. O termo remoto e virtual nestes casos é usado sem uma análise detalhada.

Simulações realísticas podem representar fidedignamente situações reais de experimentos físicos, embora experimentos reais sejam mais efetivos para o aprendizado. Laboratórios virtuais não existem fisicamente e geralmente utilizam todos os equipamentos simulados (virtuais). Experimentos reais possuem potencial educacional melhor que simulações, pois apresentam características geralmente desprezadas pelos simuladores - como ruídos (perturbações), limites de controle e efeitos não lineares -, além de possuir uma dinâmica real do processo e possibilitar o manuseio de equipamentos (*hands-on*) e de uso de equipamentos empregados na indústria.

Tabela 1: Classificação de laboratórios.

Fonte: (AUER et al., 2003).

		Usuário	
		Local	Remoto
Experimento	Real	Laboratório Tradicional	Laboratório Remoto
	Virtual	Simulação Local	Laboratório Virtual

A utilização de laboratórios reais produz frequentemente diferenças entre resultados teóricos e resultados experimentais (implementação prática). Apesar de este aspecto levar, muitas vezes, os estudantes à formulação do mito de que “a teoria na prática não funciona”, é importante frisar que a teoria, se bem aplicada, funciona resolvendo problemas do mundo real. Portanto, para se utilizarem tanto laboratórios remotos (reais) como simulações, é necessário demonstrar as diferenças entre as abstrações (modelos), os equipamentos reais e seus respectivos comportamentos em materiais educacionais.

A realidade mista aliada à experimentação traz a possibilidade de criação de cenários que envolvem tanto simulações quanto equipamentos reais. Experimentos reais se diferenciam de simuladores pela utilização de sistemas (equipamentos) físicos. Vale a ressalva que, nos casos de laboratórios remotos (vide Tabela 1), embora usuários manipulem equipamentos reais, estes não estão em contato físico com os equipamentos, pois a interação se dá através de uma interface remota. Desta forma, ainda que simulações estejam em oposição com experimentos reais, simulações fidedignas do comportamento de um determinado sistema podem ser percebidas/mascaradas remotamente como reais.

De maneira a disponibilizar uma interface remota ao usuário, sistemas de aquisição e escrita de dados (interface com o processo) e servidores *Web* são comumente utilizados no servidor da experiência de forma a tornar experimentos remotamente acessíveis. Formas e *softwares* semelhantes aos usados para simular modelos de componentes reais são utilizados para disponibilizar dados de equipamentos físicos aos usuários remotos em experimentos reais.

A Fig. 9 ilustra a estruturação e os diversos componentes presentes na experimentação remota. O gerenciador do experimento é executado no lado do servidor e é responsável pela concentração dos dados do ensaio, que podem envolver simuladores ou equipamentos reais conectados através de sistemas de aquisição de dados. A visualização remota do experimento, no lado do cliente, é geralmente vista pelo usuário através de uma interface gráfica (*GUI - Graphical User Interface*) que ilustra os dados e oferece o controle das variáveis de manipulação do ensaio.

Experimentos reais normalmente incluem os seguintes módulos para permitir uma interação com usuários: sistema de aquisição de dados e gerenciamento de experimentos. Estes últimos, por sua vez, na maioria das vezes são implementados usando ferramentas computacionais tradicionais como supervisórios (*SCADAs - Supervisory Control and Data Acquisition*) ou que oferecem funcionalidades semelhantes. Dentre estes podem-se citar os *softwares*: *LabVIEW* (LABVIEW, 2011), *Eclipse SCADA* (ELIPSE, 2011), *MatLab* e *Simulink* (MATHWORKS, 2011), *ScicosLab* (SCICOSLAB, 2011), *Comedi* (COMEDI, 2011), *Maxima* (MAXIMA, 2011) (semelhante ao *Maple* (MAPLE, 2011)). Destas ferramentas citadas, a maioria também proporciona desenvolvimento de modelos matemático-algorítmicos e suportam medidores virtuais, o que facilita a integração com equipamentos simulados. Também existem alguns *hardwares* de aquisição de dados (*DAQs*) populares como: diversos sistemas *DAQs* compatíveis com o *LabVIEW*, *USB 1208* (MC, 2011), *Festo EasyPort* (FESTO, 2011), *Arduino* (ARDUINO, 2011) (não necessariamente um *DAQ* mas que pode ser usado para tal), etc. A dissertação prévia do autor deste trabalho apresenta maiores detalhes neste assunto (SCHAF, 2006).

Experimentos podem ainda ser colaborativos, isto é, podem possibilitar a colaboração/interação de usuários/grupos num mesmo experimento. Visando isto existe o conceito dos chamados laboratórios que podem ser descritos como um local virtual onde pesquisadores ou usuários podem fazer testes (laboratório) sem preocupação com sua posição geográfica, interagindo com colegas (outros usuários), acessando instrumentos,

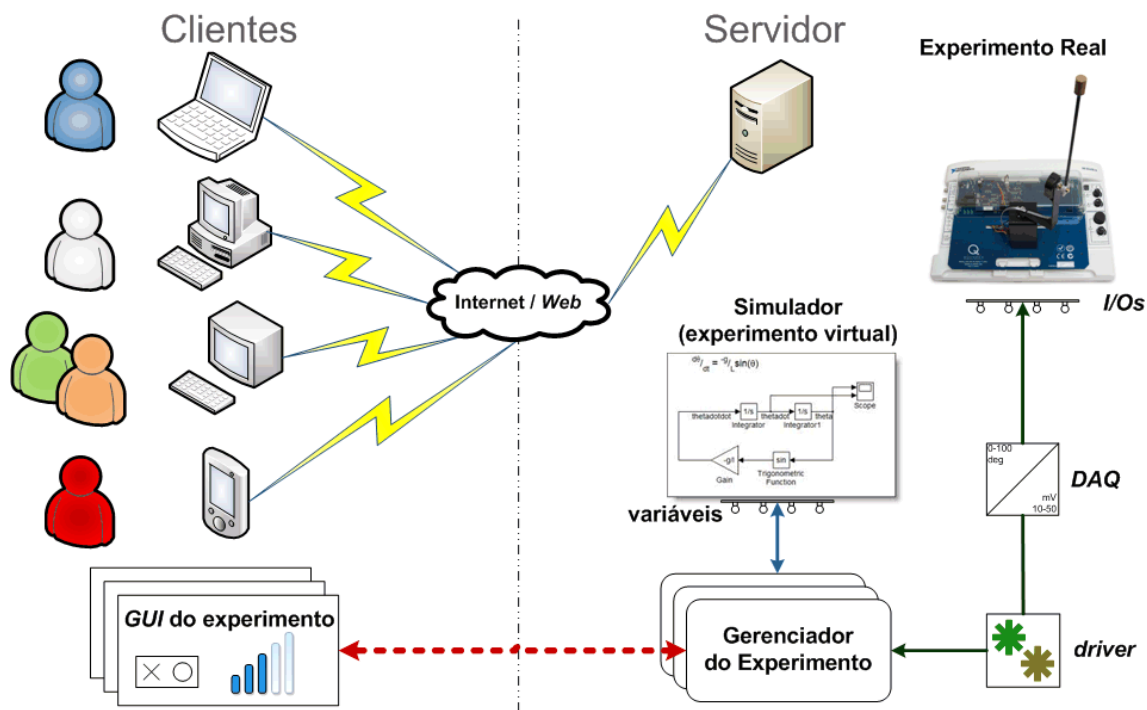


Figura 9: Experimentação remota - diagrama de estrutura.

compartilhando dados e recursos computacionais e acessando bibliotecas digitais. Este local virtual, também chamado de ambiente virtual está necessariamente associado com ferramentas de laboratórios ou experimentos remotos, isto é, é um ambiente virtual com suporte à colaboração que conta também com experimentos remotos para práticas de laboratório. A estrutura dos colaboratórios está ilustrada na Fig. 10.

Ferramentas e técnicas para projetos e desenvolvimento de colaboratórios entre pesquisadores com interesses similares numa linguagem comum são desenvolvidas há 20 anos e estão bastante documentadas na comunidade científica. Um exemplo desses estudos pode ser encontrado no trabalho de ALBRECHTSEN (2004) ou em relatórios do projeto *SoC (Science of Collaboratories)* (BENDER, 2004).



Figura 10: Colaboratórios - diagrama de estrutura.

Fontes: *Website* do projeto *SoC* (SOC, 2011) e (BENDER, 2004).

2.9 Redes Sociais

O próprio nome sugere que estas redes possuem caráter social, isto é, têm a finalidade de aproximar e possibilitar interações entre indivíduos dispersos em um sistema (ambiente) que gerencia mensagens e outras diversas mídias eletrônicas. Usuários de redes sociais podem ser vistos como nodos de uma estrutura interconectada na qual estes nodos estão ligados por: afetividade, amizade, interesses comuns, relações das mais variadas, conhecimentos, crenças, prestígio, localidades, atividades, grupos, etc.

Embora explicitamente estas redes não tenham nenhuma finalidade educacional, elas possuem uma característica importante na criação de laços sociais. Estes laços por sua vez promovem o contato (comunicação), a troca de interesses comuns (compartilhamento) que são estritamente necessários para a cooperação e a colaboração de usuários pela rede. Portanto, características sociais aplicadas a ambientes de ensino-aprendizagem, treinamento e também de trabalho são essenciais e parte psicológica intrínseca do relacionamento humano. Esta característica existe comumente no dia-a-dia interpessoal e precisa ser *transportado* para os ambientes computacionais, o que já é notoriamente conhecido pelas diversas implementações de redes sociais disponíveis na *Web*, como: *Facebook* (FACEBOOK, 2011), *Orkut* (ORKUT, 2011), *LinkedIn* (LINKEDIN, 2011), *Twitter* (TWITTER, 2011), etc.

2.10 Entretenimento *Online* (MMOGs)

Ramo potencialmente mais lucrativo e evoluído dos “ambientes virtuais” e possivelmente também das simulações, os jogos eletrônicos *online* para “grandes massas de jogadores”, os *MMOGs* (*Massive Multiplayer Online Game*), são uma categoria de *MUVEs* (*Multi-User Virtual Environments*) voltados ao entretenimento. Estes jogos se caracterizam por unirem muitos usuários distribuídos na rede sob um único ambiente lúdico no qual os participantes ou grupos (times) cooperam e/ou colaboram, ou ainda disputam por objetivos competitivos entre si ou em oposição a adversários simulados.

Os jogos possuem as mais diversas variantes: ação, aventura, estratégia, *RPG* (*Role Playing Game*), simulação, desafios (*puzzle games*), de tabuleiro, de cartas, de questionários (*trivias*), etc. Alguns ainda mesclam estas variantes. Uma derivação, por exemplo, dos *MMOGs* são os *MMORPGs* (*Massive Multiplayer Online Role-Playing Games*). Estima-se que o mais popular *MMORPG*, *World of Warcraft* (WOW, 2011) possua cerca de 12 milhões de usuários cadastrados.

Sabe-se que jogos de: ação com o domínio do estilo atirador (*shooter*); *RPG*; e estratégia dominam os *MMOGs*, mas há também uma grande comunidade de desenvolvedores e adeptos de jogos de simulação que têm muitas vezes finalidades menos competitivas. Estas simulações variam desde manuseio de aeronaves, com o clássico *Microsoft Flight Simulator* (MICROSOFT, 2011a) até o simulador de indivíduos, com a série *The Sims* (THESIMS, 2011).

Há inclusive associações de redes sociais com *MMOGs*, mais popularmente de simulação e de *trivias*, onde usuários colaboram e competem usufruindo os contatos pessoais previamente estabelecidos na rede social.

2.11 *Serious Games*

Ainda no ramo de jogos eletrônicos, destaca-se o desenvolvimento de jogos empregados tanto na educação quanto no treinamento, os chamados jogos “sérios” (*serious games*) (MICHAEL; CHEN, 2005). Estes são projetados com o intuito de oferecer um problema para a solução pelo(s) usuário(s) e, muitas vezes, sacrificam o caráter de entretenimento para focar “seriamente” no seu propósito principal, apesar de se tratar ainda de um jogo. Segundo ZYDA (2005), *serious games* são um desafio mental competido usando um computador de acordo com regras específicas que usam o entretenimento para alavancar o treinamento, a educação, a saúde, políticas públicas e objetivos estratégicos de comunicação. Alguns autores classificam *serious games* voltados ao treinamento de jogos de ensino (*game-learning*).

O termo ainda gera controvérsia entre pesquisadores desde sua (re-)criação em 2002, e muitos acreditam que esta classificação de jogos seria melhor definida por *smart gaming*. A finalidade destes jogos é vista pelos pesquisadores como uma mescla entre educação e entretenimento (*edutainment*).

Serious games podem ser vistos como simulações que imitam jogos, mas correspondem a eventos e processos avessos aos jogos. Os “jogos”, por sua vez, são desenvolvidos para proporcionar o engajamento e a autoconsolidação de contextos, que, por consequência, motivam, educam e treinam os jogadores. As simulações usadas nestes jogos podem ser classificadas em quatro categorias: i. jogos de persuasão; ii. jogos de dinâmica de grupo/organizacional; iii. jogos de sanidade - usados para recuperar/reabilitar saúde mental e física; e iv. jogos de arte - usados para idealizar expressões artísticas.

2.12 **Mundos Virtuais Tridimensionais**

São designados assim os ambientes computacionais que simulam a realidade com “mundos” 3D. Eles estão presentes em diversos ramos: comercial, industrial, educacional e entretenimento. Subnicho dos *MUVEs*, os mundos virtuais 3D também são usados em *MMORPGs* e possuem milhares de implementações com tecnologias e performance gráficas que atingem inclusive produções cinematográficas de animação.

Estes mundos também são chamados por muitos pesquisadores de *metaversos*, ou seja, metauniversos, por expandirem as características físicas do “universo” em questão proporcionando a fusão de realidade física com espaços virtuais fisicamente persistentes. Ou seja, aumentam as possibilidades do mundo real através da virtualidade. O termo foi criado pelo escritor Neal Stephenson em 1992 no romance de ficção científica *Snow Crash* (STEPHENSON, 1992). Nele, humanos são representados numa metáfora de mundo virtual como *avatars* e interagem entre si e com agentes controlados por *software*, não diferente dos *metaversos* existentes hoje.

Especula-se, que no futuro, *metaversos* enriquecidos com conteúdo *Web* substituirão os tradicionais navegadores da *Web* de interface e estruturação bidimensional existentes atualmente. Posteriormente, na Seção 3.5, são apresentadas algumas implementações existentes de mundos virtuais 3D.

2.13 **Sistemas Multiagente**

Os sistemas multiagente (SMAs ou *MASs* em inglês) são sistemas compostos por múltiplos agentes, que exibem um comportamento autônomo, mas ao mesmo tempo inte-

ragem com os outros agentes presentes no sistema. Estes agentes exibem duas características fundamentais: serem capazes de agir de forma autônoma tomando decisões levando à satisfação dos seus objetivos; serem capazes de comunicação com outros agentes utilizando protocolos de interação social inspirados nos humanos, incluindo pelo menos algumas das seguintes funcionalidades: coordenação, cooperação, competição e negociação (WOOLDRIDGE, 2002).

Os SMAs constituem um campo de pesquisa relativamente novo nas ciências da computação. Segundo WOOLDRIDGE (2002), embora o início das pesquisas neste nicho tenha se dado nos anos 80, só em meados dos 90 elas ganharam uma notoriedade digna de destaque. Ao longo dos últimos anos pesquisas no campo dos SMA cresceram muito.

Um SMA é um sistema computacional em que dois ou mais agentes interagem ou trabalham em conjunto (pode-se dizer que colaboram) de forma a desempenhar determinadas tarefas ou satisfazer um conjunto de objetivos. As pesquisas e implementações de SMA estão focalizadas na construção de padrões, princípios e modelos/*frameworks* que permitam a criação de sociedades de agentes capazes de interagir convenientemente de forma a atingirem os seus objetivos (LESSER, 1999).

Um dos pontos essenciais para permitir a construção de sociedades de agentes, consiste no gerenciamento das interações e as dependências das atividades dos diferentes agentes no contexto do SMA, isto é, coordenar esses agentes. Assim, a coordenação desempenha um papel essencial nos SMA, pois estes sistemas são inerentemente distribuídos. Aliás, o tema designado genericamente por coordenação constitui um dos maiores domínios científicos da informática e ciências da computação. Os trabalhos científicos abrangidos por este domínio, frequentemente incluem aspectos conceptuais e metodológicos, mas também implementacionais, de forma a proporcionar a expressão e a implementação de aplicações distribuídas. Diversas metodologias de coordenação foram propostas por diferentes autores dividindo-se em dois grupos principais: metodologias aplicáveis em domínios contendo agentes competitivos (*self-interested*) - agentes preocupados com o seu bem próprio, e metodologias aplicáveis a domínios contendo agentes cooperativos - agentes que incluem uma noção de preocupação pelo conjunto (REIS, 2003). No primeiro caso a coordenação por negociação é a metodologia mais estudada na comunidade internacional.

Não obstante a metodologia de coordenação por negociação possui particular relevância e aplicação em mercados e os leilões eletrônicos. À coordenação de agentes cooperativos interessa estudar metodologias que permitam construir equipes/times de agentes. Por consequência, neste contexto são de particular relevância as metodologias que permitem definir uma organização estrutural da sociedade de agentes, a definição e troca de papéis, a definição e alocação de tarefas aos diversos agentes e o planejamento conjunto multiagente.

Os SMAs incluem diversos agentes que interagem ou trabalham em conjunto, podendo compreender agentes homogêneos ou heterogêneos. Cada agente é basicamente um elemento capaz de resolução autônoma de problemas e opera assincronamente (independentemente) em relação aos outros agentes. Para que um agente possa operar como parte do sistema, é necessária a existência de uma infraestrutura que permita a comunicação e/ou interação entre os agentes que compõem o SMA, geralmente chamada de *framework* do SMA (vide Fig. 11).

Agentes componentes de SMAs possuem diferentes capacidades de percepção e ação no ambiente. Cada um possui uma esfera de influência distinta, ou seja, cada um será capaz de influenciar diferentes “partes” do ambiente (JENNINGS, 2000). Essas esferas de

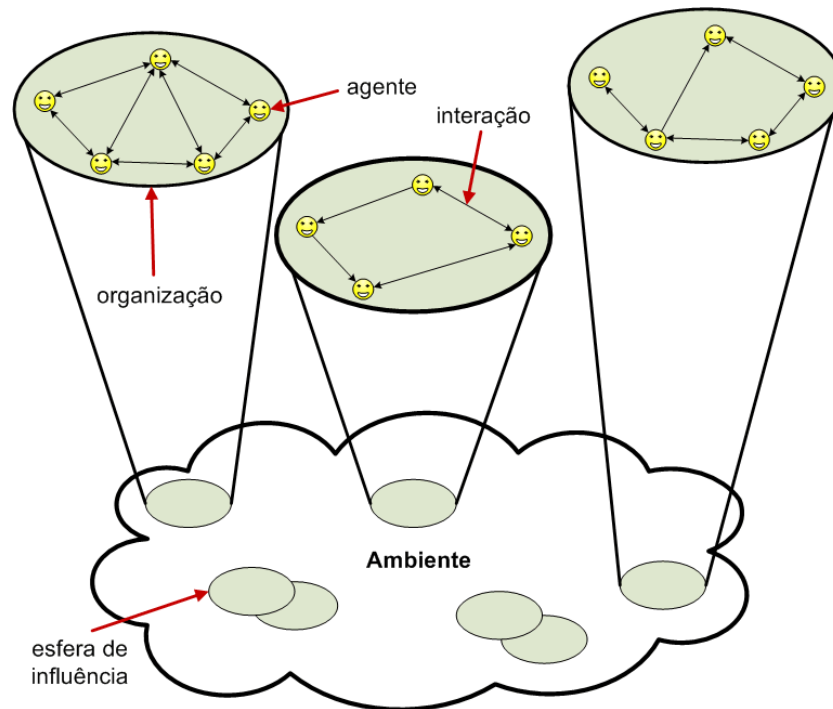


Figura 11: Sistemas multiagente - diagrama de estrutura.
 Fonte: Adaptado de (REIS, 2003).

influência podem coincidir dependendo das relações existentes entre agentes. Interações entre agentes que se encontram no mesmo ambiente podem ocorrer de diversas formas e como tais, interessa estudar os tipos e características de cada possível interação. A pesquisa em SMA está focada no desenvolvimento de princípios e modelos computacionais para construir, descrever, implementar e analisar as formas de interação e coordenação de agentes em sociedades de reduzida ou elevada dimensão (LESSER, 1999). O campo dos SMAs derivou do campo originalmente designado por Inteligência Artificial Distribuída (IAD), constituindo atualmente o núcleo deste campo.

3 ANÁLISE DO ESTADO DA ARTE

3.1 Introdução

A explosão evolutiva computacional que atualmente possibilita a divulgação de mídias eletrônicas de diversos tipos e propósitos também teve impacto na educação e nas metodologias de ensino que, com a popularização dos computadores podem ser empregadas em diversos níveis educacionais. Entre as ferramentas mais utilizadas na organização e gerenciamento de materiais de ensino e controle estudantil estão os ambientes virtuais que permitem a flexibilização de meios de comunicação entre estudantes e professores e a disponibilização de material de ensino independente de local e de horário.

A experimentação também usufruindo da infraestrutura computacional evoluiu para os chamados experimentos remotos e laboratórios híbridos, onde realidade mista é geralmente empregada para completar ou complementar equipamentos e componentes de laboratório reais. Desta forma a prática laboratorial, antes restrita pela presença tanto dos estudantes quanto dos experimentos em instituições de ensino ou de treinamento, pode ser acessada remotamente por usuários distribuídos tanto localmente quanto temporalmente (fora de horários tradicionais de funcionamento). Isto requer obviamente um desenvolvimento de um sistema (pode-se dizer de automação) para controlar tanto o acesso, manipulação e visualização quanto a segurança dos equipamentos envolvidos.

Ferramentas de NTICs aplicadas na educação ou no ensino-aprendizagem também suprimiram algumas soluções para problemas sempre frequentes em instituições de ensino que não dispõem de grande quadro docente e de assistência a colaboração entre estudantes. Entre elas o uso dos AVAs, em evolução/desenvolvimento em diversas instituições, tem sido encorajado tanto pelas universidades e outras instituições educacionais e de treinamento, quanto pelo próprio Ministério da Educação (MEC) (vide Seção 3.3).

A história dos ambientes com suporte à colaboração virtual, aqui chamados de AVCs tem o mesmo ponto de início do CSCL. Três projetos foram pioneiros em CSCL e alavancaram avanços de pesquisa em AVCs, são eles: o *ENFI* (*Electronic Networks for Interaction*) de 1985 da Universidade de Gallaudet (Washington, EUA); o *CSILE* (*Computer-Supported Intentional Learning Environments*) da Universidade de Toronto (Canadá); e o *Fifth Dimension Project* da Universidade da Califórnia (San Diego, EUA) (STAHL; KOSCHMANN; SUTHERS, 2006). Todos os três envolviam e exploravam o uso de tecnologia para melhoria do ensino.

O projeto *ENFI* produziu alguns resultados de ferramentas de *software* para composição assistida por computadores, ou o chamado *CSCWriting* (BRUCE; RUBIN, 1993). A Universidade de Gallaudet atende estudantes com necessidades especiais de audição e, por isso, muitos deles apresentam problemas de comunicação na escrita. O objetivo deste projeto foi engajar estudantes a escrever de maneiras novas: induzi-los à ideia de escre-

ver com uma “voz” e ter a plateia em mente. As tecnologias desenvolvidas neste projeto hoje são consideradas rudimentares devido ao avanço tecnológico, mas para a época foram inovadoras e de grande valia. Salas de aula especiais foram construídas com mesas e computadores colocados em círculo. *Softwares* que lembram os *chats* de hoje foram desenvolvidos para permitir que estudantes e seus instrutores pudessem conduzir discussões via texto.

Outro projeto pioneiro que influenciou pesquisas de AVCs foi desenvolvido pelos pesquisadores da Universidade de Toronto - Canadá (SCARDAMALIA; BEREITER, 1996; BEREITER, 2002), onde a “motivação” para o aprendizado nas escolas foi considerada prioridade. Eles compararam o aprendizado (tradicional) de salas de aula com o que ocorre em comunidades construtivas de conhecimento (*knowledge-building communities*), como as comunidades de pesquisadores que surgem em meio a problemas em linhas de pesquisa. No *CSILE*, conhecido mais tarde pelo seu “fórum de conhecimento”, eles desenvolveram tecnologias e métodos pedagógicos para reestruturar o aprendizado tradicional e o de comunidades de conhecimento construtivas. Como no projeto *ENFI*, o *CSILE* procurou fazer com que a escrita se tornasse mais significativa engajando estudantes na produção de textos em conjunto. Entretanto, os textos produzidos em cada caso foram diferentes. No projeto *ENFI* os textos eram de conversações, produzidos espontaneamente e geralmente não eram preservados ao final da aula. Textos produzidos no *CSILE*, no entanto, eram arquivados e tinha formas literárias científicas convencionais.

Assim como o *CSILE*, o projeto *Fifth Dimension (5thD)* começou com o interesse de melhorar habilidades de leitura (COLE, 1998). Iniciou como um programa de reforço na Universidade de Rockefeller nos EUA. Quando o laboratório de cognição humana comparativa (*Laboratory of Comparative Human Cognition - LCHC*) mudou-se para San Diego, o *5thD* foi reelaborado num sistema integrado, composto em sua maioria de atividades seletas baseadas em computadores para enriquecer habilidades de leitura e resolução de problemas para estudantes. O jogo de tabuleiro “labirinto” (*maze*), com diferentes salas representando atividades específicas, foi introduzido como mecanismo para indicar progressos de estudantes e coordenando a participação com o *5thD*. O trabalho estudantil era assistido por monitores (estudantes mais avançados) e por voluntários da escola de educação.

Todos esses projetos tinham o objetivo em comum de tornar a educação mais orientada, de forma a fazer mais sentido aos estudantes. Todos os três também empregaram computadores e recursos tecnológicos para atingir seus objetivos usando métodos sociais de atividades organizadas para a instrução. Desta forma, foi criada a base de pesquisa para subseqüentes e emergentes trabalhos com *CSCL*.

Uma das primeiras tentativas de criação de uma rede global de ensino foi proposta no projeto *Wallenberg Global Learning Network (WGLN, 2010) (WGLL)*, de 1999, proposta pela *Swedish Learning Lab (SweLL)* com os seguintes participantes: Uppsala University (Suécia), *Royal Institute of Technology* (Reino Unido), *Karolinska Institute* (Suécia) e a Universidade de Stanford (EUA). Aderiram ao projeto, em 2000, a alemãs: Universidade de Hannover, Universidade de Braunschweig e a Braunschweig School of Arts (TUTTAS; WAGNER, 2001). Este projeto já se encontra na terceira geração, *WGLL III* e tem a missão de desenvolver ambientes de aprendizagem mais eficientes usando inovações de NTICs e disseminar os resultados na comunidade científica.

O notável *MIT (Massachusetts Institute of Technology)* com sua rede de experimentos, batizada de *iLabs*, adere às tendências da evolução do sistema de ensino. Toda uma infraestrutura foi desenvolvida para o assim chamado aprendizado enriquecido por ha-

bilitação tecnológica (sigla *TEAL* - *Technology Enabled Active Learning*). Algumas colaborações do *MIT* contam com parceiros do peso da *Microsoft Research* e possuem resultados dos mais variados (vide Seção 3.5.2.1 para um dos resultados).

Outro projeto de eminente impacto na comunidade científica foi o *PEARL* (PEARL, 2010) (*Practical Experimentation by Accessible Remote Learning*), criado com financiamento da *European Commission's of Information Society Technologies*, em 2000, que durou três anos. Um consórcio de instituições de ensino, constituído por: *Open University* (Reino Unido), *Trinity College* (Irlanda), Faculdade de Engenharia da Universidade do Porto - FEUP (Portugal) e *University of Dundee* (Escócia), e, como parceira industrial, a empresa Zenon da Grécia se engajaram no projeto. Cada instituição envolvida no projeto possuía um experimento remoto distinto, abrangendo as seguintes áreas: bioquímica, física fundamental, inspeção visual e eletrônica digital.

Vale ressaltar também o projeto *MARVEL* (MARVEL, 2010) (*virtual laboratory in Mechatronics Access to Remote and Virtual E-Learning*) de grandes proporções que teve financiamento da Comunidade Européia e contou com os seguintes parceiros em diversos ramos: Universidades do Porto e Faculdade de Engenharia do Porto (Portugal) (Portugal), Universidade de Bremen (Alemanha), o colégio técnico de Delmenhorst (Alemanha), o Instituto Técnico do Chipre (HIT - Chipre), o *West Lothian College* (Escócia), as empresas Zenon (Grécia), FESTO e BMW (Alemanha), a autoridade de qualificação *Scottish Qualifying Authority* (Escócia), e o parceiro associado *Haute École Valaisanne* (Suíça) (MÜLLER; FERREIRA, 2004) (vide Seções 3.4.5 e 3.4.6 para alguns resultados).

Mais recentemente, o consórcio *RExNet* (*Remote Experimentation Network*) (ALVES et al., 2005; HINE et al., 2007; REXLAB, 2010), o qual teve como um dos resultados a dissertação de mestrado do autor (SCHAF, 2006) deste trabalho, uniu dez instituições de ensino, sendo cinco latino-americanas e cinco europeias; são elas: Universidade Federal de Santa Catarina (UFSC) e Universidade Federal do Rio Grande do Sul (UFRGS), do Brasil; Universidade Católica de Temuco (UCT) e PUC do Chile (PUCC), do Chile; Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM), do México; Universidade do Porto (UP) e Instituto Politécnico do Porto (IPP), de Portugal; Universidade de Dundee (UD), da Escócia; Universidade de Bremen (UB) e Universidade Técnica de Berlim (TUB), da Alemanha (ALVES et al., 2005). O consórcio teve como objetivo a cooperação de informação entre os parceiros e a colaboração de recursos laboratoriais. Alguns conceitos também foram desenvolvidos como o “real valor adicionado” (*added value*) ao ensino pelo emprego de laboratórios remotos. Esse seria resultado do incremento valor educacional - incluindo colaboração entre estudantes de diversas instituições diferentes e a partilha de práticas pedagógicas de sucesso; menos os custos decrescentes envolvidos - pelo compartilhamento de experimentos e reutilização de *softwares* desenvolvidos pelos participantes do consórcio (GOMES; GARCÍA-ZUBÍA, 2007).

3.2 Tecnologias na Educação

Entre as chamadas “tecnologias” na educação ressaltam-se as voltadas para disponibilizar e tornar mais interativos os materiais didáticos. Também seguem as linhas de evolução previamente mencionadas: digitalização e também virtualização; personalização; colaboração e co-autoria; distribuição e mobilidade; dinâmização e flexibilização. Assim podem-se apontar alguns marcos que ocorreram historicamente para o aparecimento das “tecnologias”: i. popularização do computador como ferramenta multiuso pessoal; ii. popularização do acesso à Internet; iii. maior capacidade gráfica e computacional dos

computadores; iv. popularização das ferramentas colaborativas na Internet (*Web 2.0*); e v. novo salto de capacidade computacional e de taxas de conexão na Internet. Aliados a este último marco ainda pode-se adicionar a mobilidade traduzida por dispositivos que podem ser usados para acesso à Internet.

Assim, inicialmente computadores que podiam ser usados para simples simulações ou ainda cálculos que auxiliavam o ensino foram empregados para disponibilização de materiais de ensino-aprendizagem (tutoriais, *how-to's* e semelhantes) através de servidores de páginas que usavam *HTML* e simples *Scripts* ou *Applets Java*. Com o desenvolvimento de ferramentas mais sofisticadas e de sistemas de aquisição de dados mais populares foram desenvolvidos os primeiros laboratórios remotos no início da década de 90 o que também culminou com alguns projetos de colaboração entre instituições (como alguns mencionados na seção anterior).

Posteriormente, com o maior poder de processamento de computadores populares surgiram ferramentas que tornavam simples mídias de aprendizado mais interativas. O emprego do *Flash*, por exemplo, alavancou as mídias eletrônicas interativas disponibilizadas na *Web*. Nos últimos dez anos, tem-se visto o avanço de ferramentas que proporcionam visualização tridimensional com “tecnologias” como o *VRML (Virtual Reality Modelling Language)* e, mais recentemente, o *X3D (WEB3D, 2011)*. Há uma extensa diversidade de ferramentas que são usadas para modelagem, desenvolvimento, simulação de modelos virtuais 3D que estão disponíveis na *Web*¹.

Em conjunto com os avanços de tecnologia também se desenvolveram padronizações para a disponibilização e uso “oficial” de materiais de ensino. Neste nicho pode-se citar a criação do termo objetos de aprendizagem (*LO - Learning Objects*) que auxiliam a reusabilidade e o repasse de materiais educacionais entre instituições adotando padrões de organização e qualidade. O *SCORM (SCORM, 2010) (Shareable Content Object Reference Model)* foi criado pelo Departamento de Defesa dos EUA (*DoD*), em 1999, e adotado como modelo de referência de objetos de aprendizagem, de conteúdos compartilhados (*SCORM*) através da chamada rede avançada de aprendizagem distribuída (*Advanced Distributed Learning - ADL*) (BOHL et al., 2002; ADLNET, 2010). É bom lembrar que esses cursos levam em conta a língua e a metodologia de ensino da instituição, isto é, é difícil padronizar um material educacional para instituições de outros países. Há também nos EUA um comitê para padronização de tecnologias de ensino, chamado de (*Learning Technology Standard Committee - LTSC*). Este é um órgão que busca a regulamentação de materiais de estudo para serem reutilizados em cursos diferentes, mas com conteúdos referenciados.

O *MIT* disponibiliza na Internet diversos cursos gratuitos na forma de apostilas de formato *SCORM*. Cada curso contém os chamados metadados de objetos de aprendizagem (*Learning Object Metadata - LOM*) que irão ser gerenciados por um sistema instrucional de gerenciamento (*AVA*). No Brasil também há incentivo à produção e disponibilização de objetos de aprendizagem pelo MEC em seu portal *Web* (PORTALMEC, 2010).

A colaboração em nível de rede também atingiu o ensino-aprendizagem, assim a *Web 2.0* alavancou o *e-Learning 2.0*. Conteúdos e interações em diversos níveis permitem o chamado ensino-aprendizado híbrido que pode ser usado em complemento ou reforçando o ensino tradicional. Um claro indicativo desta nova visão são os *Wikis* e os avanços de redes sociais.

Prevê-se para o futuro que a partir do último salto tecnológico seja possível o uso

¹Uma lista atual de *softwares* com estas funcionalidades está presente na *Wikipedia* (WIKIPEDIA, 2011a).

de ambientes virtuais tridimensionais, como navegadores da *Web* ou, possivelmente, a *Web 3D*, e por consequência, o *e-Learning 3D*. Também, paralelamente, prevê-se maior suporte autônomo aos estudantes no aprendizado com a evolução da linha de pesquisa de sistemas tutores inteligentes (*Intelligent Tutoring Systems - ITSs*). Adicionalmente, *ITSs* podem integrar suporte à colaboração ativa através de sistemas cientes.

3.3 Ambientes Virtuais de Aprendizagem

Existem diversos conjuntos de ferramentas de *softwares* que auxiliam a organização e gerenciamento de materiais educacionais, os chamados AVAs. Essas plataformas podem não oferecer todos os meios necessários para se construir um curso à distância, mas agilizam o processo de desenvolvimento e organização de um curso.

Dentre as mais populares e difundidas implementações de AVAs estão: o *MOODLE* (MOODLE, 2010), o *Claroline* (CLAROLINE.NET, 2010), o *Stud.IP* (STUDIP, 2010), o *TelEduc* (TELEDUC, 2010), e o *AulaNet* (GROUPWARE@LES, 2010), estes são *FOSSs* (*Free Open Source Softwares*); o *Blackboard* (BB, 2010), o *WebAssign* (WEBASSIGN, 2010), e o *WebAula* (WEBAULA, 2010), de licença comercial.

AVAs são implementados usando linguagens de programação simples baseadas na *Web*, como o *PHP* e armazenamento e estruturação de informação em bancos de dados *SQL* ou semelhantes. O que difere uma implementação da outra é o foco da apresentação (interface) empregada em cada um. Alguns são mais focados na organização de materiais de ensino e outros na organização geral de cursos; mas em quase sua totalidade eles são estruturados de forma a possibilitar e suportar a colaboração e a troca de informações entre os usuários (tanto tutores-estudantes quanto estudantes-estudantes).

Martin Dougiamas (DOUGIAMAS; TAYLOR, 2003), iniciou o desenvolvimento do *MOODLE* em 1999, *Modular Object-Oriented Dynamic Learning Environment*, e idealizou este AVA com a metodologia chamada de construtivismo social. O construtivismo afirma que o conhecimento é construído (criado ou recriado) na mente do aprendiz e não transmitido somente por professores e material educacional (que são vistos apenas como facilitadores) (LEIDNER; JARVENPAA, 1995). O aprendiz instrui-se baseado em sua própria experiência. Do ponto de vista do professor, ele apenas cria um ambiente centralizado no estudante (aprendiz), que ajuda na construção do conhecimento baseado em experiência e habilidades existentes, em lugar de fornecer e publicar informações que ele (professor) pensa ser necessário para o conhecimento do estudante. A formalização da teoria do construtivismo é geralmente atribuída a Jean Piaget que contou com diversos trabalhos de escritores antigos famosos, como Immanuel Kant, Jean-Jacques Rousseau, John Dewey, Jerome Bruner e Lev Vygotsky (STEFFE; GALE, 1995). O construtivismo social seria o apelo interativo entre estudantes, ou seja, a círculos sociais colaborando para um aprendizado geral.

O *MOODLE* é atualmente o AVA com o maior número de sites registrados, isto é, instrutores ou instituições de ensino que usam esta implementação com cerca de 50.000. Tanto o MEC quanto a UFRGS e outras Universidades adotaram o *MOODLE* como ambiente padrão para repositório de materiais de ensino e sistema de cursos à distância.

Ainda existem outros AVAs que não estão disponíveis fora da instituição de desenvolvimento, como o *EDUCA*, da Universidade Católica de Temuco, no Chile; o *Parla!*, usado pelo SENAI; o *Instructional Design*, usado pela PUC-Rio; o *Nou-Rau*, desenvolvido pela Unicamp; o *ROODA* desenvolvido na UFRGS e muitos outros.

3.4 Experimentação Virtual/Remota

Existem diversos experimentos remotos nas mais variadas áreas, e nesta seção serão apresentados alguns de maior relevância para o tema deste trabalho. Um estudo anterior do autor, durante o mestrado (SCHAF, 2006), foi realizado que identificou e classificou um maior número de experimentos virtuais e/ou remotos.

Conforme a topologia disponibilizada para o acesso do experimento (recursos laboratoriais), podem-se caracterizar duas topologias da interface no lado do cliente: *thin*²- e *thick-client*. Um laboratório (experimento) remoto possui interface *thin-client* se o usuário para acessar o experimento não necessita de *software* nem *plug-in* adicional não convencional para controlar e visualizar o experimento. Esta categorização é notoriamente intuitiva pela tradução do termo do inglês que significa, “cliente leve”. Interfaces baseadas em “tecnologias” amplamente usuais incorporadas à maioria dos navegadores *Web*, como suporte a *Java*³ e *Flash*, ou ainda mais simples como *HTML* são consideradas *thin-client*. Caso contrário, se o usuário necessitar adicionalmente um *software* específico ou ainda *plug-ins* não convencionais em conjunção com o navegador estas interfaces são ditas *thick-client*.

3.4.1 VCLab

O *Virtual Control Lab (VCLab)* (VCLAB, 2010) é conjunto de experimentos organizados como um laboratório. Foi desenvolvido por pesquisadores da universidade do Ruhr em Bochum na Alemanha. Todos os experimentos do *VCLab* são construídos empregando *HTMLs* básicos, modelos projetados e simulados no *MatLab / Simulink*, *Applets Java*, e modelos tridimensionais virtuais *VRML*. O *link* (elo de ligação) entre a visualização 3D (dos modelos *VRML*) é feito através de um *Applet Java* chamado de *HotAnimApplet* que envia dados da simulação (executada no *MatLab / Simulink*) ao *plug-in VRML*. O caminho inverso é gerenciado pelo *HotAnimApplet*, ou seja, quando no *GUI* são acionados/pressionados “botões virtuais”. Outros *plug-ins* como o de visualização de gráficos 2D e o do *MatLab* estão embutidos na página *HTML* que é visualizada no cliente (estudante). Este necessita de um *Web Browser*, *plug-ins* necessários para visualização e configuração do experimento além *MatLab/Simulink* instalado, pois todo processamento da simulação é feita na máquina do cliente (SCHMID, 2000). Portanto, esta configuração é típica de *thick-client*.

Para a utilização do *VCLab* no currículo de engenharia pesquisadores da Universidade de Bochum criaram um curso de engenharia de controle baseado em mídias da Internet, chamado de *DynaMit* (DYNAMIT, 2010) (do alemão *Dynamische Systeme - Erlernen von Systemeigenschaften mit Multimedia-Mitteln*), que emprega: tutoriais, exercícios e experimentos virtuais que lembram (simulam) experimentos reais presentes nos laboratórios.

Na Fig. 12 estão ilustrados: (a) virtualização do ambiente de experimentação; e alguns dos experimentos desenvolvidos: (b) o controle de nível de três tanques interconectados, (c) braço manipulador flexível e (d) parte de um veículo de decolagem e aterrissagem vertical (*VTOL*) (SCHMID, 2000).

3.4.2 SimQuest

O *SimQuest* (JOOLINGEN; JONG, 2003; SIMQUEST, 2010; SIMQUEST-WIKI, 2010) é um sistema de autoria livre desenvolvido pela Universidade de Twente (Holanda)

²existe outro uso deste termo na computação.

³*scripts* e *Applets Java* suportados pelo *JRE (Java Runtime Engine)* da *Sun Microsystems*.

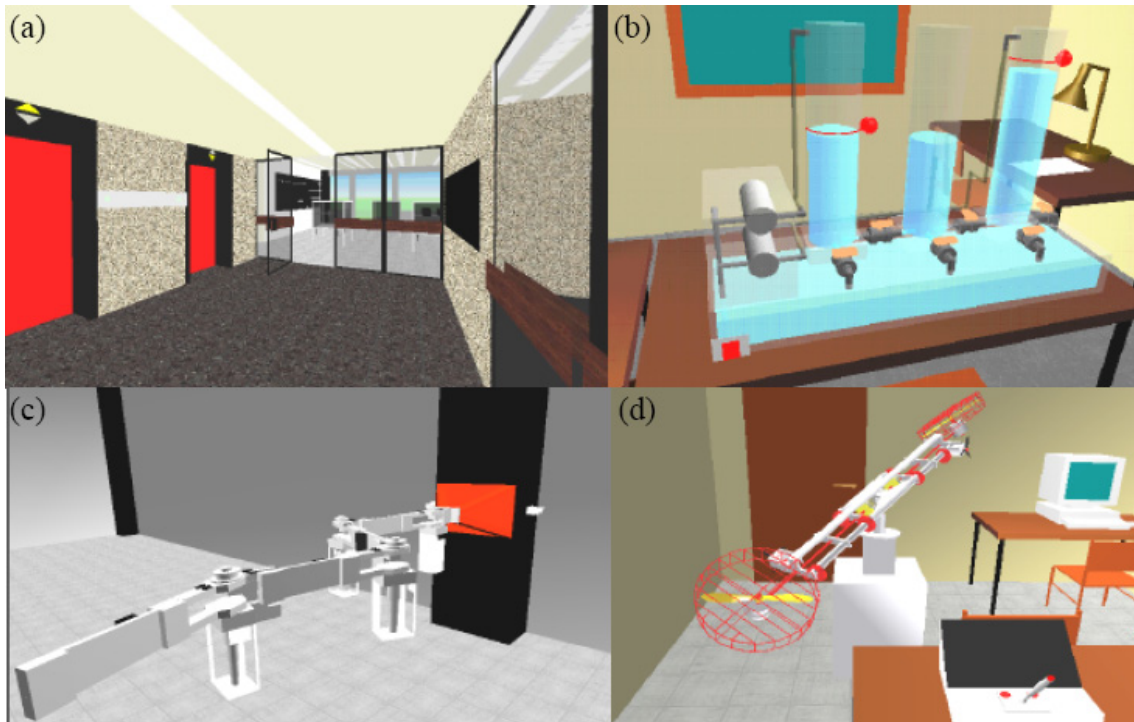


Figura 12: *VCLab* - modelos VRML de experimentos diversos.

Fonte: (SCHMID, 2000).

para projetar e criar ambientes de aprendizado baseados em simulações. Ele é particularmente apropriado para projetos interativos dinâmicos e a ambientes orientados para a descoberta (*guided discovery learning environments*).

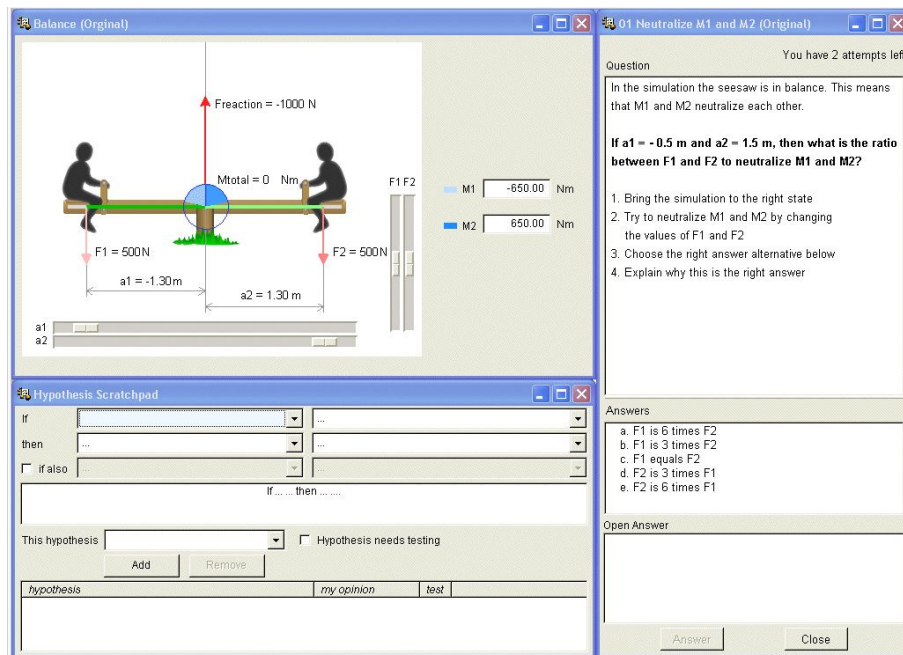


Figura 13: *SimQuest* - interface com exercício didático proposto ao estudante.

Fonte: (JOOLINGEN; JONG, 2003).

A biblioteca do *SimQuest* consiste em elementos que podem ser parte do ambiente de

ensino, como modelos de simulação, tarefas, explicações, etc. O ambiente usa modelos matemáticos para calcular valores da simulação. A interface do estudante permite interação com a simulação: modificando valores, explorando botões, etc. A interface *thick-client*, pois usa um *software* especificamente desenvolvido, apresenta comportamentos do modelo e também gráficos e outros tipos de representações gráficas para visualização (vide Fig. 13).

O sistema também permite a definição de tarefas, que são exercícios didáticos que exploram um determinado tópico educacional, e explicações que proporcionam aos estudantes mídias didáticas que ilustram dúvidas e têm por objetivo dar *feedback* aos estudantes. O ambiente (interface e gerenciador) é implementado usando a linguagem de programação em *Java* com a biblioteca gráfica *Swing* e é livremente (necessário somente um registro) distribuído na *Web*. Uma grande biblioteca de simulações (diversos modelos) também está disponível livremente no site do *SimQuest*.

3.4.3 Automatic Control Telelab

O *ACT* (*ACT*, 2011), criado pela Universidade de Siena na Itália, possui diversos experimentos de controle acessíveis abertamente para projetos de controladores em plantas reais (*CASINI; PRATTICHIZZO; VICINO*, 2004). A lógica do controlador é simulada por *software* enquanto a planta é real, esta configuração é chamada de *Software-in-the-Loop*. Todos os experimentos podem ser controlados tanto com controladores predefinidos, isto é, ajustando parâmetros, quanto por controladores com lógica desenvolvida pelo usuário. Os controladores desenvolvidos pelos usuários têm que seguir regras de um modelo (template) do *MatLab/Simulink* que então é enviado ao servidor que faz a conexão com parâmetros do experimento usando o pacote *Real Time Workshop* do *MatLab*. A interface do usuário utiliza a topologia *thin-client* com a utilização de somente aplicativos *Java* e o *Web Browser*. A transmissão de vídeo é de baixa qualidade através de *Applets Java*.

Entre os experimentos disponíveis destacam-se o controle de nível em tanques de água acoplados; o de controle de fluxo (baseado no mesmo equipamento de controle de nível); o de levitação magnética; um simulador de helicóptero com dois graus de liberdade; e o experimento de times de robôs móveis, ainda em desenvolvimento, construído com o uso do *Kit LEGO NXT* (*CASINI et al.*, 2009) (vide Fig. 14).

O experimento de levitação magnética consiste de um sistema de suspensão de uma esfera imantada. O objetivo da experiência é controlar a altura de suspensão da esfera ajustando a corrente elétrica de um eletroímã através da aplicação de tensão elétrica no mesmo. A força magnética utilizada só pode ser atrativa, portanto comandos de corrente negativa são desligados, isto é, não se admite repulsão magnética. O experimento de simulação de helicóptero consiste de um modelo criado pela *Humusoft* conectado a uma base sólida com duas hélices que controlarão o azimute e a elevação do modelo. Os motores CC (corrente contínua) que giram as hélices são controlados por *PWM* (*Pulse Width Modulation*). Os ângulos de rotação são medidos por sensores infravermelhos. É possível mudar o centro de gravidade do helicóptero através de um peso colocado no eixo horizontal que pode ser movido por um servomotor. Além do controle predefinido de controladores *PID*, pode-se selecionar entre controladores avançados de dois “tipos” *LQR* chaveados (*switching LQR controller*) e controle neural inverso preditivo (*predictive inverse neurocontrol*).

Em alguns experimentos há dois modos de utilização, o uso normal, e o uso do experimento com foco na competição estudantil (*CASINI; PRATTICHIZZO; VICINO*, 2005).



Figura 14: ACT - interface e experimentos diversos.

Fontes: (CASINI; PRATTICHIZZO; VICINO, 2004; CASINI et al., 2009) e *Website do ACT*.

Neste último, estudantes ou grupos de estudantes competem para desenvolver ou parametrizar um controlador que atinja o melhor desempenho de controle (menor *overshoot*, menor tempo de estabilização e acomodação). Esta competição realmente motiva estudantes e faz com que o aspecto investigativo de cada controlador (PID, lead-lag, ou controladores próprios) seja explorado com maior interesse.

Outro estudo de pesquisadores que criaram o ACT usa técnicas de rastreamento de imagem (*tracking*) via *Webcam*, SMAs e sistemas de supervisão para controlar times ou grupos de robôs construídos com o kit *LEGO Mindstorms* (BENEDETTELLI et al., 2009). Esta pesquisa apresenta uma solução de baixo custo para testes de estratégias de controle para sistemas multiagente. A ideia inicial usa um controle centralizado, mas o foco futuro é no controle descentralizado dos robôs móveis. Este estudo será utilizado para a criação do experimento de times de robôs móveis (CASINI et al., 2009), que ainda não está disponível para acesso no ACT.

3.4.4 CoLab e ModellingSpace

O *CoLab* é um ambiente que oferece desafios para o aprendizado investigativo colaborativo (*collaborative inquiry learning*) e suporta características cognitivas e processos de aprendizado que são necessários neste tipo de ambiente. O *CoLab* oferece experimentos remotos reais e simulações, modelagem e colaboração por quadros (*whiteboards*) (CARRERAS et al., 2005; COLAB, 2011).

Este laboratório de colaboração emprega topologia de cliente-servidor baseada em *Java* com um *software thick-client* especificamente desenvolvido para disponibilizar repositórios de materiais didáticos; ferramentas de colaboração; visualização de dados; e interface de experimentos (vide Fig. 15). O ambiente foi projetado para o ensino de ciências naturais no segundo grau e primeiros anos da Universidade. O conteúdo didático disponível está categorizado em quatro grupos: ciclo e gerenciamento de água, efeito estufa, mecânica e eletricidade. Cada uma destas categorias conta com experimentos remotos, material didático teórico, manuais do professor e instruções para estudantes.

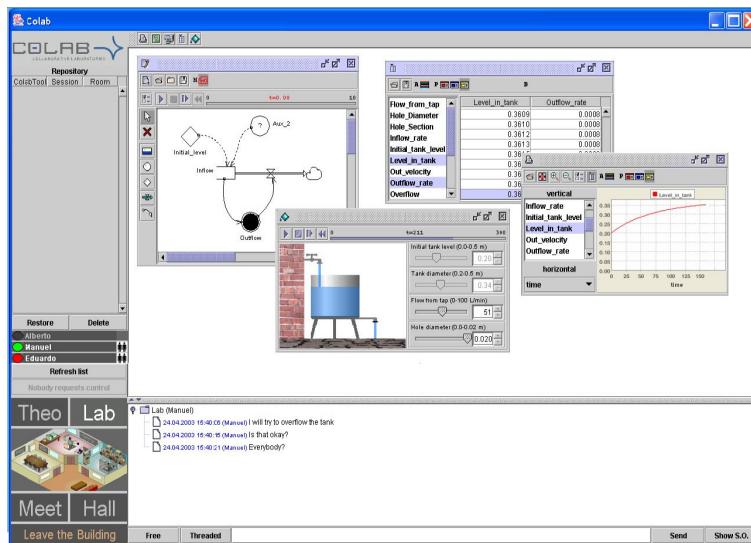


Figura 15: *CoLab* - interface do ambiente.
Fonte: (CARRERAS et al., 2005).

O *CoLab* foi desenvolvido por um consórcio europeu composto por cinco parceiros: Universidade Twente, Holanda (instituição coordenadora); Universidade de Amsterdam, Holanda; Universidade de Múrcia, Espanha; Studio Teos, Itália; e a Universidade de Kiel (IPN), Alemanha. O projeto foi patrocinado pela Comunidade Europeia e a implementação segue padrões de *FOSSs*.

Semelhante ao *CoLab*, o *ModellingSpace (MS)* (AVOURIS et al., 2003; MS, 2011) é um ambiente de aprendizado aberto que suporta colaboração síncrona e assíncrona de grupos pequenos de estudantes engajados na solução de um problema. Este ambiente foi desenvolvido e construído com base na experiência de ferramentas anteriores como o *ModelsCreator 2.0*, que foi usado no passado para o ensino multidisciplinar de ciências naturais em várias configurações educacionais.

A ferramenta principal é o *ModelEditor (ME)* que é acessado pelo professor e pelos estudantes. Este é um espaço de manipulação direto que foi desenvolvido para ser acessado em sua maioria por estudantes para construir modelos primitivos. O *ME* foi projetado para diferentes tipos de modelos, na maioria para estudantes da faixa etária de 11 a 16 anos. Os modelos podem ser semi- ou quantitativos relacionados a expressões matemáticas ou a mapas conceituais. O *ME* tem grande ênfase na visualização das entidades modeladas, suas propriedades e no suporte à investigação por jovens estudantes. A interface *thick-client* do *ME* está ilustrada na Fig. 16.

Num ambiente de solução de problema colaborativo, estudantes têm a seu dispor um conjunto abstrato de entidades primárias, nas quais eles constroem suas representações. Estas primitivas podem ser retângulos, elipses, quadrados e diferentes tipos de afirmações, etc. Portanto a base do conhecimento é o entendimento geral destas primitivas que são comuns a todos os usuários e são a base dos elementos/objetos da biblioteca do ambiente. Os objetos são representados por *XML* que permitem a associação do significado semântico e da validação da sintaxe.

A sincronização entre os participantes da colaboração é alcançada usando um protocolo ponto a ponto (*P2P - peer-to-peer*) sem a intervenção de um centralizador (servidor) de informação. O mecanismo é fundamentado num conjunto de agentes reagentes que sincroniza o *host* (mestre da conexão) baseados no modelo de estímulo e resposta. Assim,

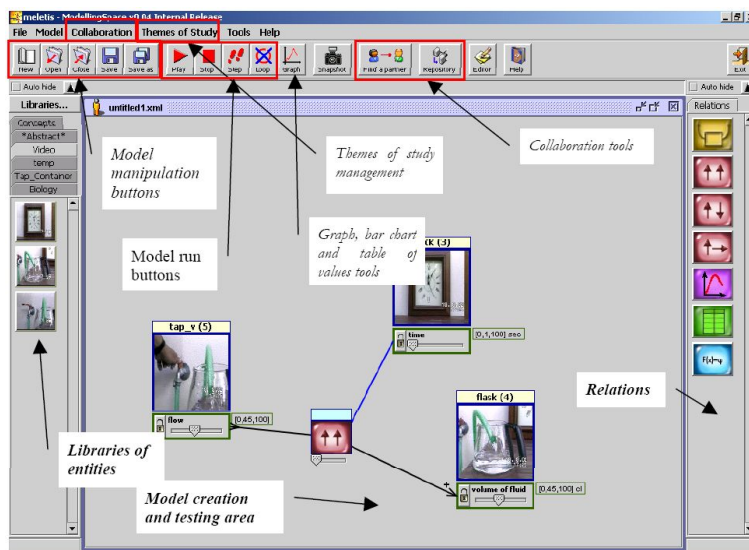


Figura 16: *ModellingSpace* - interface do ambiente.

Fonte: (AVOURIS et al., 2003).

na solução em conjunto de um determinado problema, cada objeto e relação é introduzida e age como agente reativo. O comportamento de cada agente depende se ele está no lado do usuário passivo ou no lado do usuário ativo. Caso esteja no lado do usuário ativo, este monitora os eventos gerados pelo usuário e que estão relacionados ao objeto em particular (por exemplo: movimento, mudança de propriedades, etc.).

A arquitetura de conexão do *MS* usa a topologia *thick-client* implementada em linguagem de programação *Java*. O ambiente contém várias ferramentas interoperáveis para expansão. Embora utilize geralmente *P2P*, também pode ser configurado para funcionar com um servidor que contém os seguintes serviços: (i) gerenciamento de repositórios; (ii) administração de usuários e escolas; e (iii) gerenciamento de grupos de colaboração.

3.4.5 HTI Solar Energy Laboratory

Criado pelo Instituto Superior de Tecnologia do Chipre (*HTI*), o experimento conta com painéis de energia solar para geração de energia elétrica e também para aquecimento de um sistema de água (MICHAELIDES; ELEFTHREIOU; MÜLLER, 2004; MARVEL, 2010). Cursos criados dentro do AVA *MOODLE*, chamados pelos autores de livros virtuais, possibilitam aos estudantes o conhecimento prévio do experimento e até questionários de identificação dos equipamentos envolvidos no laboratório e de processos relativos à geração de energia térmica e elétrica. A Figura 17 mostra as instalações do laboratório com os reservatórios e diversos equipamentos de acionamento (controle de fluxo) e sensores e também o diagrama de funcionamento do experimento (equipamentos).

Interessante notar que este experimento está totalmente integrado no AVA *MOODLE*, o que torna tanto o acesso integrado quanto o acompanhamento do material visitado pelos estudantes, respostas de questionários, etc. Estas informações podem ser usadas pelo professor para direcionar estudantes ou inclusive montar mapas de avaliação. O curso está configurado de tal modo dentro do *MOODLE* que o estudante só poderá efetuar o controle da planta quando diversas etapas (questionários) de familiarização com os equipamentos já foram suplantadas. Esta interface também apresenta características *thin-client*.

Outro aspecto interessante deste laboratório foi a preocupação com a locação de horários para usuários, isto é, um sistema de agendamento para utilização do experimento.

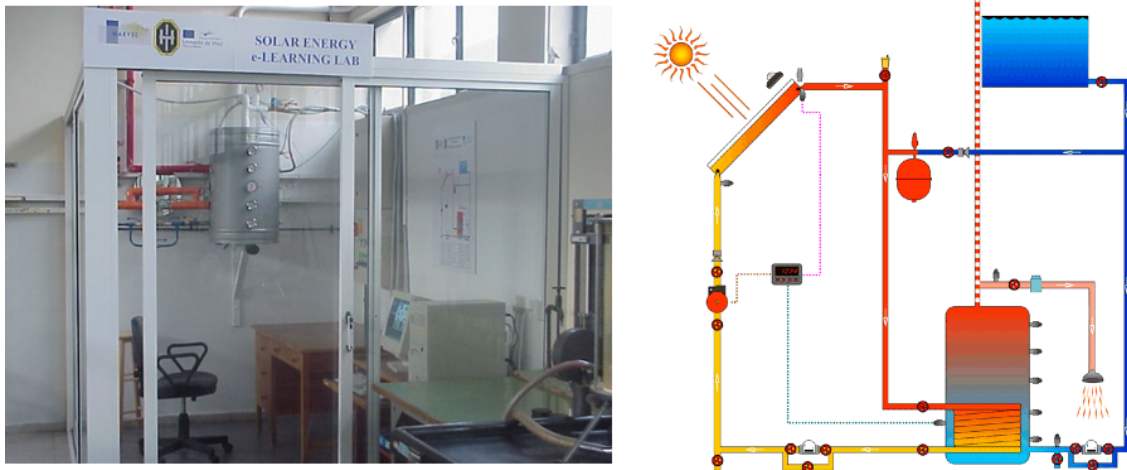


Figura 17: *HTI Solar Energy Laboratory* - experimento e diagrama didático de funcionamento.

Fonte: *Website* do laboratório.

Isto alavancou o desenvolvimento deste sistema que foi criado como módulo do *MOODLE* para integração com ambientes de ensino-aprendizagem.

3.4.6 *Mixed Reality Mechatronics Lab - DeriveSERVER e variantes*

O *deriveSERVER*, acrônimo para *Distributed Real and Virtual Learning Environment for Mechatronics and Teleservice*, foi desenvolvido pelo grupo de pesquisa *ArtecLab* da Universidade de Bremen, Alemanha, e também fez parte do projeto europeu *MARVEL* (MÜLLER; FERREIRA, 2004). Trata-se de uma bancada de realidade mista (vide Seção 2.6) onde equipamentos eletro-pneumáticos são utilizados tanto em simulações quanto em experimentos reais. O *link* entre a bancada virtual e real é alcançado usando o conceito de *hyper-bonds* (BRUNS, 2005) que permite que sinais virtuais digitais e reais (tensão e pressão de ar) sejam bidirecionalmente trocados. O conceito de *hyper-bond* usa hiperconexões (*hardware* e *software* para converter os sinais para representações lógicas digitais) e teoria de *Bond Graphs* (PAYNTER, 1960; KARNOPP; ROSENBERG, 1990) para modelagem matemática da ligação.

Os experimentos são criados com uma infinidade de equipamentos de eletro-pneumática que vão desde simples válvulas a simuladores de lógica digital (*softCLPs*). A interconexão entre real-virtual é feita graficamente usando réguas de conexão (em ambas as bancadas) que transfere os sinais entre os equipamentos ligados no mesmo borne. Desta maneira, pode-se, por exemplo, ligar uma válvula controlada por um botão normalmente aberto, presente na bancada virtual, a um cilindro pneumático real, presente na bancada real, através da passagem de sinais pela régua, que representa o *hyper-bond*. Este exemplo pode ilustrar o que é chamado pelos desenvolvedores de *air through the Internet* (FAUST; BRUNS, 2003), já que a arquitetura do *deriveServer* é distribuída e um usuário conectado ao sistema através da Internet pode acionar o cilindro pneumático através de “ar virtual” estando do outro lado do mundo.

A Fig. 18 ilustra as bancadas reais e sua contra partida virtual projetada nas telas. O projeto original foi expandido usando-se técnicas de componentes intercambiáveis durante um projeto em conjunto da Universidade de Bremen, com a UFRGS e o SENAI Mecatrônica de Caxias do Sul o qual se beneficiaria com a remodelagem do sistema para

melhorar o ensino de sistemas eletro-pneumáticos e de automação em seus laboratórios (SCHAF et al., 2007; SCHAF, 2006).

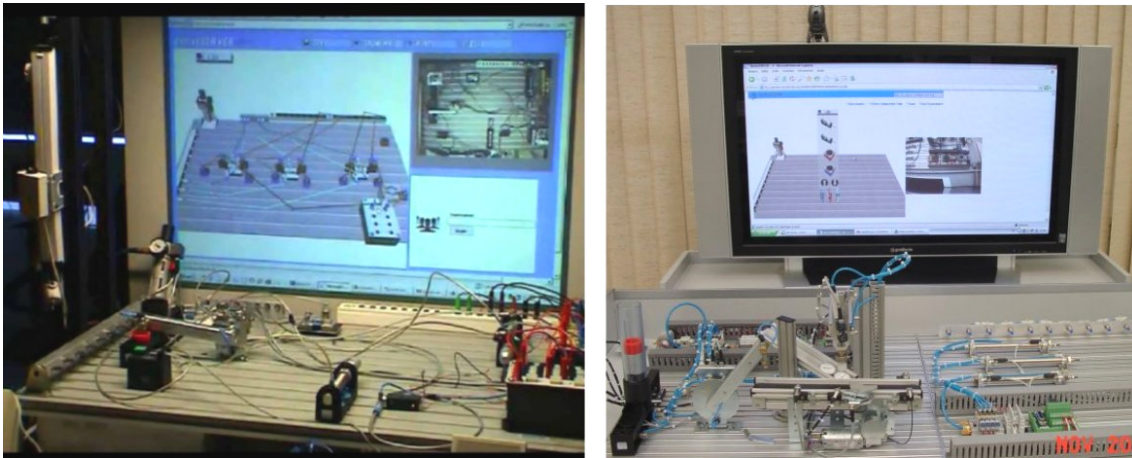


Figura 18: *DeriveServer* - interface e bancada montada no *ArtecLab* e no SENAI de Caxias do Sul.

Fonte: (SCHAF et al., 2007).

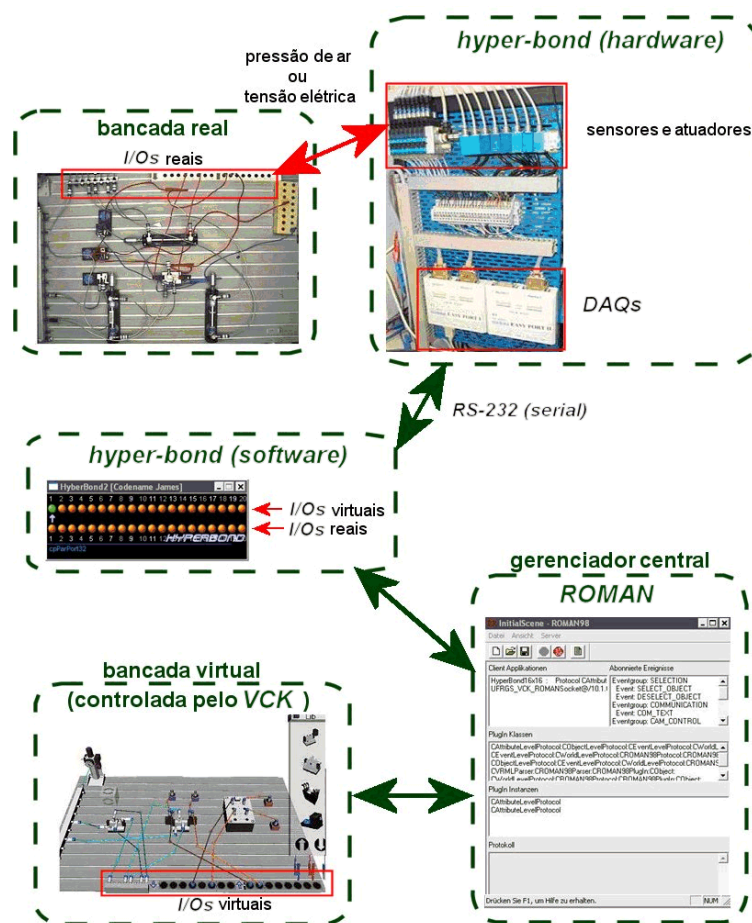


Figura 19: *Hyper-bond* do *deriveServer* - diagrama de interconexão, *hardware* e *software*.

Os softwares do *deriveServer* são inteiramente desenvolvidos usando a arquitetura de comunicação cliente-servidor. Um software chamado *ROMAN* (*Real Object Manager*)

desenvolvido usando a linguagem de programação *Visual C++* é o gerenciador das bancadas, pois centraliza todas as ações e comportamentos dos equipamentos, tanto virtuais como reais. Para se ilustrar a utilização do conceito de *hyper-bond* foi desenvolvido um *software*, que leva o mesmo nome do conceito, responsável pelo interfaceamento do *hardware*, isto é, ele é responsável em transmitir sinais (esforços) dos equipamentos virtuais aos reais e também dos reais aos virtuais. O *VCK (Virtual Construction Kit)* é uma aplicação desenvolvida em *Java* que carrega a interface virtual do experimento (*thick-client*). Os equipamentos virtuais utilizados são descritos em *VRML*. *Java Applets* e *Java Scripts* são utilizados para que as ações (entradas do *mouse* e teclado do cliente), efetuadas no manuseio dos equipamentos virtuais, possam ser capturados e transmitidos ao *ROMAN*. O *ROMAN* gerencia a cena do experimento virtual e, através de um protocolo de comunicação próprio, envia a descrição *VRML* da cena ao *VCK*. Uma câmera de vídeo (simples *WebCam*) disponibiliza imagens para visualizar os equipamentos reais conectados à experiência. A arquitetura assim com a implementação do *hyper-bond (hardware e GUI)* está ilustrada na Fig. 19.

O *deriveSERVER* também possibilita colaboração nos experimentos usando *CAVEs (Computer Automatic Virtual Environment)* para representação de ambientes multiusuário (MÜLLER et al., 2007). A *CAVE* permite a visualização de vários experimentos colaborativos que estão conectados via *hyper-bonds* remotos numa bancada central (vide Fig. 20). O sistema *CAVE* também permite a colaboração inter-*CAVEs* (vide Fig. 20), criando assim espaços de trabalho distribuídos e colaborativos baseados em *CAVEs*. O sistema emprega o *AVA MOODLE* com materiais didáticos e manuais de equipamentos para questões de ensino integrado ao experimento.

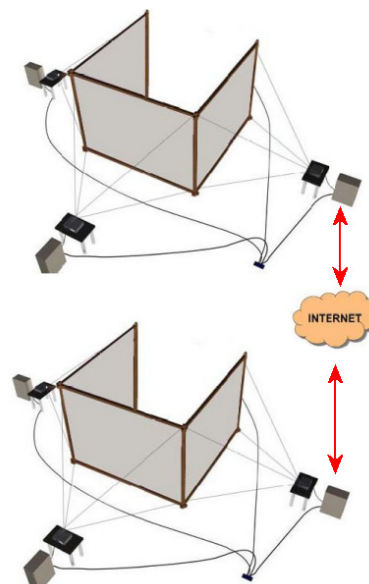
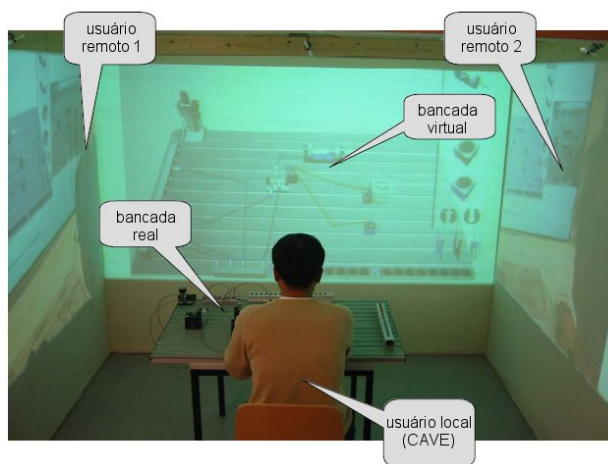


Figura 20: *CAVE* - exemplo de colaboração na experimentação e configuração multi-*CAVE* de espaços de trabalho distribuídos.

Fonte: (MÜLLER et al., 2007).

3.5 Ambientes/Mundos Virtuais Imersivos

Nesta seção serão apresentadas algumas implementações de maior representatividade para a abordagem desta tese que incluem um caráter imersivo aos seus usuários. Por imer-

sividade entende-se aqui a propriedade que engloba usuários e ambientes em uma única representação tridimensional, fazendo com que o usuário tenha o sentimento psicológico de presença virtual no ambiente em questão. Como o ambiente e representação do usuário é virtual, pode-se chamá-lo de simulação virtual, segundo a classificação da Seção 2.5. Obviamente, para fatores psicológicos estes ambientes/mundos devem lembrar (possuir analogia) com ambientes/mundos reais.

O modo de visualização destes ambientes, isto é sua interface gráfica para os usuários (GUI), é bastante comum no entretenimento eletrônico, popularmente conhecida no meio de jogos computacionais como *First Person Shooter - FPS* (tradução literal, atirador em primeira pessoa) devido à predominância e pioneirismo deste modo no ramo de jogos.

3.5.1 Active Worlds



Figura 21: AWEDU - interface do usuário no mundo virtual *iGarden*.
Fonte: Website do AWEDU.

Active Worlds (AW) é uma plataforma tridimensional acessível na *Web* para exploração de mundos virtuais criados por diversos usuários. O objetivo original do AW, em 1995 quando foi desenvolvido, era ser o equivalente 3D do *Browser* (navegador da *Web*) 2D convencional. Ao invés de criar páginas da *Web*, usuários poderiam construir escritórios, prédios ou áreas para disponibilizar produtos (SCHROEDER; HUXOR; SMITH, 2001; AW, 2011).

Desde 1999, há em conjunto com o AW o AWEDU (*Active Worlds Educational Universe*), que é divisão especial do AW para iniciativas educacionais no qual há um suporte para criadores de mundos virtuais com uma vasta biblioteca de objetos pré-definidos (vide Fig. 21).

O AW, assim como diversas outras implementações de mundos virtuais, tem fins comerciais. A exploração necessita além de um navegador (*software-cliente*), de um cadastro de usuário, por se tratar de um ambiente controlado onde há propriedade sobre objetos e porções de terra virtuais, além da tradicional identificação de usuários.

A comunicação entre usuários no *AW* se dá através de troca de mensagens de texto (*chat*). Esta implementação da empresa *Active Worlds Incorporated* - *AWI* já está na versão 5.0 e conta com diversos mundos virtuais criados por dezenas de instituições de ensino de vários países, sendo três do sul do Brasil⁴.

Há diversas outras implementações que competem no mesmo nicho de mercado do *AW*, entre elas pode-se citar: o *SmallWorlds* (*SMALLWORLDS*, 2011), que é mundo virtual desenvolvido em *Flash*, portanto pode ser executado a partir de navegador tradicional da *Web* com suporte para *Flash*; o *Entropia Universe* (*ENTROPIAUNIVERSE*, 2011), onde mundos são tratados como planetas; *IMVU* (*IMVU*, 2011), que é voltado para troca de mensagens instantâneas em ambientes 3D; o *Kaneva* (*KANEVA*, 2011), que é voltado somente a redes sociais, compartilhamento de mídias inclusive com associações com a rede social *Facebook*; e as outras implementações que serão abordadas nas próximas subseções.

3.5.2 *Open Wonderland*

O antigo “projeto *Wonderland*”, criado pela *Sun Microsystems* em 2008, foi abandonado pela *Oracle* que cessou o financiamento deste projeto após a compra da *Sun Microsystems* no início de 2010. O *Open Wonderland* (*OPENWONDERLAND*, 2011) é justamente a continuação, pela *Open Wonderland Foundation*, uma comunidade aberta de desenvolvedores, em caráter *FOSS*, deste projeto/implementação de mundos colaborativos sociais 3D imersivos. O objetivo do *Open Wonderland* é propiciar um conjunto de aplicativos (chamado de *toolkit*) para o desenvolvimento de ambientes colaborativos para educação ou comerciais que sejam interativos, de simulação para multiusuários, e altamente dinâmicos.

A implementação está baseada na linguagem de programação *Java* e funciona sobre plataformas previamente desenvolvidas, como o *RedDwarf Server* (*REDDWARF*, 2011) (antigo Projeto *Darkstar* também descontinuado pela *Oracle*), que tem o foco para *MMOGs*; o *Java 3D* para visualização e representação de simulações tridimensionais; e o *jVoiceBridge* (*JVOICEBRIDGE*, 2011) para gerenciamento de áudio no ambiente.

O *Open Wonderland* é disponibilizado sob licença *GPL* (*GNU-GPL*, 2011). Tanto o cliente como o servidor necessitam da instalação da máquina virtual *Java* (*JVM*) para sistemas operacionais que não sejam de linguagem nativa *Java*. O uso da *JVM* traz as características intrínsecas desta “tecnologia”, como a teórica independência de sistemas operacionais específicos e o vasto alcance e disponibilização dos aplicativos *Java*.

3.5.2.1 *3D Physics Lab*

O trabalho da Universidade Tecnológica de Graz, da Áustria, em conjunto com o *MIT* e da Universidade de Tecnológica de Curtin, da Austrália, apresenta uma ótima visão do emprego do *Open Wonderland* (na época do trabalho era Projeto *Wonderland*) na educação em conjunto com experimentação, o *3D Physics Lab* (*SCHEUCHER et al.*, 2009).

Integrando o *LabVIEW* para interfacear o experimento de força num dipolo magnético com o servidor (computador), previamente desenvolvido no âmbito dos *iLabs* (*ILABS*, 2011) do *MIT*, uma ferramenta chamada de *TEALsim* (*TEAL*, 2011) também desenvolvido num projeto do *MIT* e o *Open Wonderland*, foi possível alterar a interface tradicional deste experimento para uma interface imersiva e social. Desta maneira tanto o experi-

⁴segundo consta do site do *AWEDU* são elas: Universidade Católica de Pelotas, Universidade Comunitária do Estado do Rio Grande do Sul e Universidade do Vale do Rio dos Sinos - Centro de Ciências Humanas.

mento real pode ser visualizado (vídeo *streaming*) e controlado (via *Applet Java* do *LabVIEW*) quanto a simulação pode produzir efeitos no ambiente tridimensional (vide Fig. 22). Por se tratar de tecnologias compatíveis, pois todos empregam *Java*, a integração foi facilitada e experimentos com a mesma modelagem podem ser facilmente também integrados.

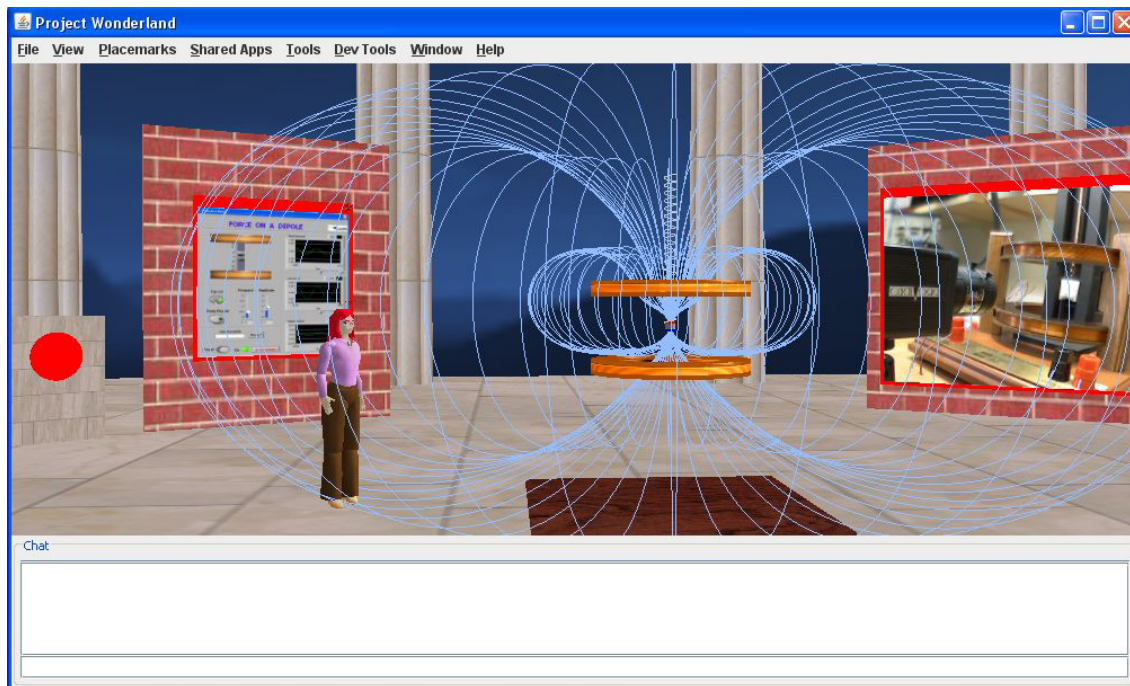


Figura 22: *3D Physics Lab* - uso do *Open Wonderland* como ambiente educacional.
Fonte: (SCHEUCHER et al., 2009).

Assim como o *TEALsim* existem outros simuladores que empregam a linguagem *Java* para descrever e simular os comportamentos dos modelos. Esses poderiam também ser empregados, como o *EasyJava (Ejs)* (ESQUEMBRE, 2004; EJS, 2011) e o *Processing* (PROCESSING, 2011).

3.5.3 *Second Life*

O *Second Life (SL)* (Second Life Inc., 2011), tradução literal “segunda vida”, funciona como um servidor de mundos virtuais muito popular criado pela empresa *Linden Lab*. O *SL* está organizado como um conjunto de mundos virtuais interconectados e estruturados como ilhas (ou parcelas de terra virtuais). O usuário que desejar explorar estes mundos virtuais deve efetuar inicialmente um cadastro e criar um “codinome” (*alias*) para sua representação (personificação) virtual, chamada de *avatar*. Cada residente pode interagir com diversos elementos virtuais e outros residentes que estiverem presentes (conectados e ao alcance) no mundo através de um *software*-cliente, chamado de *Viewer*, que disponibiliza a interface tridimensional e a conexão com o servidor do *SL*. O *Linden Lab* também oferece gratuitamente o *software SL Viewer*, embora existam uma infinidade de outros *viewers*, também gratuitos e alguns de código aberto, que podem ser usados para a conexão com o servidor do *SL*.

Embora o cadastro do residente e a exploração dos mundos seja gratuita, o desenvolvimento dos mundos não o é, i.e. o *Linden Labs* comercializa qualquer objeto virtual e somente usuários proprietários de terra virtual podem construir em suas parcelas (com

exceção de alguns espaços reservados para testes chamados de *sandboxes*). Não obstante, muitas dessas “ilhas” foram desenvolvidas com fins educacionais (EaD) por todo o tipo de profissionais do ensino. Nelas estudantes (com seus *avatares*) podem frequentar virtualmente aulas inclusive no ensino superior (RITZEMA; HARRIS, 2008).

3.5.3.1 Projeto SLOODLE

O uso do *SL* em combinação com o AVA (*MOODLE*) foi implementado pelo projeto *SLOODLE* (KEMP; KABUMPO, 2006; SLOODLE, 2011). Este projeto visa propiciar um ambiente com interface mais amigável (intrinsecamente social e imersivo) aos usuários para colaborarem e visualizarem material de ensino. Aspectos e experiências do pacote *SLOODLE* no *SL* foram abordados por KIRRIEMUIR (2007) que desenvolveu um mundo chamado *VirtuAlba* (vide Fig. 3.5.3.1) e empregou-o no ensino universitário no Reino Unido.



Figura 23: *SLOODLE* - encontro no *VirtuALBA* e *SLOODLE Moot*.

Fonte: (KEMP; KABUMPO, 2006) e Website do *SLOODLE*.

Para interconectar metaobjetos do *SL* (*MUVE* usado como base do projeto) com o *MOODLE* o projeto *SLOODLE* faz uso de *scripts* do *SL*, chamados de *Linden Scripts*, associados a objetos virtuais no *SL* que trocam informações com módulos do *SLOODLE* para o *MOODLE*. O pacote *SLOODLE* (objetos, *scripts* e módulos do *MOODLE*) é livremente distribuído e pode ser usado por qualquer usuário. A intercomunicação se baseia em trocas de informações usando chamadas *HTTP* e *XML-RPC* (*XML-RPC.COM*, 2011) do *SL* para o *MOODLE* (vide Fig. 24).

3.5.3.2 Magee Campus

A Universidade de Ulster, na Irlanda do Norte, desenvolveu diversos materiais educacionais e simulações utilizando o *SL* na parcela chamada de *Magee Campus* (MAGEE, 2011) que lembra o próprio campus físico (real) (CALLAGHAN et al., 2010, 2008).

Além de disponibilizar material didático de forma organizada, esta parcela de terra virtual do *SL* oferece simulações para práticas aos estudantes como um tutorial do funcionamento de computadores (e processadores), funcionamento de motores elétricos, funcionamento de circuitos básicos, etc (vide Fig. 25).

Também foi desenvolvida uma comunicação básica com equipamentos reais usando o *SL* como interface gráfica (*GUI*) e a capacidade do *SL* de conexão *XML-RPC*. Desta forma foram desenvolvidos uma manipulação remota de braços robóticos virtuais (repre-

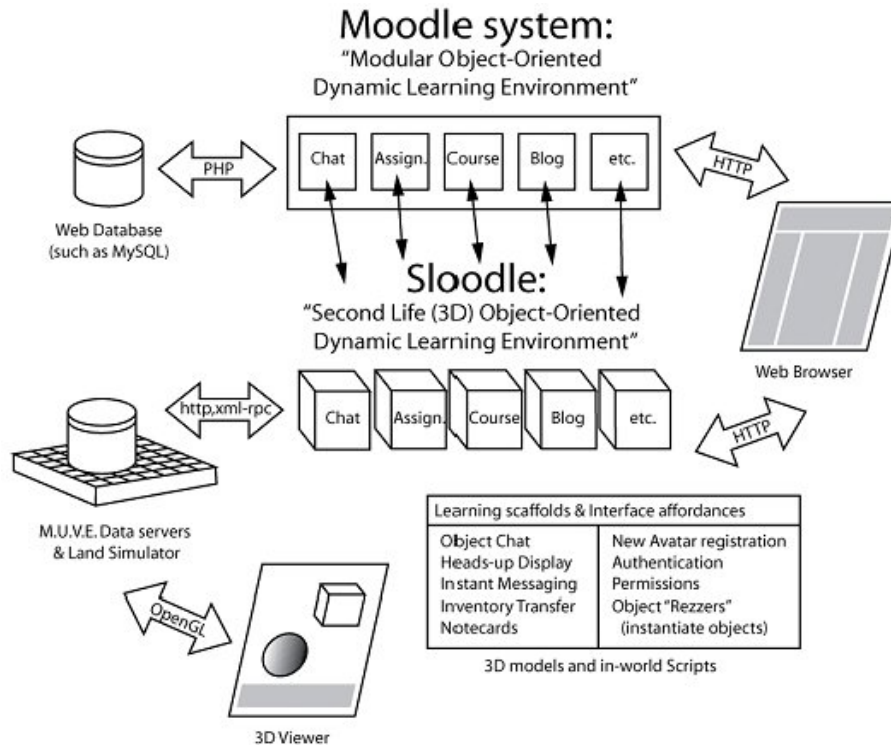


Figura 24: *SLOODLE* - diagrama do funcionamento e de trocas de informações.
Fonte: (KEMP; KABUMPO, 2006).

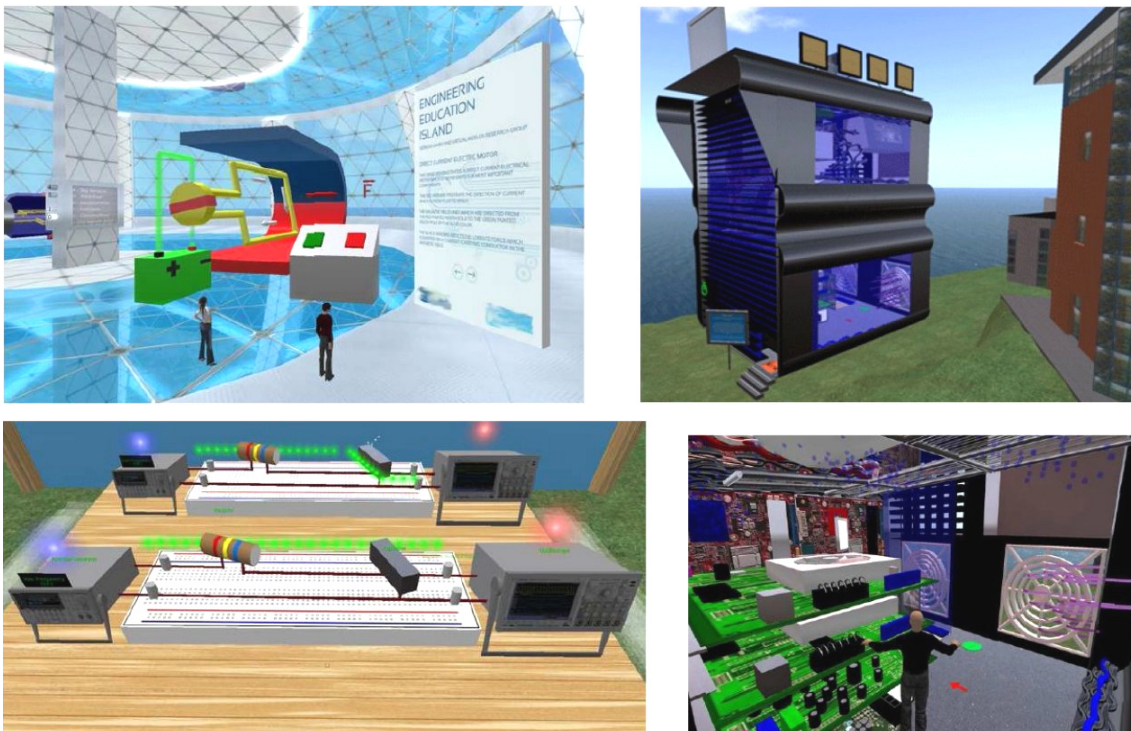


Figura 25: *Magee Campus* no *SL* - exemplos de simulações disponíveis.
Fonte: (CALLAGHAN et al., 2010).

sentados no *SL*) através de controles do videogame *Nintendo Wii*, chamado de *Wiimote*, e um experimento híbrido (real e simulado) do controle de uma máquina de lavar (vide Fig.

26).

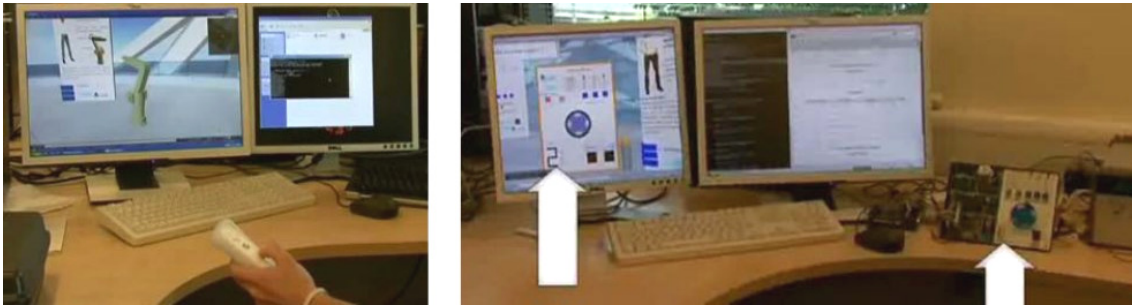


Figura 26: *Wii* e experimento híbrido de máquina de lavar.
Fonte: (CALLAGHAN et al., 2010).

3.5.3.3 *Second Lab*

O *Second Lab* (GARCÍA-ZUBÍA et al., 2010) é voltado para experimentos remotos reais usando como interface o *SL*. A conexão com sua arquitetura distribuída (ORDUÑA et al., 2009) de experimentos remotos, chamada de *WebLab Deusto*, é feita através de *Linden Scripts* usando chamadas *HTTP* e *XML-RPC*. O experimento desenvolvido no *Second Lab* é o chamado *SecBot*, um microbot autônomo (robô pequeno especificamente desenvolvido). Este é programado pelo usuário para executar uma série de movimentos através do carregamento de código usando o *SL* tanto para visualização quanto para configuração e execução do experimento. Após enviado o código de programação, a compilação é efetuada no servidor e, em seguida, é enviado ao microbot o código de execução (arquivo *HEX*). A sequência de execução do robô pode ser visualizada através de vídeo online (usando o protocolo de transmissão de vídeo *RTSP - Real Time Streaming Protocol*) (RFC2326, 2011) na interface, o *SL* (vide 27).



Figura 27: *Second Lab* da Universidade de Deusto - interface e experimento.
Fonte: (GARCÍA-ZUBÍA et al., 2010).

3.5.4 *OpenSim*

OpenSimulator ou *OpenSim* (OS, 2011) é uma plataforma de servidor para hospedagem de mundos virtuais de código aberto (*open source*), licença *BSD (Berkeley Software Distribution)* (WIKIPEDIA, 2011b) e livremente acessível (*FOSS*) na rede. O funcionamento do *OpenSim* é praticamente idêntico ao *SL*. Existe um servidor que concentra

todas as informações e um *software*-cliente (*Viewer*) responsável pela representação da interface 3D ao usuário. O servidor do *OpenSim* é compatível com o cliente do *SL* e o desenvolvimento dos mundos é praticamente idêntico.

O *OpenSim* também suporta os *Linden Scripts* para programação *in-World* (ou seja, no *metaverso*), além do uso de uma linguagem própria de programação *in-World*, a *OpenSim Scripting Language (OSSL)*. A comunidade de desenvolvimento do *OpenSim* orgulhosamente destaca que sua implementação é multiplataforma, suporta uma variedade imensa de *viewers* e possui interface de comunicação para uma variedade de protocolos.

O código-fonte da implementação está escrito em *C# (C sharp)* e, por isso, pode ser executado em sistemas operacionais (OSs) *Windows* via *framework .NET (dot net)* ou em outros OSs **ix (Unix e semelhantes)* através do *framework Mono (MONO, 2011)*.

Atualmente o servidor se encontra na versão alpha 0.7.0.2 e suporta os protocolos de comunicação *XML-RPC* e *REST (JSON/HTTP e XML/HTTP)*. O servidor do *OpenSim* suporta uma variedade de gerenciadores de bancos de dados onde armazena e concentra todas as informações dinâmicas necessárias para seu funcionamento, entre eles o *SQLite*, *MSSQL* e *MySQL*. Existem inclusive diversos *plug-ins* que podem ser carregados como módulos para que o *OpenSim* possa incorporar outras funcionalidades.

O servidor possui três modos de operação: *Stand Alone*, *Grid* e *Hypergrid*. No *Stand Alone* o servidor funciona isolado de outros *metaversos* (servidores existentes) dos outros dois modos onde o *metaverso*, criado no servidor, está interconectado com uma rede de *metaversos*. O modo *Hypergrid* difere do *Grid* pela ausência de um centralizador, isto é, o “mundo” é um conjunto de simulações (do *OpenSim*) com conexões fracas (*loosely connected*) que lembra a *hypermedia* da *Internet*. No modo *Grid* o “mundo” é um conjunto de simulações ordenadas como um mapa (mosaico) e cada porção quadrangular do mapa representa um *metaverso* em algum servidor. Este mapa é chamado de *OSGrid* e cada simulação deve estar registrada no servidor central do *OSGrid* que centraliza as informações de cadastro e de inventário dos usuários.

3.5.5 *RealXtend*

O *RealXtend (REALXTEND, 2011)* pode ser visto como uma expansão do *OpenSim* garantindo funcionalidades extras ao tradicional simulador ou também como uma implementação do *OpenSim* com módulos extras que dão, além de maior flexibilidade na criação de *metaversos*, também maior realidade aos cenários 3D.

O *RealXtend* é um projeto para criar uma plataforma de mundos virtuais de código aberto de forma a expandir características do *OpenSim* para incluir funcionalidades que não estão disponíveis no projeto original do *SL*. O projeto foi fundado pela *Oulu Innovation* da Finlândia, em cooperação pelas três companhias *Evocativi*, *LudoCraft* e *Playsign*.

Assim como o *OpenSim*, com o qual este projeto também colabora e no qual deriva seu código, o *RealXtend* é direcionada a usuários e grupos que querem desenvolver seus mundos e utilizá-los sem conectá-los a servidores de mundos enormes como o *SL*.

Diferentemente de outros *viewers* para *SL* e *OpenSim*, o *RealXtend Viewer* (livre, de código aberto sob a licença *GPL*) usa a *Object-Oriented Graphics Rendering Engine (OGRE) (OGRE3D, 2011)*, que é uma máquina de renderização 3D completamente livre (licença do *MIT*), escrita em *C++* e multiplataforma em oposição aos outros *viewers* que utilizam a *Havok Physics Engine* proprietária do *Linden Lab (SL)*. Em comparação, a *OGRE* proporciona renderização de sombras em “tempo real”, melhoramento da simulação de iluminação, *meshes* (formato de objetos 3D), e *avatars* mais realísticos e personalizáveis. Diferentemente do padrão *SL* para objetos virtuais o *mesh* é hierárquico

(suporta múltiplos *meshes* e *sub-meshes*) e pode incluir um *skeleton* para definir o movimento e dinâmica de *avatares*. *Meshes* podem ser desenvolvidas uma grande variedade de ferramentas como: o *Blender* (BLENDER, 2011), que é um *software* livre bastante versátil para criação de gráficos 3D; o *3DS Max* (AUTODESK, 2011) (antigo *3D Studio MAX*), que é um *software* comercial da *Autodesk* bastante utilizado no mercado gráfico; etc.

O *RealXtend Viewer* se conecta ao servidor (implementação) *OpenSim* através de uma conexão separada chamada de *ModRex* (módulo do *RealXtend*). No início de 2009, foi dado início, ao desenvolvimento de um novo *viewer* chamado de *Naali* sob licença BSD para implementar uma arquitetura de visualização mais robusta (vide Fig. 28).



Figura 28: *RealXtend* - interface com gerador de *avatar*, importação de *meshes* e o *Naali viewer*.

Fonte: Website do *RealXtend*.

Por sua vez, em 2009, também foi lançado um novo conjunto-servidor (*server suite*) para o *RealXtend*, chamado de *Taiga*, que incorpora ideias e metaobjetos do projeto *ScienceSim* (SCIENCESIM, 2011) (apoiado em parte pela *Intel* e derivado também do *OpenSim*), que inclui utilidades do *ModRex* e *ModCableBeach* (outro módulo criado pelo projeto *RealXtend*) que proporciona acesso de inventário via *R.O.B.U.S.T.* (*Redesigned OpenSim Basic Universal Server Technology*), serviços de *grid*, inventário e *avatares*

gerenciáveis por ferramentas *Web* (*WebDAV* (WEBDAV, 2011) - *Web-based Distributed Authoring and Versioning*) e autenticação aberta (*OpenID* (OPENID, 2011)).

Pesquisadores da Universidade de Graz, na Áustria, utilizaram o *RealXtend* para a criação de salas de conferências virtuais para fins educacionais, embora tenham tido alguns problemas de instabilidade, o código pôde ser alterado para transmissão de vídeo online (KAPPE; GUETL, 2009).

3.5.6 3DXplorer

O *3DXplorer* (3DXPLORER, 2011) é simultaneamente uma ferramenta de desenvolvimento de mundos virtuais e um conjunto de mundos virtuais propriamente ditos. É um produto comercial criado pela empresa *Altadyn* em 2004 e destaca-se dos ambientes imersivos anteriormente apresentados por dispensar aplicação cliente (*viewer*) para o acesso, sendo que para isso basta a utilização de um *Web Browser* (navegador comum 2D) (vide Fig. 29). No entanto, faz-se uso de *Applets Java* que, por sua vez, necessitam da instalação da *JVM*.

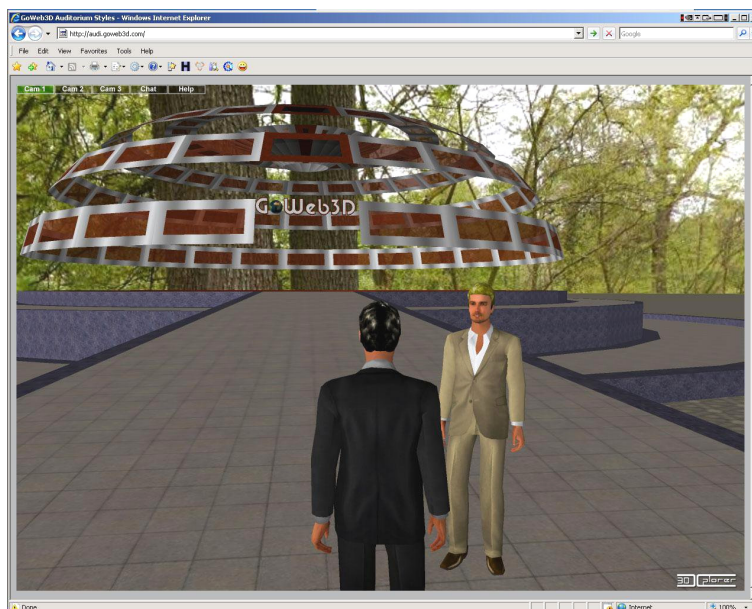


Figura 29: 3DXplorer - interface.

Fonte: Website do 3DXplorer.

A abordagem de exploração dos mundos é também diferente das outras implementações apresentadas. Esses mundos virtuais constituem-se como produtos *3DXplorer*, e em todos é possível criar diferentes ambientes 3D com a ferramenta *3DX Studio* e personalizar *avatars*. Os mundos virtuais são basicamente para uso em ambientes empresariais e podem estar hospedados tanto nos servidores da *Altadyn* como em servidores das próprias empresas-clientes. Existem diferentes modalidades de produtos variando de acordo com o propósito, número máximo de usuários simultâneos, tráfego, forma de apresentação e interação entre usuários.

Num nível mais básico, o *3DXplorer* disponibiliza um produto chamado *3DXplorer Platform* completamente gratuito que permite a configuração do espaço em redor e a configuração de *avatar* como o *3DXStudio*. Os utilizadores para terem acesso aos mundos virtuais *3DXplorer* precisam de se autenticar com o seu nome de *avatar* e palavra-chave, em alguns mundos é possível entrar como convidado apenas com o nome de *avatar* mas

neste caso não é possível configurar o mesmo.

Não há utilização desta ferramenta para educação já que os custos afastam esta possibilidade para a maioria das instituições.

3.5.7 Outros

Existem outros ambientes imersivos similares aos anteriormente apresentados como o *Teleplace* (TELEPLACE, 2011), comercial; o *ScienceSim*, que usa o *OpenSim* em conjunto com repositórios próprios; o *QSim* (QSIMS, 2010), comercial mas baseado no *OpenSim* com algum suporte empresarial; e o *Vastpark* (VASTPARK, 2011), FOSS, que usa uma linguagem inovadora pra descrever conteúdos interativos, o *IMML* (*Interactive Media Markup Language*) e possui uma arquitetura aberta que proporciona conteúdos da *Web* imersivos e adaptativos (*Immersive Adaptive Web*).

3.6 Serious Games

Ainda são poucas as implementações de ambientes educacionais que tenham características lúdicas ou de *serious games* (vide teoria na Seção 2.11). O *JDoc* (SLINEY; MURPHY, 2008) é uma destas implementações, um simulador 3D para instrução de médicos no tratamento de pacientes que proporciona maior interesse por parte dos estudantes. Os estudantes, além de poder formar laços sociais como no *SL*, também aprendem divertindo-se e tornam as tarefas que empregam computadores mais confortáveis. A Fig. 30 ilustra a interface do jogo e uma captura do momento de diagnósticação de um paciente.



Figura 30: *JDoc* - interface do jogo.

Fonte: (SLINEY; MURPHY, 2008).

Outros, como o trabalho de HENDAOU; LIMAYEM; THOMPSON (2008) apontam desafios e pesquisas da associação de ambientes 3D com interfaces lúdicas e também sociais, os chamados mundos 3D sociais (*3D Social Virtual Worlds - 3SVWs*). Há também diversos estudos de implementação de jogos computacionais tridimensionais empregados na educação de engenheiros, como o trabalho de MAYO (2007), que aponta o potencial de jogos (e *video games*) como meio massivamente efetivo de complemento à educação (se usado paralelamente com o ensino tradicional) por ao menos cinco razões:

- i. alcance - *video games* possuem apelo de uma grande massa de pessoas;
- ii. flexibilidade temporal - usuários jogam a qualquer momento;

- iii. envolvente - jogos cativam a atenção e podem ser projetados seguindo paradigmas de ensino efetivo;
- iv. química - jogos estimulam mudanças cerebrais que promovem o aprendizado;
- v. efetividade - alguns estudos apontam para melhorias no aprendizado de 30% com o emprego de jogos.

3.7 Tecnologias de Colaboração

Este tópico de pesquisa é associado às ferramentas que auxiliam na interação para a colaboração. São sistemas direcionados para colaboração de usuários chamados de sistemas cientes de colaboração (*CASs - Collaboration Aware Systems*). Estes sistemas elevam a *CMC* para níveis comparados ao da comunicação presencial tradicional (face-a-face) (CHASTINE; ZHU; PRESTON, 2006). Exemplos dessa consciência do sistema podem ser a simples identificação de gestos ou referências a certos objetos ou figuras de professores/usuários em AVCs. A ciência da inter-referência (*inter reference awareness*), isto é, da referência entre as interações de AVCs, permite que participantes se refiram a objetos de um ambiente colaborativo e que esta referência possa ser entendida e visualizada por participantes fisicamente dispersos (mas virtualmente “presentes no ambiente”).

O projeto *ECOSPACE* (PRINZ et al., 2006; ECOSPACE, 2011) se baseia no ponto de vista de que em 2012 todos os profissionais na Europa estarão aptos para colaboração dinâmica, criativa e transparente através de grupos, organizações e comunidades usando um ambiente personalizado de trabalho colaborativo. O *ECOSPACE* é financiado pela Comissão Europeia e contribui para este ponto de vista em quatro pontos:

- i. definição de paradigmas de trabalhos inovadores pela análise de profissionais com suporte eletrônico e suas respectivas organizações;
- ii. projeto e desenvolvimento de uma arquitetura orientada a serviços que utilize padrões abertos para sistemas de AVCs;
- iii. uso de um *middleware* e de serviços que permitam colaboração transparente e instantânea entre trabalhadores formados em grupos na rede além dos limites organizacionais;
- iv. Desenvolvimento de novas ferramentas que simplifiquem a complexidade da colaboração em ambientes de trabalho dinâmicos e que permitam usuários desempenharem tarefas intensamente criativas e de conhecimento.

Uma ideia da arquitetura de tecnologias/ferramentas e atividades do *ECOSPACE* está representada na Fig. 31, onde é idealizada uma plataforma centrada no usuário e que permite interoperabilidade de ferramentas e serviços inovadores assim como a evolução dos serviços de colaboração para ambientes com consciência de cooperação.

Também como objetivo do projeto é proposto o desenvolvimento de um ambiente para integração dos serviços existentes que parceiros do projeto já utilizam (QUEMADA et al., 2006). Por este motivo foram desenvolvidos um *middleware* e um modelo de referência (*reference model*) para colaboração dos “ambientes”. No topo deste *middleware*, ferramentas de colaboração inovadoras serão desenvolvidas para aumentar a consciência de cooperação para usuários e ferramentas.

Os componentes básicos deste ambiente cooperativo são formados por serviços existentes como *e-mail*, espaços de trabalhos compartilhados e compartilhamento de aplicações ou de tarefas. Entre os novos serviços, serviços de presença e consciência irão desempenhar um papel crucial. Estes serviços não necessários na cooperação distribuída

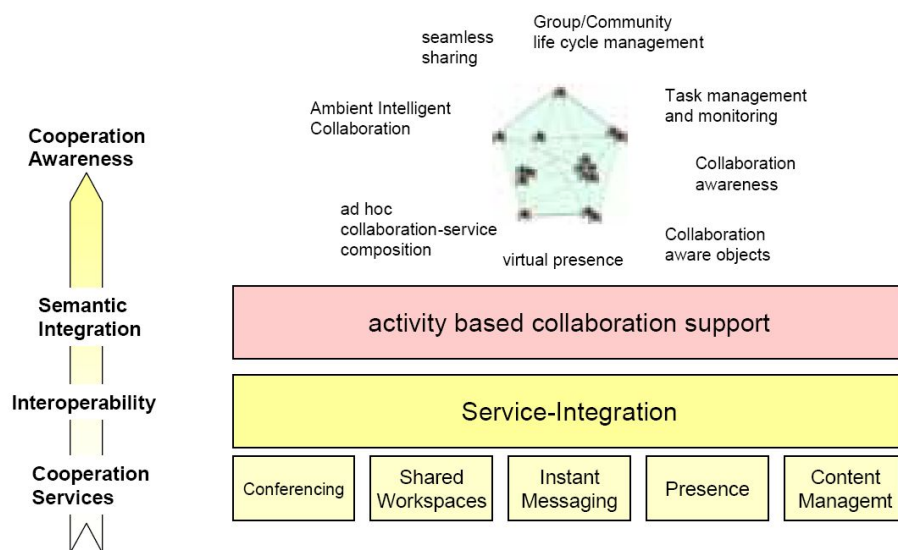


Figura 31: *ECOSPACE* - diagrama organizacional.

Fonte: (PRINZ et al., 2006).

para assistir usuários nas suas compreensões mútuas e na situação e no progresso de trabalhos, assim como no ritmo de trabalho de outras organizações.

De acordo com PRINZ et al. (2006), futuros profissionais do ramo eletrônico ou que usam tecnologias eletrônicas, chamados de *e-Professionals*, que irão usar ambientes colaborativos de trabalho necessitarão cada vez mais de características como: flexibilidade, mobilidade e comunicação *ad hoc*. Este cenário no qual ele estará envolvido requer uma mudança de aplicações orientadas ao projeto de ambientes de trabalho com consciência de colaboração que suportem cooperação e interação em termos de atividades ao contrário de somente cooperação nas funções técnicas. Sistemas avançados que suportem gerenciamento de tarefas distribuídas, espaços de trabalhos compartilhados ou fluxos de trabalho, ou ainda colaboração “tempo-real”, que ainda estão nos estágios iniciais de adaptação comparados com o uso de correio eletrônico resultando num sobre carregamento cognitivo dos usuários.

3.8 Tutoriamento Autônomo

Outra importante linha de pesquisa para AVCs que focam o aprendizado é o tutoriamento autônomo, isto é, a capacidade do sistema desenvolvido guiar e oferecer *feedback* aos usuários durante o processo de aprendizado⁵. Isto pode ser visto como uma tarefa de automação na qual se deseja, com base em interações e dados estudantis, interagir com o usuário oferecendo alternativas de ensino-aprendizagem. Também é visto como ramo dos *ITSs*, onde a inteligência pode ser associada com a automatização do processo.

Existem muitos trabalhos que para o tutoriamento autônomo empregam SMA em AVAs. O trabalho de OTSUKA; ROCHA; BEDER (2007) cita a importância desta integração para automatizar tarefas de AVAs e também proporcionar algum tipo de *feedback* automático aos usuários do ambiente.

Implementações que usam o *framework* de desenvolvimento de sistemas multiagentes *JADE* (JADE, 2011) com o *MOODLE* para assistir professores na monitoração e no

⁵há alguns pesquisadores que relacionam este tema ao termo *embedded assessment*.

acompanhamento dos estudantes ainda não alcançaram todo o potencial desta associação (SCUTELNICU et al., 2007). A Fig. 32 ilustra o diagrama e as interações entre os agentes, AVA e usuários em um cenário com agentes “rastreado” informações de usuários de *logs*, bancos de dados, etc. onde estas informações são registradas pelo AVA. O SMA, por sua, vez pode propiciar *feedback* ao estudante através da adaptação de conteúdo do AVA.

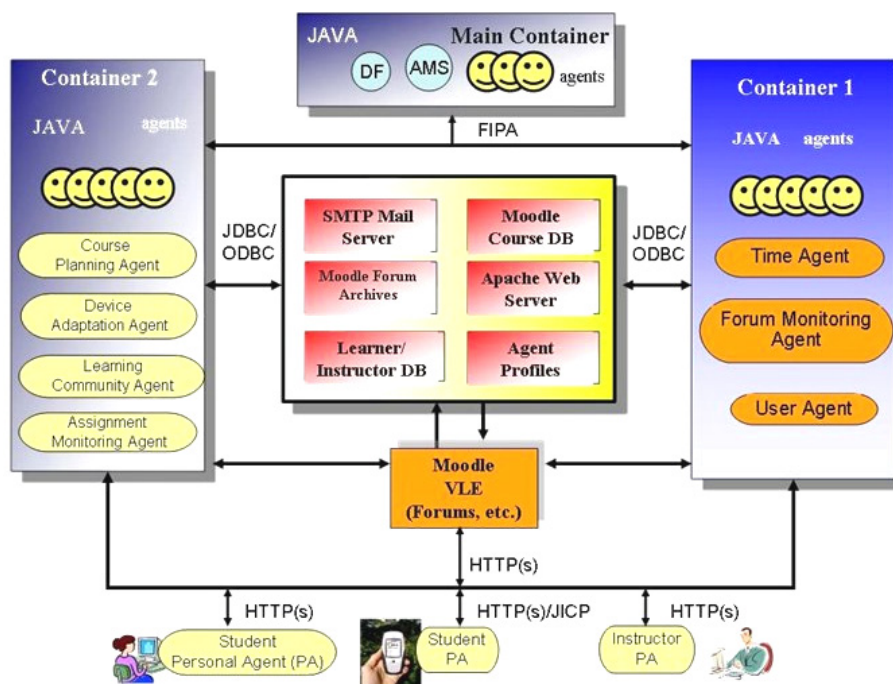


Figura 32: *ITS* simples - emprego do *Framework* para SMA *JADE* em associação com o AVA *MOODLE*.

Fonte: (SCUTELNICU et al., 2007).

3.9 Tecnologias de Rede e Distribuição de Recursos

Existem diversas ferramentas e tecnologias que podem ser empregadas em rede e distribuição de recursos computacionais ou ligados a computadores. O *Access Grid* (AG) (CHILDERS et al., 2000; AG, 2011) possui um conjunto de ferramentas que permite a colaboração multimídia via Internet. Esta implementação gerou bastante interesse de instituições de ensino no início de 2000 com diversas instituições colaboradoras formando *AG Nodes*. O AG usa *IP multicasting* e *RTP* para transmissão de áudio e vídeo entre os “nodos”. Uma ilustração de um dos nodos com a interface e um mapa dos nodos espalhados pelo mundo está presente na Fig. 33.

Outro trabalho no mesmo sentido, mas focado praticamente no compartilhamento de recursos computacionais, é o *CSGrid* (LIMA et al., 2005; CSGRID, 2011), um projeto brasileiro do grupo Tecgraf da PUC-RJ, que usa um *framework* de *grid* baseado em um projeto anterior chamado de *CSBase*, no qual adicionalmente ao suporte para uso e administração de recursos computacionais distribuídos, oferece instalações para integrar aplicações e gerenciamento de dados e usuários. Assim, o *CSGrid* dispõe aos seus usuários através de seus navegadores da *Web* um espaço de trabalho com todas as aplicações disponíveis e com arquivos de dados de usuários organizados por projeto. Usuários podem estender o sistema com aplicações adicionais e o sistema também proporciona carac-



Figura 33: *Access Grid* - nodo e distribuição dos nodos.
Fontes: (CHILDERS et al., 2000) e Website do *Access Grid*.

terísticas de trabalho colaborativo. A interface é construída utilizando a linguagem de programação *Java* que permite independência de plataforma. A Fig. 34 ilustra a interface do espaço de trabalho disponível ao usuário com diversos recursos.

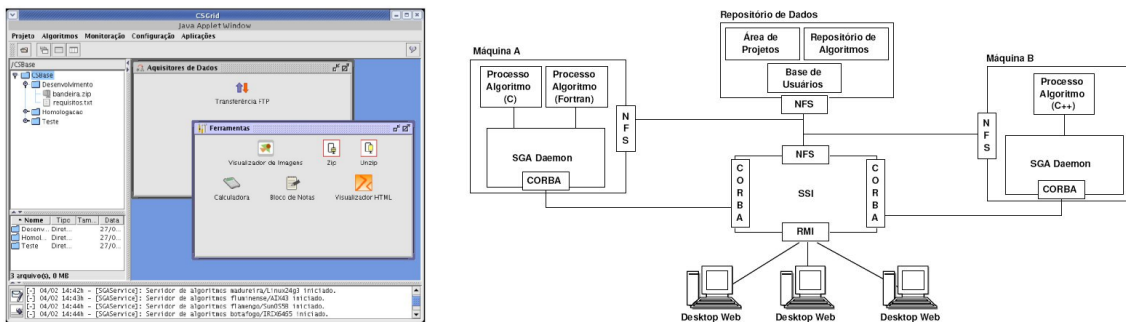


Figura 34: *CSGrid* - interface do espaço de trabalho e arquitetura.
Fonte: (LIMA et al., 2005).

3.10 Integração e Interoperabilidade de Ferramentas de Software

A pesquisa de interoperabilidade de conteúdos colaborativos é crucial para o gerenciamento e integração de materiais (mídias eletrônicas e recursos educacionais) hospedados em centenas de AVCs e em AVAs acessíveis na Internet. Como os conteúdos são desenvolvidos por muitos educadores/usuários, não há garantia que todos sigam o mesmo padrão nem que os conteúdos sejam compatíveis ou confiáveis para visualização em AVCs e AVAs diferentes dos quais eles foram desenvolvidos. Consequentemente ações de padronização tanto para mídias colaborativas quanto para comunicação entre as mídias e os AVCs ou AVAs são de grande importância. Alguns formatos prevêm esta padronização, como o SCORM (SCORM, 2010). Mesmo assim diferentes implementações de AVC ou AVA dificilmente podem interagir (ser compatível) com outra implementação, a não ser que esta interação seja em nível básico como, por exemplo, a visualização de material descrito em *HTML*.

O projeto *ECOSPACE*, anteriormente mencionado (vide Seção 3.7), possui preocupação especial com a interoperabilidade e adota acima dos serviços oferecidos uma camada aberta de integração que permite a comunicação vertical e horizontal (vide Fig. 31). Isto permite a integração de diferentes espaços de trabalhos de fabricantes distintos; e de serviços complementares (por exemplo: gerenciamento de usuários entre sistemas do espaço de trabalho e *softwares* de mensagens instantâneas). A camada de integração emprega três

diferentes topologias de arquitetura de comunicação: *SOA* (*Simple Object Access*), *P2P* e cliente-servidor.

Do mesmo modo do *ECOSPACE* o consórcio *ATHENA* (ELVESÆTER et al., 2006; *ATHENA*, 2011) tem por objetivo dar mais interoperabilidade entre ferramentas e organizações existentes, suprimindo uma semântica para interfacear diferentes aplicações. *ATHENA* é um acrônimo para *Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Applications*. O projeto do consórcio propõe um centro de interoperabilidade aberto, neutro e independente de companhias no qual todos os participantes podem colaborar. A interoperabilidade neste projeto visa integrar sistemas TIC com bases de conhecimento e de negócios através de uma semântica simples.

O projeto *MOSAIC* (SCHAFFERS et al., 2006; *MOSAIC*, 2011) propõe um ambiente de suporte a trabalhadores “móveis” (*Mobile Worker Support Environments*). O objetivo principal do projeto é promover espaços de trabalho inovadores e lançar comunidades de trabalho em ambientes inteligentes (*AMI@WORK* (*AMI@WORK*, 2011)). O *MOSAIC* tem caráter diferenciado no tratamento dos cenários móveis para espaços de trabalho, que são estruturados por dois extremos: a atitude humana (individualista ou orientada à comunidade) e o desenvolvimento organizacional (controle hierárquico ou auto-organizacional). Esta implementação pode ser hoje encontrada comercialmente com o nome de *BSCW* (*ORBITEAM*, 2011) (*Basic Support for Cooperative Work*) projetado em linguagem de programação *Python*. Foi originalmente desenvolvido pela *Fraunhofer Society* e atualmente é comercializado pelo *OrbiTeam Software*.

Igualmente, como anteriormente mencionado (vide Seção 3.9), o *CSgrid* possui um servidor integrador do sistema (*System Server Integrator - SSI*) responsável por integrar (assegurar comunicação entre) diferentes máquinas ou usuários no *grid*. A interoperabilidade é garantida usando um *middleware* comum com *Java RMI* ou *CORBA*. *Proxies* permitem o controle de usuários e permissões de usuários para diversas funcionalidades disponíveis no *grid*, como: execução de algoritmos, transferência de dados, armazenamento de dados, etc. (vide Fig. 34). O *software* (agente) de gerenciamento de algoritmos - SGA administra a distribuição dos algoritmos. A interação entre o SGA e o *SSI* está implementada nas linguagens de programação *Lua* (*LUA*, 2011) e *C++*, através da interface (*middleware*) *CORBA*.

Outro projeto que visa integração e interoperabilidade entre ferramentas e recursos é o *Gaia* (ROMÁN et al., 2001), resultado de seis anos de pesquisas em: *middleware* “reflexivo” (*reflective middleware*) (ROMÁN; KON; CAMPBELL, 2001), sistemas metaoperacionais (*meta-operating systems*), *middleware* para dispositivos portáteis e embarcados e computação ubíqua (*ubiquitous computing*). Anterior ao *Gaia* os membros do projeto desenvolveram o sistema metaoperacional *2K* (KON et al., 2000), um sistema operacional de *middleware* reflexivo desenvolvido sobre sistemas operacionais tradicionais como: *Windows*, *Linux*, *Solaris* e *PalmOS*. O *2K* foi fortemente influenciado por pesquisas anteriores de *middleware* reflexivo e foi construído sob uma versão modificada do *TAO* (KON et al., 2000), o *ORB CORBA* baseado em padrões propostos por SCHMIDT; LEVINE; MUNGEE (1998). O *2K* oculta os dispositivos e a heterogeneidade de sistemas operacionais, podendo-se adaptar dinamicamente para mudanças no ambiente enquanto mantém a integridade geral do sistema. Usuários no *2K* interagem com o sistema usando dispositivos diferentes, eliminando, portanto, o mapeamento de um usuário para um dispositivo. Assim, dispositivos individuais no *2K* se tornam portais do sistema.

Seguindo pesquisas do *2K*, uma nova linha de pesquisa foi iniciada para integrar dispositivos móveis com limitações de recursos nos ambientes de computação distribuída.

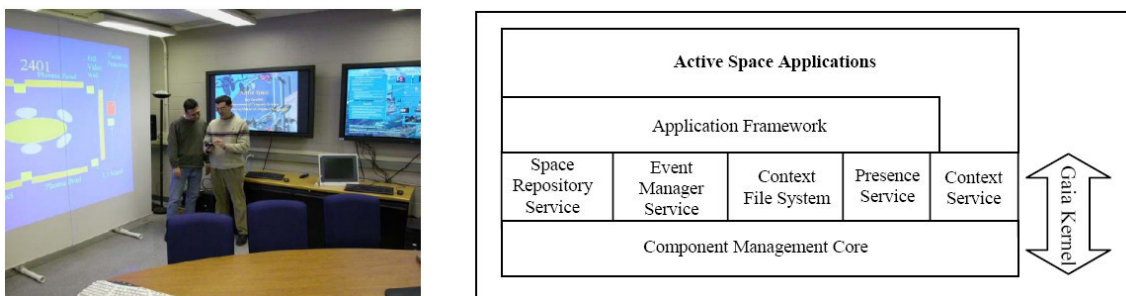


Figura 35: *Gaia* - espaço ativo experimental e arquitetura.

Fonte: (ROMÁN et al., 2001).

Estes dispositivos “usam” o *middleware* para interoperar com o *2K* e disponibilizar suas funções nos serviços do *2K*. Uma parte dessa pesquisa desenvolveu um protótipo de *middleware* personalizado para dispositivos portáteis, o que envolveu uma infraestrutura totalmente reconfigurável de *middleware*. O *middleware* permite interação bidirecional entre dispositivos e o sistema metaoperacional. Como resultado o sistema operacional *Gaia* foi criado o *GaiaOS*, influenciado por pesquisas do *Georgia Tech*, *MIT* e *Berkeley* em computação ubíqua. O *GaiaOS* é um sistema metaoperacional personalizado para espaços físicos (vide Fig. 35) que suporta o desenvolvimento de aplicações personalizadas para este ambiente. O *GaiaOS* disponibiliza uma *API* que abstrai a complexidade e a heterogeneidade associada aos ambientes ubíquos de computação. A explícita conexão entre o *GaiaOS* e o espaço físico requer novos serviços (não presentes no *2K*) para gerenciar questões, como contexto e detecção de recursos adicionados e removidos do espaço físico.

O *Gaia* disponibiliza suporte para aplicações móveis centradas no usuário e gerencia recursos e serviços de um espaço ativo (*active space*). Assim, proporciona serviços de localização, contexto, eventos, e informações sobre o espaço ativo. O *GaiaOS* é construído como um sistema distribuído de objetos e seus blocos mais importantes são: o *kernel*, o *framework* de aplicações, e as aplicações em si. Os cinco serviços básicos do *Gaia* são: gerenciador de eventos (*event manager service*); de presença (*presence service*); de contexto (*context service*); espaço de repositório (*repository space service*); e sistema de arquivos de contexto (*context file system*).

O *Gaia* usa aplicações escritas em linguagem de *script* de alto nível chamada *LuaORB*, para programar e configurar espaços ativos e coordenar as atividades que eles contêm. O *LuaORB* é baseado na linguagem interpretada *Lua*, que simplifica o gerenciamento e a configuração de tarefas e permite a prototipação rápida e testes. O interpretador de *Lua* é rápido e tem um *footprint* (consumo) de memória baixa, o que o torna apropriado para dispositivos com limites de recursos. O *LuaORB* implementa conexões de linguagem (*language bindings*) entre o *Lua*, *CORBA*, *COM* e *Java*. Isso permite a fácil interação com os componentes do sistema.

3.11 Análise e Comparações

Visto que a evolução das “tecnologias” dita o rumo das implementações de AVCs e outros afins, só pode-se analisar as diferentes implementações e estudos apresentados de forma cronológica e apontar algumas vantagens dos diversos trabalhos analisados e descritos e comparações entre AVCs de mesmo foco.

A Fig. 36 ilustra a cronologia e assinala a evolução das técnicas, conceitos e im-

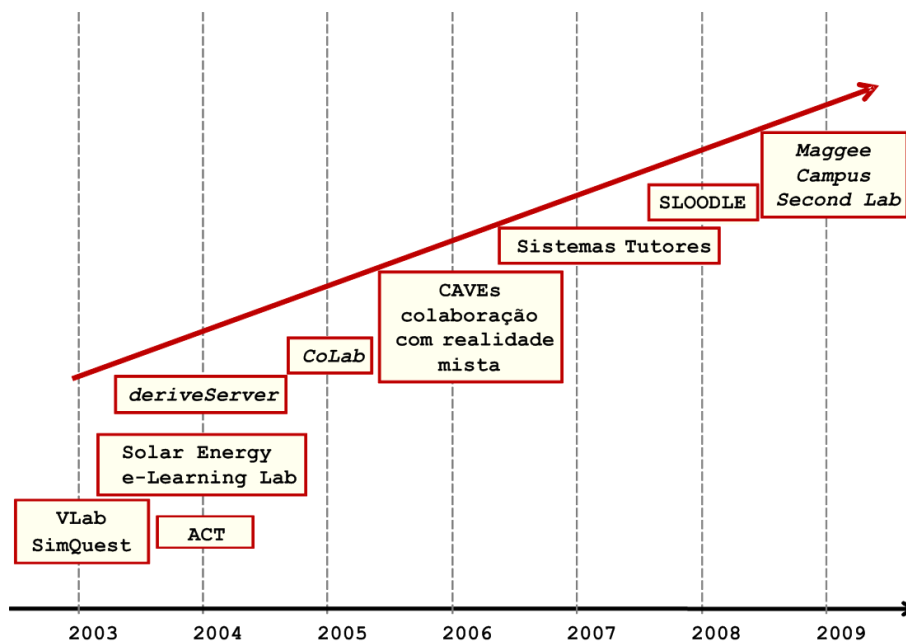


Figura 36: Cronologia de trabalhos relacionados a AVCs.

plementações empregados nos trabalhos ligados a AVCs de maior repercussão para esta proposta⁶. Nota-se que há uma clara tendência a incorporar cada vez mais funcionalidades que venham a incrementar ou enriquecer os ambientes com fins educacionais. Inicialmente, (vide Seção 3.2 para uma cronologia anterior), laboratórios remotos com simulações eram desenvolvidos somente para aulas práticas e muitos sem vínculo com materiais de ensino ou AVAs (como o *VLab* ou *SimQuest*). Em um próximo passo laboratórios incorporaram experimentos remotos reais complexos (*ACT*) e foram integrados a sistemas de ensino, como AVAs (*HTI Solar e-Learning Lab*). Também passaram a usar realidade mista e visualizações mais “reais” (*deriveServer*). A colaboração no experimento ganhou grandes adeptos a partir da popularização de *CoLabs*. O uso de outras interfaces de usuários como *CAVEs* também foi testada num próximo passo de colaboração. Sistemas de tutoriamento autônomo deram a sistemas “passivos” uma interatividade e *feedback* aos estudantes com simples *ITSS*. Uma nova interface social e imersiva foi testada em conjunção com AVAs no projeto *SLOODLE*. E, por consequência, o uso de AVCs com interface social foi empregado em conjunto com a experimentação.

No estado da arte, duas grandes categorias de AVCs serão separadas em duas tabelas com características e comparações diferentes. A primeira grande área de comparação contém implementações que seguem em sua maior parte conectividade sem a preocupação maior com o caráter educacional. A Tabela 2 compara funcionalidades e características, das implementações apresentadas no estado da arte que não possuem vínculo explícito com a educação, como suporte a vídeo e reunião virtual, suporte a *IM*, suporte a espaços de trabalho compartilhados, interface do usuário (cliente), suporte a interoperabilidade, topologia de conexão entre cliente e servidor, e a linguagem de programação usada para o desenvolvimento do *software*.

Note que nenhuma das implementações da tabela de comparação têm uma interface de usuário *thin-client*, isto é, todas necessitam de um *software* adicional. Existem duas

⁶o período analisado naturalmente engloba também os anos recentes de 2010 e 2011, mas não notou-se uma representatividade maior nestes anos a ponto de mudar a caracterização deste nicho de pesquisa.

Tabela 2: Comparação dos AVCs apresentados sem pretensões educacionais explícitas.

implement. vs. características e funcionalidades	<i>Gaia</i>	<i>ECOSPACE</i>	<i>AccessGrid</i>	<i>CSGrid</i>
vídeo/reunião	sim	sim	sim	sim
<i>IM</i>	próprio	próprio	<i>jabber</i>	próprio
espaço de trab. compart.	<i>active spaces</i>	pers.	pers.	<i>grid computing</i>
interface	<i>thick</i>	<i>thick</i>	<i>thick</i>	<i>java (thick)</i>
interoperab.	<i>CORBA(TAO)</i> e <i>LuaOrb</i>	<i>SOA</i>	pers.	<i>CORBA</i>
topologia de conexão	cliente-servidor	<i>P2P</i>	cliente-servidor	cliente-servidor
ling. de prog.	<i>Lua</i>	não espec.	<i>python</i>	<i>Lua</i> e <i>C++</i>

implementações não abordadas que possuem interface *Web (thin-client)*. O *Adobe Connect* (ADOBE, 2011) é um *software* comercial da *Adobe* para *Web* conferências baseado em tecnologia *Flash* com funcionalidades, como áudio/vídeo conferência, sincronizadas de apresentações remotas, compartilhamento para visualização de janelas ou tela do cliente, *chat* e *whiteboards*. E o *OpenMeetings* (OPENMEETINGS, 2011), *FOSS* embora também baseado em *Flash* (que não é livre), que possui praticamente as mesmas funcionalidades do *Adobe Connect* para apresentações/reuniões remotas, mas com algumas restrições de tecnologias comerciais (por exemplo, não converte apresentações do *Microsoft PowerPoint* para *Flash* e nem possibilita compartilhamento e visualização de telas/janelas do cliente, isto é, do apresentador). Semelhantemente o *software BigBlueButton* (BBB, 2011) também oferece as mesmas funcionalidades do *OpenMeetings* pois é baseado no mesmo.

Todos os AVCs da Tabela 2 possuem tecnologias usuais para interoperabilidade assim como funcionalidades de mensagens instantâneas (usualmente *chats* síncronos) e espaço de trabalho compartilhado. O *Gaia* possui um espaço de trabalho compartilhado com *active spaces* empregando computação ubíqua e o *CSGrid* usa técnicas de computação de *grid* para gerenciamento de recursos de uma subrede de computadores. O *ECOSPACE* destoa um pouco no quesito de arquitetura de comunicação, pois é o único que emprega *P2P* com *SOA*, os outros usam topologia cliente-servidor.

A Tabela 3 apresenta comparações da segunda categoria de AVCs descritos no Capítulo estado da arte. Esta categoria pode ser chamada de AVCs com pretensões educacionais explícitas. As implementações serão comparadas de acordo com os suportes e gerenciamento de material didático, suporte a mensagens instantâneas (*IM*)/*chats*, suporte a experimentação remota; interface *thin-client*; capacidade de interoperação, arquitetura de comunicação e qual a linguagem de programação empregada.

Notamos que a característica que difere nas implementações de AVCs é o suporte à experimentação. A maioria dos AVCs normalmente suporta simulações. Já o *Mixed Reality Mechatronics Lab* suporta simulações que são construídas em seu ambiente, ou seja, na bancada virtual, e qualquer outro equipamento real ou simulado que esteja conectado à bancada por meio de *hyper-bonds*, isto é, uma gama muito maior de experimentos, tanto reais como simulados (realidade mista), são suportados. O *SLOODLE*, por sua vez, suporta simulações que podem ser incorporadas no *MOODLE* ou que foram desenvol-

Tabela 3: Comparação dos AVCs apresentados com pretensões educacionais explícitas.
“-” não suporta e “X” suporta.

x	mat. did.	exp.	interface	esp. trab. compart.	interop.	top. de com.	ling. de prog.
<i>VCLab</i>	-	sim, <i>MatLab</i>	<i>thick</i>	-	-	cliente-servidor	<i>Java</i> , <i>HTML</i> , <i>VRML</i>
<i>SimQuest</i>	X	sim.	<i>thick</i>	-	em parte <i>Java</i> <i>RMI</i>	cliente-servidor	<i>Java</i> e <i>HTML</i>
<i>ACT</i>	-	real e sim.	<i>thin</i>	-	-	cliente-servidor	<i>MatLab</i> , <i>Java</i> e <i>HTML</i>
<i>CoLab</i>	-	real e sim.	<i>thick</i>	X	-	cliente-servidor	<i>Java</i>
<i>MS</i>	-	sim.	<i>thick</i>	-	-	<i>P2P</i>	<i>Java</i> e <i>XML</i>
<i>HTI SEL</i>	X	real	<i>thin</i>	no AVA	-	cliente-servidor	não espec.
<i>DeriveServer</i>	X	misto	<i>thick</i>	X	em parte	cliente-servidor	<i>Java</i> <i>Scripts</i> , <i>PHP</i> , <i>C++</i> e <i>VRML</i>
<i>AWEDU</i>	X	-	<i>thick</i>	em parte	em parte	cliente-servidor	não espec.
<i>SLOODLE</i>	X	-	<i>thick</i>	em parte	em parte	cliente-servidor	<i>C++</i> , <i>PHP</i> e <i>Linden</i> <i>Scripts</i>
<i>3D Physics Lab</i>	X	misto	<i>thick</i>	em parte	X	cliente-servidor	<i>Java</i>
<i>Second Lab</i>	X	real	<i>thick</i>	em parte	em parte	cliente-servidor	<i>C++</i> , <i>PHP</i> , <i>AJAX</i> e <i>Linden</i> <i>Scripts</i>
<i>Magee Campus</i>	X	misto	<i>thick</i>	em parte	em parte	cliente-servidor	<i>C++</i> , <i>PHP</i> e <i>Linden</i> <i>Scripts</i>

vidas no *metaverso* usando a linguagem de programação *Linden Scripts* do *Second Life*. Também se nota a clara escalada evolutiva entre uma implementação e outra da tabela.

O AVC *MS* é o único dos estudados que emprega *P2P* como topologia de comunicação, todos os outros usam cliente-servidor. Algumas usam interfaces 3D sociais, isto é, *metaversos*, como o *AWEDU*, o *SLOODLE*, o *3D Physics Lab*, o *Second Lab* e o *Magee Campus*. O *Mixed Reality Mechatronics Lab (DeriveServer)* e o *SLOODLE* usam uma

implementação livre de AVA, o *MOODLE*, e, por consequência, essas duas implementações permitem a maior flexibilidade na construção dos materiais educacionais e também na incorporação de maiores funcionalidades ao AVA.

4 PROPOSTA DE ARQUITETURA

4.1 Introdução

A comunidade científica anseia por novas metodologias de ensino-aprendizagem que utilizem ferramentas inovadoras capazes de incentivar e motivar a aprendizagem e auxiliar o ensino e/ou o treinamento. Esforços de instituições de ensino no mundo inteiro foram apresentados no capítulo anterior (vide Capítulo 3) neste sentido e algumas inclusive constituem um grande intercâmbio de sistemas e elementos de ensino-aprendizagem (metodologias, conceitos e ferramentas).

Novas metodologias, ainda mais no ensino-aprendizagem, sempre enfrentam alguma resistência por parte dos educadores. Por se tratar de uma área normalmente conservadora, a evolução é muito lenta, embora sejam notórias as diversas dificuldades na educação tecnológica. Entre os desafios constantes que causam dificuldades estão: a necessidade do estreitamento entre teoria e prática e a motivação dos estudantes em contraste com a acelerada evolução tecnológica. Ainda que algumas destas dificuldades possam ser atenuadas pelo uso de tecnologia na educação, este ainda é um tema onde há muita controvérsia entre os educadores. A utilização de suporte computacional como ferramenta na instrução educacional torna-se com o passar do tempo absolutamente imprescindível em áreas técnicas.

Foco constante das tecnologias de ensino-aprendizagem quando é necessária a concentração de esforços, os ambientes virtuais são peças-chave da evolução do ensino-aprendizagem. A colaboração virtual, quando aplicada ao ensino-aprendizagem, aproxima estudantes e professores distribuídos por diversas localidades sem a necessidade de deslocamento aos centros instrucionais, o que educadores chamam de ensino aberto. AVCEAs, AVCs de Ensino-Aprendizagem, são ferramentas que auxiliam a aprendizagem colaborativa focando ações no refinamento e na integração de processos para auxiliar os seus usuários a colaborarem de forma a aprender em conjunto. Os esforços e resultados apresentados reforçam a teoria de que o aspecto motivacional e inovador dos AVCs influenciam o ensino positivamente. Desta forma, o ensino colaborativo permite que estudantes aprendam contextos educacionais: relativamente realistas, cognitivamente motivacionais e socialmente enriquecidos; comparados a outros paradigmas tutoriais como: ensino socrático, ensino investigativo e ensino integrado (KUMAR, 1996).

AVCs além de possibilitarem a colaboração e a cooperação quando empregados na instrução de usuários, que o autor chamará no restante do trabalho de AVCEAs, possuem características que são benéficas para toda a metodologia. Segundo KREIJNS; KIRSCHNER; JOCHEMS (2003), estas características se fundem com as semelhanças de colaboração e cooperação. Em ambas há características como:

- aprendizado ativo (*active learning*) (WATSON, 1995; SMITH, 1989);
- o professor ou tutor torna-se mais um facilitador do que um centralizador/ transmissor do conhecimento;
- o ensino e a aprendizagem são “experiências” compartilhadas;
- estudantes/estudantes participam em atividades de grupos pequenos;
- estudantes/estudantes tornam-se responsáveis pelo aprendizado;
- estudantes/estudantes são estimulados a refletir sobre suas próprias convicções e pensamentos;
- habilidades sociais e de grupo são desenvolvidas com consenso através de trocas “dar e ganhar” (*give-and-take*).

Embora existam inúmeras implementações de AVCs, não existe nenhuma equivalente em termos educacionais ao ensino tradicional. Há, também, inúmeros estudos experimentais que enfatizam a eficácia da colaboração no aprendizado. Experimentos de interação construtiva conduzidos por MIYAKE (1986) confirmam que no processo de aprendizado as críticas construtivas ocorrem em sua maioria durante a colaboração de estudantes. Refletindo-se sobre este estudo pode-se rapidamente avaliar a colaboração nos métodos tradicionais de ensino. Geralmente a colaboração neste caso limita-se a tarefas de grupo em sua maioria extrainstituição. Por este motivo empregar ferramentas como AVCs como complemento de ensino-aprendizagem extrainstituição não apresenta nenhuma mudança drástica e sim apenas uma união de interesses. Professores e estudantes se beneficiariam em um mesmo “ambiente” em prol da aprendizagem e aperfeiçoamento mútuo. Além disso, o enriquecimento por flexibilidade temporal e espacial, adicionados ao emprego de tecnologia (mídias interativas) motiva, a evolução dos conjuntos aprendizado-estudantes e ensino-professores.

O aumento de usuários que utilizam ou “frequentam” implementações como *MUVEs* ou *MMOGs*, redes sociais, ou ainda *socialwares*, como por exemplo o *SL* ou semelhantes (vide Seção 3.5), aponta para a possibilidade de utilizar tais “ferramentas” e conceitos semelhantes no ensino com conteúdos cada vez mais interativos. Esta interface “lúdica” (*game-like*) capta a atenção dos estudantes enquanto pode focar na colaboração para a aprendizagem.

Parte do sucesso dos “ambientes lúdicos” também é creditada às representações tridimensionais e ao realismo nos gráficos e simulações dos quais os jogos computacionais são dotados. Direcionar esta “tecnologia” para o ensino (vide Seção 3.6) aproxima estudantes de situações “reais” (no sentido da ocasião/situação e da visualização) que são simuladas em ambientes virtuais. Esta característica também se mostra de valia educacional para ser integrada ao ensino-aprendizagem de engenharia, pois situações reais que podem ser simuladas não faltam neste cenário.

Além do aspecto motivacional e colaborativo é importante manter o compasso do ensino tecnológico (engenharias e outras) com a demanda prática crescente no mercado de trabalho. O uso de laboratórios remotos, que são uma alternativa para atender às demandas por mais laboratórios (COOPER, 2000), pode remediar tal demanda. Experimentos remotos, baseados na *Web*, possuem diversos aspectos positivos para a implantação, pois utilizam frequentemente equipamentos industriais de alto custo, que requerem uma área específica para a instalação. Este tipo de experimento é acessível a um maior número de estudantes, pois possui tanto flexibilidade espacial (acessível fora da instituição de ensino), quanto temporal (podem ser acessados 24 horas nos 7 dias na semana). Comprovadamente podem-se adaptar diversos experimentos existentes e transformá-los em

experimentos remotos, atendendo a uma faixa maior de estudantes, e podendo proporcionar maior segurança a equipamentos da instituição utilizados na experiência. Apesar do alto custo de desenvolvimento, os experimentos remotos com plataformas de aprendizagem são amplamente benéficos para a educação. Este tema foi endereçado em um trabalho anterior do autor e será incorporado neste trabalho (SCHAF, 2006).

A aplicação de técnicas de realidade mista aplicadas à ambientes de trabalho colaborativo (CSCW) aumenta e enriquece as possibilidades tanto de experimentação remota quanto de colaboração nos espaços de trabalho. Isto possibilita também a configuração de diversos cenários com componentes tanto reais quanto simulados para a educação. Tal estratégia de criação de cenários foi estudada e proposta anteriormente pelo autor e se chama “componentes intercambiáveis” (SCHAF; PEREIRA, 2009). A aplicação deste conceito em mundos virtuais sociais tridimensionais é ainda muito escassa, deixando esse campo de pesquisa aberto a estas inovações que trazem mais realidade à virtualidade.

Como efetivamente suportar a colaboração ou cooperação em AVCs é uma tarefa desafiadora. Para que a colaboração ou cooperação se torne uma tarefa praticamente intuitiva (ou transparente) entre usuários em AVCs, existem conceitos, chamados de consciência de colaboração (*collaboration awareness*) que auxiliam usuários sem experiência no ambiente a se comunicarem ou interagirem através dos mais diversos canais disponíveis.

Para que a colaboração ou mesmo o aprendizado dos estudantes no ambiente sejam monitorados e assistidos, tutores devem estar sempre verificando interações dos estudantes com o AVC. Esta verificação (ou monitoramento) pode ser automatizada por conceitos de sistemas de tutores inteligentes (ITSs). Existe ainda pouca pesquisa nesta linha de pesquisa, o que significa que esta área possui ainda muitos desafios. Ferramentas que buscam melhorar o conhecimento de estudantes através do auxílio automático aproximam AVCs voltados para o ensino de sistemas automáticos de ensino com tutoriamento autônomo.

A interoperabilidade entre ferramentas e entre AVCs é uma característica importante que deve estar presente em qualquer nova proposta para aumentar sua usabilidade e principalmente sua capacidade de integração. Pesquisas de padronização de *middleware*, assim como de mídias, devem ser conduzidas para que haja compatibilidade entre AVCs ou AVAs e que, por consequência, o conteúdo colaborativo se torne mais reutilizável e “colaborativo” (com a inclusão de mais usuários que podem incrementar o conteúdo).

Estudos anteriores do autor mostraram que mesmo simples AVAs associados a certos experimentos remotos, aumentam tanto a motivação de aprendizado dos estudantes quanto a taxa de aprendizado. O GCAR-EAD, que foi desenvolvido durante pesquisa anterior, está em uso contínuo por disciplinas de sistemas de controle e automação na UFRGS (SCHAF, 2006).

A partir do estudo do estado da arte deste trabalho, foram identificados problemas e possíveis soluções para questões educacionais integradas à tecnologia. Assim, funcionalidades identificadas como chaves para o sucesso de AVC foram organizadas de maneira a constituir uma base para a construção de sistemas educacionais voltados a AVCs com infraestrutura de rede para o treinamento e ensino de engenharia elétrica e áreas afins. Esta base é fundamentada na arquitetura proposta neste trabalho que prevê modularidade entre as funcionalidades e interoperabilidade entre os conceitos e características presentes e vantajosas de AVCs. O foco do trabalho mostrou-se original, pois nenhuma pesquisa recente propõe o estudo de arquiteturas para ambientes deste tipo e nem propostas de ambientes parecidos, somente tecnologias envolvidas em implementações diversas.

4.2 Conceitos e Características Relevantes Associados aos AVCs

Com base no profundo estudo de conceitos, publicações e resultados nas linhas de pesquisa de tecnologia de educação, ambientes virtuais de aprendizagem (AVAs), ensino aberto/híbrido (*blended/hybrid learning*), aprendizado colaborativo assistido por computador (CSCL), trabalho cooperativo assistido por computador (CSCW), experimentação remota (*remote experiments/laboratories*), realidade mista, sistemas multiagentes (SMAs) e sistemas de tutoriamento autônomos (ITSs), identificaram-se características vitais para a proposição de uma arquitetura baseada em AVCEAs. Estas características são:

- ▷ suporte, gerenciamento e organização de materiais didáticos;
- ▷ suporte à diversos mecanismos e meios de comunicação para interação, inclusive social;
- ▷ suporte à colaboração e à cooperação no aprendizado e em tarefas didáticas;
- ▷ suporte à interfaces imersivas com representações tridimensionais (*metaversos*);
- ▷ suporte à características lúdicas (*serious gaming*);
- ▷ suporte à experimentação remota;
- ▷ suporte à realidade mista;
- ▷ suporte à configuração de cenários com componentes intercambiáveis;
- ▷ suporte à *feedback* educacional;
- ▷ suporte à incentivo de colaboração ativa;
- ▷ suporte à adaptação de conteúdo;
- ▷ suporte à diferentes perfis de usuário;
- ▷ suporte à identificação e a controle de usuários;
- ▷ suporte à armazenamento de diferentes interações dos usuários no ambiente;
- ▷ características modulares;
- ▷ características de interoperabilidade;
- ▷ características de adaptabilidade do sistema;
- ▷ emprego de ferramentas livres e possivelmente de código-aberto (*FOSSs*);
- ▷ interface acessível e de interação via *Web*;
- ▷ interface amigável, transparente e de uso intuitivo aos usuários;
- ▷ suporte a avaliação de dados de interações.

Fundamentado em seções anteriores apresentadas, não é difícil justificar as características identificadas e listadas acima. Todo ambiente de ensino-aprendizagem naturalmente suporta a organização e disponibilização de materiais didáticos, mesmo que seja de forma simples organizados em arquivos, *hypermedia*, etc. Também é naturalmente necessário para caracterização educacional que o AVC hospede material didático, pois do contrário não há base teórica para qualquer atividade oferecida no AVC. Sem organização e gerenciamento destes materiais tanto professores quanto estudantes perdem o interesse no ambiente.

Colaboração e cooperação foram defendidas durante todo este trabalho e não podem estar ausentes durante o processo de aprendizagem. Para que isso seja possível é necessário suprir meios e mecanismos de comunicação entre todos os usuários do ambiente. Para que haja colaboração efetiva, os usuários envolvidos necessitam algum laço social, nem que seja um conhecimento prévio via ambiente, a interação social deve ser encorajada e estimulada também através dos meios de comunicação disponíveis.

Embora ainda muito recente, as interfaces de ambientes imersivas são uma característica desejável de AVCs, pois apresentam de uma forma amigável (*friendly*) e intrinsecamente aceitável o ambiente que passa a ter dimensão e representação semelhantes à

realidade. A personificação dos usuários por *avatars* provoca o sentimento de presença virtual (*sense of being there*) muito mais efetiva do que em ambientes virtuais tradicionais (2D com interface comum na Web). A Fig. 37 ilustra a diferença de interfaces: tradicional (ambientes virtuais comuns na Web) *versus* interface de ambiente imersivo. Nela fica evidente a personificação dos usuários presentes nos ambientes imersivos em contraste com uma lista de usuários *online* na interface tradicional. Outra característica marcante é a representação do ambiente. Enquanto em uma são usadas janelas para organizar os diferentes elementos, na outra o ambiente é intuitivo (*easy-to-use*) e os usuários naturalmente identificam os elementos presentes. Obviamente que a “escolha” de interface é uma questão de abordagem do ambiente e a intuitividade aqui lembrada é subjetiva (dependente da opinião pessoal).



Figura 37: Ambientes imersivos - exemplo de comparação com ambiente tradicional.

Visto que grandes problemas na educação de engenheiros são a motivação e a atenção durante as tarefas e que jogos eletrônicos possuem apelo quase total nesta geração de estudantes, um foco em ambientes virtuais que possuam características de *serious games* em tarefas ou em elementos interativos educacionais é de grande valia para remediar tais problemas. A tarefa ou aprendizado, desta maneira, poderia inclusive ser realizada quase que inconscientemente devido ao envolvimento do usuário com o jogo. A fixação do conteúdo relativa ao processo de assimilação seria recompensada com o cumprimento de objetivos no jogo. A competição e/ou cooperação/colaboração entre os usuários obviamente também pode ser incentivada neste cenário educacional.

A prática remota é teoria recorrente neste trabalho e que não poderia faltar no suporte de um AVC ideal para o ensino-aprendizagem de engenheiros de automação e de controle. Embora não seja o ideal de validade educacional, a experimentação remota é considerada a “segunda melhor alternativa de estar lá” - *SBBT* (ATKAN et al., 1996) em contraponto com a melhor alternativa que seria o conjunto da presença física com os equipamentos laboratoriais reais, chamado de *hands-on (engineering)*. Para muitas instituições o custo laboratorial é excessivo e o *SBBT* passa a ser uma alternativa razoável e benéfica para ambos os lados, instituição e estudantes, pois os custos são reduzidos, a disponibilidade e a segurança são elevadas e é inclusive possível aumentar a oferta de experimentos com cooperações com outras instituições que possuam outros experimentos remotos.

O conceito de realidade mista pressupõe a harmoniosa integração de “elementos” virtuais (simulados) e reais. Este conceito é válido para qualquer nível, seja na interface de usuário, na colaboração, no experimento ou ainda no tutoriamento. O correto emprego desta técnica pode inclusive mascarar elementos reais e virtuais que, do ponto de vista do usuário, não é sequer percebida, isto é, a conexão entre eles é transparente (*seamless*).

Imaginando-se o uso de uma interface imersiva, é possível a substituição/sobreposição de gráficos, de vídeo e áudio no mundo real, e permitindo, assim, a criação de espaços compartilhados que combinam vantagens dos ambientes virtuais e colaboração transparente com o ambiente real (MÜLLER et al., 2007). A mescla de informações é usada pelos colaboradores remotos para incrementar a visão do usuário ou pode melhorar a interação produzindo modelos interativos virtuais compartilhados. Desta forma a realidade mista pode produzir um sentimento compartilhado de presença e de realidade (ENLUND, 2001). Portanto, abordagens de realidade mista são ideais para aplicações de trabalho e laboratórios colaborativos de multiusuários (MÜLLER et al., 2007).

Apesar de intimamente ligados, a realidade mista não implica uso de componentes intercambiáveis, que, por sua vez, servem para flexibilizar cenários educacionais. Esta característica foi abordada anteriormente em SCHAF (2006) e será integrada à arquitetura proposta de maneira natural na experimentação remota.

Sempre presente em sistemas de controle e também na automação, o *feedback* neste trabalho representa a resposta do sistema, neste caso o AVC, a interações/ações do usuário. O *feedback* educacional assim seria tarefa de um elemento tutor autônomo. Este inferiria adaptação de conteúdo baseado em dados coletados pelo próprio AVC. O incentivo à colaboração também seria tarefa de um tutor (visível ou não) que auxiliaria a colaboração entre usuários, preferencialmente entre estudantes. A adaptação de conteúdo torna-se uma grande necessidade para a variação de cenários educativos conforme perfil de aprendizado ou histórico de cada usuário.

Características de qualquer ambiente virtual que tenha níveis de usuários diferentes o controle e suporte a perfis de usuários é extremamente comum em todos os ramos e de grande aceitação, uma vez que ações individuais são identificadas e relacionadas aos usuários.

Pela engenharia ou pela ciência da computação sabe-se que projetar sistemas computacionais usando módulos, além de dar flexibilidade ao projeto, também possibilita distribuição de processamento e/ou recursos. Para se usarem módulos há uma necessidade de criar meios de comunicação entre eles abrindo assim a oportunidade de integração com sistemas ou módulos de outros sistemas. Esta integração por sua vez se dá através da interoperabilidade de tecnologias, ou ferramentas, ou sistemas operacionais, que convergem para um mesmo padrão de comunicação no qual as informações são trocadas. Estas características intrinsecamente também tornam o sistema de configuração adaptável para diversas situações distintas de rede e recursos acessíveis.

Outro notório ponto de convergência no projeto de sistemas é o uso de ferramentas livres, independentes de licenças e de configurações específicas para fundamentação de AVCs. FOSSs são indicados para fins acadêmicos por unirem alta reusabilidade e adaptabilidade de programas. Há também a questão de custo que é inferior a um *software* com licença comercial. Também é importante usar linguagens de programação que não restrinjam sistemas operacionais e que possivelmente possibilitem a utilização em qualquer dispositivo.

Naturalmente, por se tratar de AVCs, a acessibilidade através da *Web* é de suma importância, além de democratizar o uso para todos os usuários.

4.3 Tecnologias e Ferramentas

Atualmente conta-se com uma grande gama de ferramentas de *software* que auxiliam o projeto de AVCs. Estas auxiliam na idealização de uma arquitetura de referência

baseada em módulos com funcionalidades distintas. Da seção anterior, foram definidas características desejáveis para AVCs. Estas características podem ser reunidas em grupos, que mais tarde podem ser organizados em módulos. Assim, segundo a Fig. 38, podem-se identificar seis grandes grupos que reúnem funcionalidades que agem em conjunto.



Figura 38: Grupos de características identificadas para AVCs.

O primeiro grande grupo composto pelos AVAs enfoca o propósito da arquitetura que é educacional. Este naturalmente engloba funções clássicas de AVAs (vide Seção 2.7.2 e Fig. 6) como o suporte, gerenciamento e organização de materiais didáticos; suporte a diversos mecanismos e meios de comunicação para interação, inclusive social; suporte à colaboração e à cooperação no aprendizado e em tarefas didáticas; interface acessível e de interação via *Web*; suporte a diferentes perfis de usuário; suporte à identificação e ao controle de usuários; suporte ao armazenamento de diferentes interações dos usuários no ambiente.

Outro grupo engloba as representações tridimensionais sociais, ou seja, *metaversos* com cunho social, que podem ser empregados como interface estendendo o AVC e adicionando características relacionadas, como imersividade, suporte a características lúdicas (*serious gaming*); realidade mista no sentido estudante material de ensino/aulas virtuais; interface amigável, transparente e de uso intuitivo aos usuários; suporte a diferentes perfis de usuário; suporte à identificação e controle de usuários; suporte à colaboração ativa; suporte ao armazenamento de diferentes interações dos usuários no ambiente.

Outro grande grupo, onde é atribuída a "inteligência" ou autonomia para adaptabilidade e *feedback* de aprendizagem envolve sistemas tutores, suporte à colaboração e adaptação de conteúdo. Analisando-se os paradigmas computacionais presentes na comunidade científica e com base em alguns trabalhos do estado da arte, o uso de SMA para coleta, monitoração e inferência, a partir de técnicas como *data mining*, é apropriado. Este assunto será retomado em uma seção subsequente.

O próximo grande grupo engloba características de prática e flexibilização de cenários práticos com experimentos remotos (experimentação), componentes intercambiáveis e espaços de trabalhos/treinamento compartilhados. Ou seja, reúne os esforços na *SBBT* com a conjunção de equipamentos reais e simulados.

Os últimos grupos focam no projeto dos AVCs visando características que possibilitem a distribuição, reutilização e expansão através da modularidade e a comunicação e integração através da interoperabilidade do sistema como um todo.

Na sequência da proposta de arquitetura cada um destes grupos será melhor descrito e suas funcionalidades devidamente interconectadas conceitualmente.

4.4 Módulos

Extrapolando a Fig. 38 e agrupando funcionalidades e características diferentes, podem-se organizar alguns módulos principais: i. módulo de organização de material educacional (espaço compartilhado de mídias educacionais teóricas); ii. módulo de interface social tridimensional; iii. módulo responsável pela adaptabilidade e “inteligência” do AVC; e iv. módulo de cenários práticos (espaço compartilhado de experimentação). Naturalmente, para se obter uma troca de informações entre estes módulos principais, dois “módulos” adicionais são necessários; v. repositório de dados (“centralização” e “sincronia” das informações); e vi. *middleware* de comunicação (responsável por suprir a troca de informações entre os módulos). Cada um desses será apresentado em detalhes e suas funcionalidades descritas nas subseções seguintes.

4.4.1 Módulo de Espaço Compartilhado de Mídias Educacionais Teóricas

Qualquer método educacional efetivo de alguma forma ou de outra tem seu cerne em materiais teóricos, mesmo que indiretamente. Logo, toda proposta de arquitetura para AVCAEs contempla um espaço para visualização de mídias educacionais que ilustrem a teoria de determinado curso ou objeto de aprendizagem. Por objeto de aprendizagem entende-se: elementos de um tipo de instrução baseados em computador, construídos sob o paradigma da ciência da computação, permitindo a construção de pequenos componentes, os quais podem ser reutilizados inúmeras vezes em diferentes contextos de aprendizagem¹ (WILEY, 2000).

AVCEAs não somente necessitam mas devem suprir um espaço compartilhado onde a colaboração e a troca de mensagens possa ocorrer livremente e estas funcionalidades, suporte a mídias educacionais e à colaboração, são normalmente desempenhadas por ferramentas notoriamente empregadas na EaD. Estas ferramentas são os AVAs (vide Seção 2.7.2). Portanto, a arquitetura proposta obviamente considera a utilização de AVAs como módulo que supre a disponibilização, controle de visualização e gerenciamento de objetos educacionais através da identificação de usuários que acessam o AVCAE (por consequência o AVA).

Todo material didático ou *LO* composto por mídias eletrônicas necessita ser cuidadosamente desenvolvido para provocar tanto para desafiar quanto motivar estudantes a refletirem e colaborarem de forma a resolver “problemas” ou questões investigativas propostos. A metodologia de desenvolvimento dos *LOs*, portanto, segue teorias construtivistas em que há literalmente a construção de conhecimento pelos estudantes. As metodologias de aprendizado baseado em problemas (*PBL*) e do aprendizado investigativo (*IBL*) são duas “técnicas” de ensino-aprendizado ativo onde o estudante por meio de reflexão, investigação e interação constrói o seu conhecimento. Fundamental nesta técnica é prover o espaço compartilhado para a comunicação entre os usuários, já que sem comunicação não há interação entre estudantes e muito menos colaboração.

¹ não há um consenso na comunidade científica para uma definição única de objetos de aprendizagem (*LOs - Learning Objects*) e até o comitê de padronização internacional de tecnologias educacionais propõe uma definição ligeiramente diferente da apresentada. Para o *IEEE LTSC* os objetos de aprendizagem são “entidades, digitais ou não, que podem ser usadas, re-usadas ou referenciadas durante o ensino com suporte tecnológico”.

O AVA, a fim de se integrar com o restante da arquitetura de AVCEA, deve possuir interface tradicional *Web* e suportar armazenamento “externo” de informações para obtenção de dados através do *middleware* (módulo posteriormente descrito). Idealiza-se na arquitetura de AVCEA proposta um AVA que desempenhe no mínimo as condições ilustradas na Fig. 39 abaixo. O armazenamento dos dados em um repositório ou banco de dados acessível externamente ao AVA é crucial para a interconexão das informações entre os módulos. O espaço compartilhado une usuários dispersos sob um único ambiente onde há troca de informações e, conseqüentemente, interação pessoal.

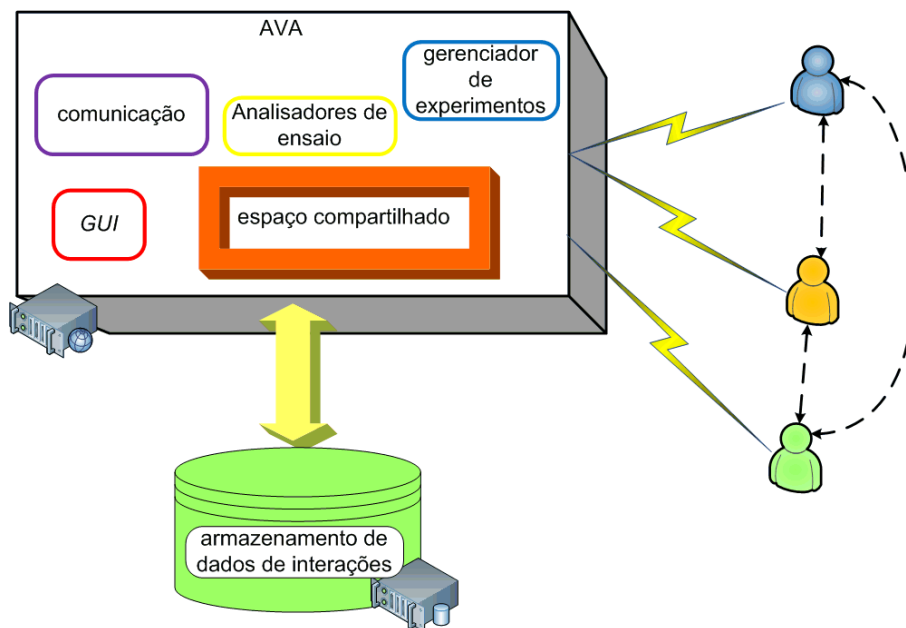


Figura 39: Módulo de espaço compartilhado de mídias educacionais teóricas - diagrama.

4.4.2 Módulo de Interface Social 3D

De forma a enriquecer o conteúdo e chamar/prender a atenção de usuários de ambientes virtuais, que geralmente possuem interface *Web* tradicional, o emprego de interfaces tridimensionais é encorajado, pois supre necessidades psicológicas, lúdicas e sociais. Interfaces *Web* tradicionais (visualizadas por *Web Browsers*) geram efeitos psicológicos que muitas vezes são negativos para a colaboração de usuários, pois fazem com que estes se sintam meros observadores “sem corpo” (KIRRIEMUIR, 2007). Em ambientes tridimensionais (mais frequentes em jogos eletrônicos de “primeira pessoa” como *MMOGs*), participantes do ambiente são representados por projeções corporais, *avatars*, o que permite um efeito psicológico de “presença” (mesmo que virtual) no ambiente. Outros *avatars* também podem eventualmente estar presentes aumentando assim o “sentimento de presença” e a “imersividade” do ambiente (vide Fig. 37 e Seção 2.7.4).

A partir dos estudos conduzidos aqui percebe-se que é difícil conceber a colaboração entre usuários de um sistema virtual sem que estes possam formar laços sociais (amizades). Desta maneira, características de redes sociais (vide Seção 2.9) são importantes e devem ser incorporadas a AVCEAs. Tarefas lúdicas possuem reconhecidamente a capacidade de prender a atenção de estudantes se comparados a cenários de aprendizagem tradicional, como leituras, exercícios de repetição, etc. Da mesma forma, interfaces lúdicas aplicadas a tarefas virtuais, além de enriquecer a representação, prendem a atenção

dos estudantes. Sendo assim, AVCEAs devem incorporar ferramentas de comunicação e promover intercâmbio de ideias entre os usuários.

Embora incorporado no modelo de referência de arquiteturas de AVCEAs este módulo não exclui a interface tradicional do módulo anterior (AVA). Ambas as interfaces coexistem e são interoperáveis apesar das diferentes funcionalidades. O módulo interface 3D social se espelha em funcionalidades de implementações dos mundos 3D com suporte social (vide Seção 2.12).

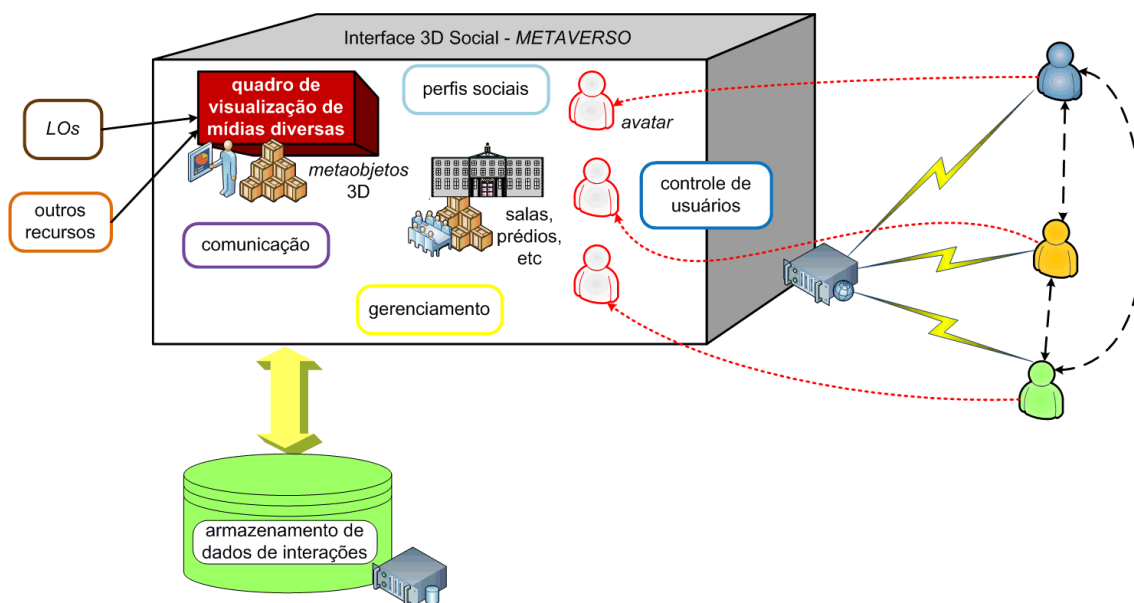


Figura 40: Módulo de interface tridimensional social - diagrama de idealização.

A Fig. 40 ilustra a idealização desta interface, onde existem representações tridimensionais para os mais variados elementos de AVCEAs. Usuários são representados por *avatars*, *LOs* teóricos podem ser visualizados em quadros de visualização (painéis virtuais). Outros *LOs* que tenham alta interatividade podem, por sua vez, ser modelados em objetos virtuais tridimensionais que lembram a representação bidimensional tradicional. A interação social se dá *avatar-a-avatar* através de troca de mensagens ou mesmo por áudio no *metaverso* (*in-world*). Cada *avatar* terá a possibilidade de criar um perfil social e gerenciar suas criações ou colaborar em criações compartilhadas. A organização dos objetos 3D deve lembrar ambientes tradicionais de ensino como prédios, salas, equipamentos, etc. deixando o ambiente intuitivo e possibilitando a imersividade do usuário. Pode-se argumentar que o *overhead* de interação e de possibilidades distrai o estudante e pode afastar do objetivo original da interface que é prender a atenção, mas de outro ponto de vista a distração também pode ser positiva aos estudantes que estão acostumados a aprender se distraindo. Este fato se deve a evolução natural dos métodos de aprendizagem dos estudantes expostos às tecnologias desde o berço. Alguns educadores referem-se a faixa etária de estudantes que tiveram intensa interação desde criação com computadores e Internet de “geração informação ou Internet” (*Net/information age*).

Esta interface, por sua vez, também possui conexão com o módulo de armazenamento de informações e registra interações dos usuários com qualquer objeto e de comunicações entre usuários. O espaço compartilhado é naturalmente o *metaverso* visualizado pela interface.

4.4.3 Módulos Associados à Experimentação

O próximo grupo de módulos representa os cenários práticos e de colaboração e cooperação na experimentação. Como exposto anteriormente, nas áreas técnicas, sobretudo nas engenharias, é necessária a experimentação. É de conhecimento geral que laboratórios possibilitam aplicação e testes de conhecimentos teóricos em situações práticas (AUER et al., 2003), ou seja, experimentação. Laboratório virtual é um termo comumente utilizado para descrever a interface a experimentos remotos ou virtuais. Um laboratório virtual é composto por um ou mais experimentos empregados para ilustrar um conceito teórico.

Do ponto de vista pedagógico, a utilização laboratorial proporciona conceitos de aprendizado ativo, aprendizado distribuído e aprendizado de grupo. Os experimentos remotos² são alternativas econômicas para laboratórios reais com equipamentos de alto custo (COOPER, 2000). Esta alternativa de laboratórios proporciona aos estudantes “a segunda melhor alternativa de estar lá” (SBBT) (ATKAN et al., 1996)).

O estudo deste grupo de módulos teve grande influência de trabalhos anteriores do autor, sobretudo em sua dissertação de mestrado (SCHAF, 2006), onde uma proposta de arquitetura para integração de experimentos remotos de realidade mista foi apresentada. Nesta proposta foi descrita a estratégia de componentes intercambiáveis focando na maximização de cenários educativos. Estes componentes foram inicialmente associados a partes de sistemas de automação, ou seja, ou plantas ou controladores. Para este caso tanto sensores quanto atuadores fazem parte da planta, mas estes também podem ser, por sua vez, componentes distintos tanto simulados (virtuais) quanto reais. A Fig. 41 ilustra o diagrama dos cenários possíveis usando esta estratégia de intercâmbio de componentes de sistemas de automação. Embora aqui ilustrado somente para sistemas de automação, estes componentes podem ser naturalmente e analogamente empregados em outros sistemas. Existem algumas denominações dos cenários. A configuração, usando controladores reais com plantas simuladas, é comumente chamada na área de sistemas embarcados e prototipação rápida de *Hardware-in-the-Loop (HiL)*, enquanto que o oposto, ou seja, controlador simulado com planta real, se chama antonimamente de *Software-in-the-Loop (SiL)*. Uma outra forma similar ao *SiL* especificamente quando o controlador é um CLP (Controlador Lógico Programável), é chamada de *softPLC*.

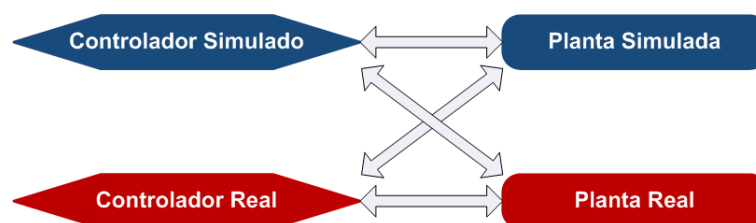


Figura 41: Componentes intercambiáveis - diagrama de cenários possíveis.

Os componentes intercambiáveis são possíveis empregando técnicas de realidade mista à experimentação (vide Seção 2.6). A conexão de componentes reais com virtuais se dá através do uso de hiperconexões. A expansão de cenários educacionais com o emprego de componentes, tanto reais quanto virtuais e em combinações diversas, possibilita ilustrar conceitos presentes em ambos os “mundos”, virtualidade e realidade. Isto é muito

²experimentos remotos: na área da EaD são organizados dentro da linha de pesquisa chamada de telemática; e em publicações científicas internacionais também são chamados de *WebExperiments* ou *WebLabs*.

útil na área da engenharia elétrica e da computação, pois normalmente fenômenos físicos difíceis de se demonstrar podem ser apresentados em simulações didáticas. Nelas o tempo de ensaio pode ser manipulado oferecendo visualização passo-a-passo. Outra vantagem desta associação é o acesso a informações do ensaio impossíveis fisicamente, como por exemplo, medição de campos elétricos em máquinas ou visualização do interior de elementos do experimento, etc. Apesar de vantajosos em alguns aspectos, elementos virtuais, por sua vez, possuem normalmente o caráter de desfiguração da realidade, ou seja, a perda da conexão com a realidade. Esta conexão pode ser retomada com o emprego de componentes reais que oferecem características necessárias para o treinamento efetivo de engenheiros, como instabilidades, não-linearidades, interferência causada por ruídos, etc. Esse elo entre realidade e virtualidade adiciona elementos educacionais ilimitados e flexibilidade de criação para instrutores.

O suporte a espaços compartilhados é justificado pela necessidade de colaboração também durante o ensaio, pois esta é também uma tarefa investigativa e que requer reflexão dos estudantes. A colaboração através de interações no experimento em conjunto (no espaço compartilhado) instiga a crítica e a curiosidade mútua.

O uso de interfaces de *feedback* de força, e outros esforços físicos (*haptics*) eleva a realidade de experimentos e há diversos estudos para o emprego de conceitos de *hyperbonds* para a interconexão dos esforços mediados por computador. Este conceito também está incorporado neste grande grupo de módulos, embora ainda sejam poucos os estudos desta linha de pesquisa associadas à educação. Há, entretanto, diversos estudos para treinamento profissional e na teleoperação de máquinas remotas onde a presença humana é dificultada.

Na Fig. 42 estão ilustrados os módulos e também os conceitos associados (caixas pontilhadas) ao grupo de experimentação da arquitetura proposta. Nota-se a presença do gerenciador e do espaço compartilhado em destaque, pois estes são os grandes responsáveis pela organização das funcionalidades. É no espaço compartilhado que usuários dispersos se reúnem virtualmente e visualizam e realizam o(s) ensaio(s) no(s) experimento(s) que é/são controlados pelo gerenciador de experimentos responsável por estabelecer as conexões entre os componentes do experimento. Estão também ilustrados os módulos de sistema de agendamento (*booking systems*) e os analisadores de experimento/ensaio. No primeiro há um controle de uso do experimento que, conforme a configuração, agendará recursos aos usuários que somente nesta reserva poderão utilizar tais recursos. Naturalmente, componentes reais não restringem (inerentemente) o uso simultâneo de diversos usuários, por este motivo os sistemas de agendamento são necessários para organizar esta restrição e o uso individual (ou de grupos definidos de usuários). Para experimentos onde há somente componentes virtuais normalmente a replicação é natural, e várias instâncias podem ser simultaneamente utilizadas por grupos distintos de usuários. O analisador de experiência recolhe informações referentes aos ensaios executados pelos estudantes. Estas informações podem ser usadas futuramente para *feedback* educacional. Como nos outros módulos apresentados o armazenamento de informações também está presente.

Existem muitas ferramentas e meios de comunicação que podem ser empregados para o acesso de experimentos remotamente. Na Seção 2.8 foram estudadas algumas arquiteturas de comunicação para acesso remoto de experimentos e também interfaces e diferentes configurações de componentes dos laboratórios (vide Seção 3.4). A topologia de acesso remoto de maior aceitação é a de *thin-client* onde um simples navegador *Web* com *plugins* tradicionais (*JRE* ou *Flash*) são suficientes para suprir o acesso aos usuários. No lado do servidor são empregadas as mais diversas ferramentas de *software* e ainda não há um

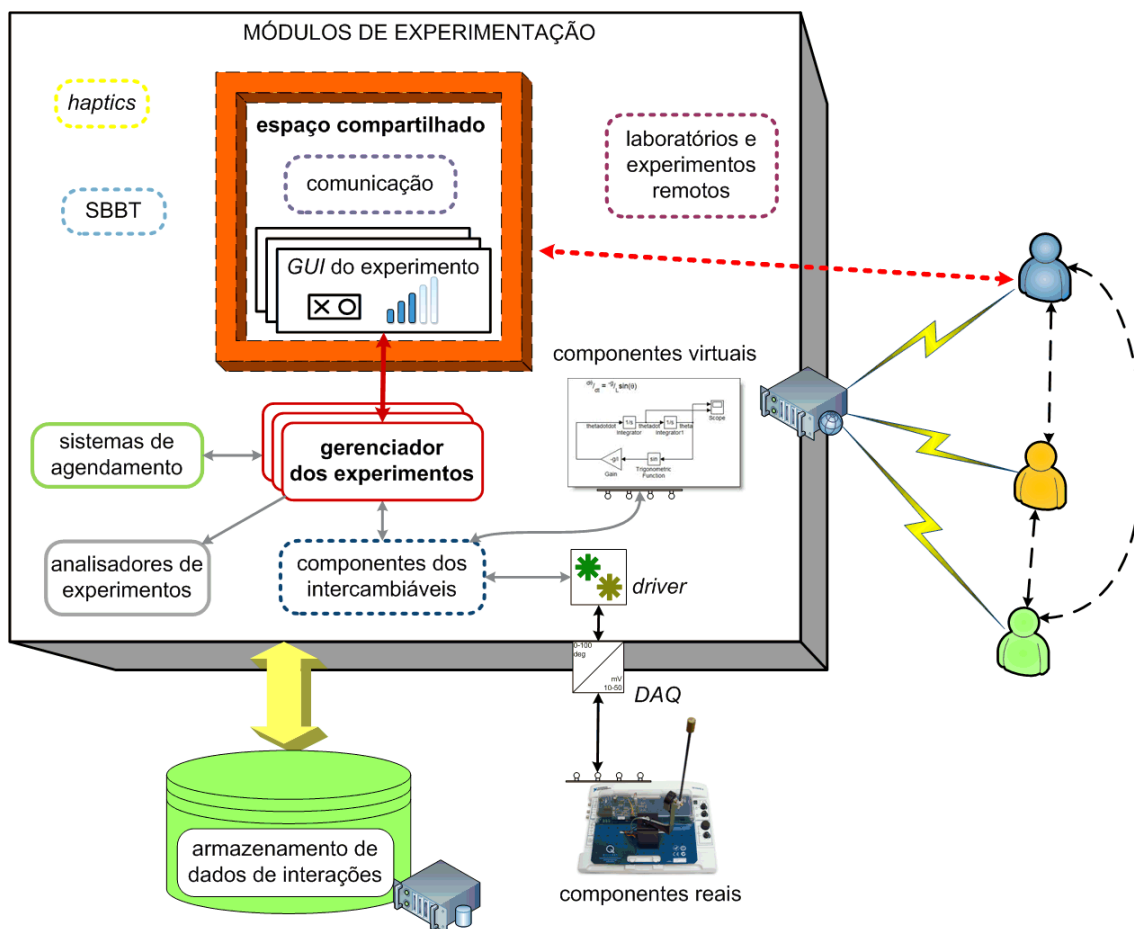


Figura 42: Módulos do grupo de experimentação - diagrama de idealização.

consenso de arquitetura ideal, embora trabalhos onde ferramentas simples com sistemas baseados em *Java*, *AJAX* ou *PHP* para elaboração de serviços *Web* possibilitem alto grau de efetividade e reusabilidade com baixo custo.

Ferramentas *FOSSs* para implementação de *WSs* (*Web Services*) apresentam os melhores resultados, pois geralmente possuem características de independência de plataforma e compatibilidade com diversos dispositivos e navegadores *Web*.

4.4.4 Módulos Associados à Adaptação de Conteúdo e de *Feedback* ao Usuário

A adaptação de conteúdo (*content adaptation*) é uma área de pesquisa que busca soluções para a personalização ou individualização de conteúdos, segundo solicitação ou necessidade dos usuários. Assim, mesmo que a fonte do conteúdo seja a mesma, a visualização (configuração) pode ser adaptada de várias maneiras diferentes para atender diferentes tipos e classificações de usuários. Esta adaptação é importante também no ensino-aprendizagem, pois, embora tradicionalmente estudantes sejam tratados igualmente por instrutores, estes têm consciência da variação da aprendizagem e também do conhecimento de cada um (MAGOULAS; PAPANIKOLAOU; GRIGORIADOU, 2003). Pode-se ilustrar que, enquanto estudantes com facilidade autodidática necessitam normalmente de livros-texto e alguns exercícios para adquirir o conhecimento necessário em determinado curso, outros necessitam interação, prática, e, possivelmente, contato com a realidade para obter o mesmo nível dos estudantes autodidatas. Assim, notoriamente em AVCEAs onde o professor não tem a devida percepção dos estudantes um sistema

de adaptação de conteúdo é vital. Este recurso também pode ser vital para estudantes com necessidades especiais, pois o conteúdo possa ser adaptado para outros meios de interação/representação, basicamente vídeo para áudio ou o inverso.

Na Fig. 43 está ilustrado um exemplo de sistema que adapta o conteúdo de acordo com critérios diversos dos usuários. Pode-se imaginar que usuários diferentes possuem capacidades de aprendizado diferentes e de acordo com elas são disponibilizados conteúdos diferentes a eles; ou também a adaptação devido a capacidade computacional de cada dispositivo de cada usuário, etc. Esta adaptação é conduzida normalmente por uma “negociação” entre o cliente e o servidor. Dados do cliente são enviados ao servidor, como, por exemplo, nome do navegador (*Web Browser*), tamanho da tela (*display*) em pixels ou real, dados do usuário, recursos de conexão, processador do cliente, etc; e, com base nestes, um gerenciador de recursos, que tem acesso ao universo de conteúdos, envia ao cliente o conteúdo. A adaptação é normalmente transparente aos usuários que não têm, normalmente, acesso ao universo de conteúdos. O conteúdo adaptado é “montado” através de regras de perfis, ou seja, o gerenciador de recursos possui perfis predefinidos de usuários e os respectivos recursos contidos em cada um. Pode-se exemplificar um caso onde um usuário possui restrições de tamanho de tela (caso típico de dispositivos portáteis, como *PDA*s e *smartphones*), assim, este dispositivo é identificado pelo servidor (através de *scripts* contidos no próprio recurso acessado do servidor), e o gerenciador, com base no perfil com restrições de tamanho de tela, modifica (adapta) o conteúdo do recurso acessado para somente documentos textuais e com figuras em tamanho reduzido que não ultrapassem o limite do tamanho da tela.

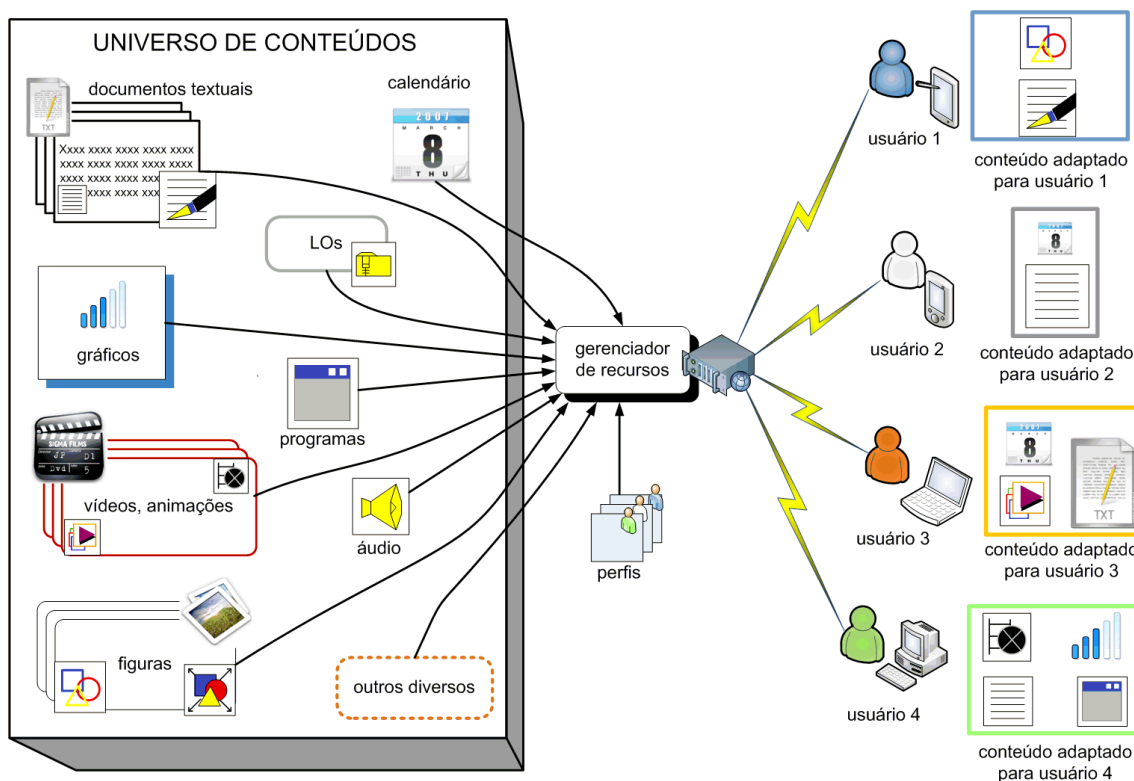


Figura 43: Adaptação de conteúdo - exemplo para quatro usuários distintos.

A adaptação da interface é comumente empregada para visualização de dados, inclusive para controle e supervisão remotos (PEROZZO, 2007). Esta “técnica”, no entanto, ainda é pouco utilizada na educação. A adaptação segue critérios ou perfis configurá-

veis de acordo com um conjunto de informações. O trabalho de LUM; LAU (2002) cita algumas destes parâmetros de configuração de adaptação, enquanto que no trabalho de RUMETSHOFER; WÖB (2003) é desenvolvido um framework com base em informações psicológicas do usuário para direcionar o aprendizado. A coleta destas informações para adaptação é constante e dinâmica. Para tal, são necessários registros de interações de usuários e ferramentas de refinamento de seleção destes dados para a criação de perfis. Estas ferramentas se baseiam em técnicas conhecidas como *data mining*. Como coletar informações em sistemas com múltiplos usuários dinamicamente é um desafio computacional, já que isso pode demandar muito processamento ou programas computacionais complexos. É neste nicho que surge a possibilidade da utilização de sistemas multiagente (SMA) que possuem funcionamento naturalmente autônomo, distribuído e paralelo (concorrente) (vide Seção 2.13).

Agentes coletores de informações seriam chamados de “monitores” que garimpariam os dados necessários e os disporiam de maneira a organizar diferentes perfis de mídias que estariam disponíveis para diferentes usuários. Este comportamento pode ser ilustrado na Fig. 44, onde existe um monitorador para cada usuário (o “disparo”/replicação de agentes é natural em SMA) e estes coletam os dados de interações dos usuários no AVC através de acesso ao módulo de armazenamento e transmitem estas informações (devidamente tratadas) ao agente detector de perfis que cria/atualiza os perfis do banco de perfis (de usuários). Outro agente adapta o conteúdo ao usuário com nas informações do banco de perfis, do *framework* de conteúdo e do universo de conteúdos. Essas informações, por sua vez, podem, naturalmente, também estar armazenadas com os outros dados no módulo de repositório de informações. O *framework* serve como modelo parametrizado do conteúdo, ou seja, baseado em valores de parâmetros do perfil do usuário o conteúdo é configurado. Uma simples exemplificação para este processo poderia ser descrito por um controlador de sequencia, isto é, o código:

```
IF (width > image.width()) echo "<img src=figura.jpg>";
```

descrito em *PHP*, onde *width* seria a largura do *display* do cliente, e *image.width()* a largura da figura a ser mostrada ou não (adaptação de conteúdo).

Agentes monitoradores podem ainda adaptar conteúdo com base em informações do analisador de experiência apresentado na subseção anterior, agindo assim como tutores autônomos baseados em informações de ensaios em experimentos didáticos. O uso de SMA expande o uso de tutores simplificados, baseados em ensaios, que foi proposto anteriormente pelo autor (SCHAF, 2006).

Embora os módulos que apresentam espaços compartilhados e ferramentas de interação entre usuários ofereçam infraestrutura e meios de comunicação para que usuários colaborem, o processo de colaboração é extremamente delicado e exige que os usuários tenham consciência das ferramentas disponíveis no ambiente e das ações de outros usuários, para a colaboração (GUTWIN; GREENBERG, 2002). Esta consciência pode ser adquirida com o tempo de uso do ambiente, exigindo paciência e concentração dos usuários. Para evitar que usuários possam rapidamente se frustrar pelo não conhecimento do ambiente (ou sistema), isto é, pela falta de consciência, um auxílio aprimorado à colaboração é proposto. Alguns trabalhos de “tecnologias de colaboração (vide Seção 3.7) pesquisados referenciam este auxílio como suporte à consciência de colaboração (*collaboration awareness support*) (NEALE; CARROLL; ROSSON, 2004).

A coleta de informações também possibilita o projeto de sistemas que proporcionam *feedback* de aprendizagem. Baseado nos históricos estudantis e de interações, e em resultados práticos (experiências, exercícios, tarefas, trabalhos, projetos, questionários, etc)

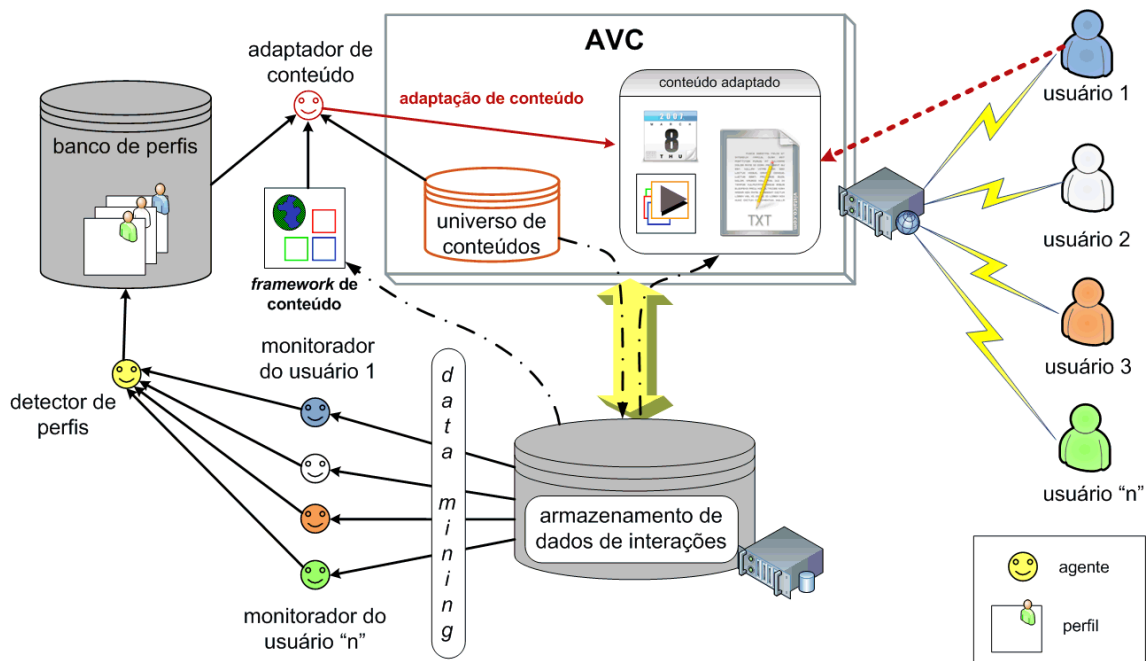


Figura 44: Módulo dos monitores de interações - exemplo da coleta de informações e organização de perfis para adaptação de conteúdos.

um sistema tutor pode interagir com o estudante, inferindo *LOs*, sugerindo a visualização de resultados de outros estudantes, incentivando a cooperação e colaboração na aprendizagem ou nas tarefas práticas, etc. Esta linha de pesquisa, comumente referida como sistemas de tutoriamento inteligente (*ITS*), ainda se encontra em estágio embrionário de evolução. Alguns resultados de pesquisas foram apresentados na Seção 3.8. A autonomia do tutor pode ser alcançada usando analogamente a adaptação de conteúdo por SMA.

A Fig. 45 ilustra a idealização de funcionamento dos diversos agentes envolvidos para tutoriamento autônomo com responsabilidades diversas, entre elas adaptação de conteúdo, *feedback* educacional e suporte à colaboração ativa. Para fins de simplificação o analisador de ensaio (da experiência estudenteprática) foi incorporado ao tutor. Percebe-se que ações do usuário se refletem no ambiente que devolve o conteúdo adaptado através da coleta de informações do repositório. O tratamento das informações inicia no módulo monitorador (agentes) que através de leituras periódicas informa o tutor de mudanças no repositório (interações do usuário com o AVC). O módulo tutor, por sua vez, inicialmente composto por um agente mestre responsável por iniciar (“disparar”) um agente-tutor para cada usuário conectado, configura e armazena perfis de conteúdo do usuário de acordo com seu histórico de interações e ensaios práticos. Estes perfis são usados pelo gerenciador de conteúdos que adapta os elementos e retorna ao AVC e, por consequência, torna-se disponível ao usuário. Na figura são ilustradas quatro interações de um único usuário/estudante: conexão, *login*, requisição de *LO* e ensaio de experimento. A extrapolação para multiusuários é natural, mas de difícil representação gráfica. Importante notar que, quando há dados suficientes, o tutor também desempenha o papel de facilitador de colaboração sugerindo adaptação de conteúdo para incluir elementos de outros usuários e ferramentas de comunicação.

A adição de dados dos estudantes ao repositório, como histórico de disciplinas, notas, dificuldades e mapas conceituais de estudantes, também podem ser usados para a avaliação de perfis e inferência e/ou adaptação de conteúdos. Algumas pesquisas com modelos

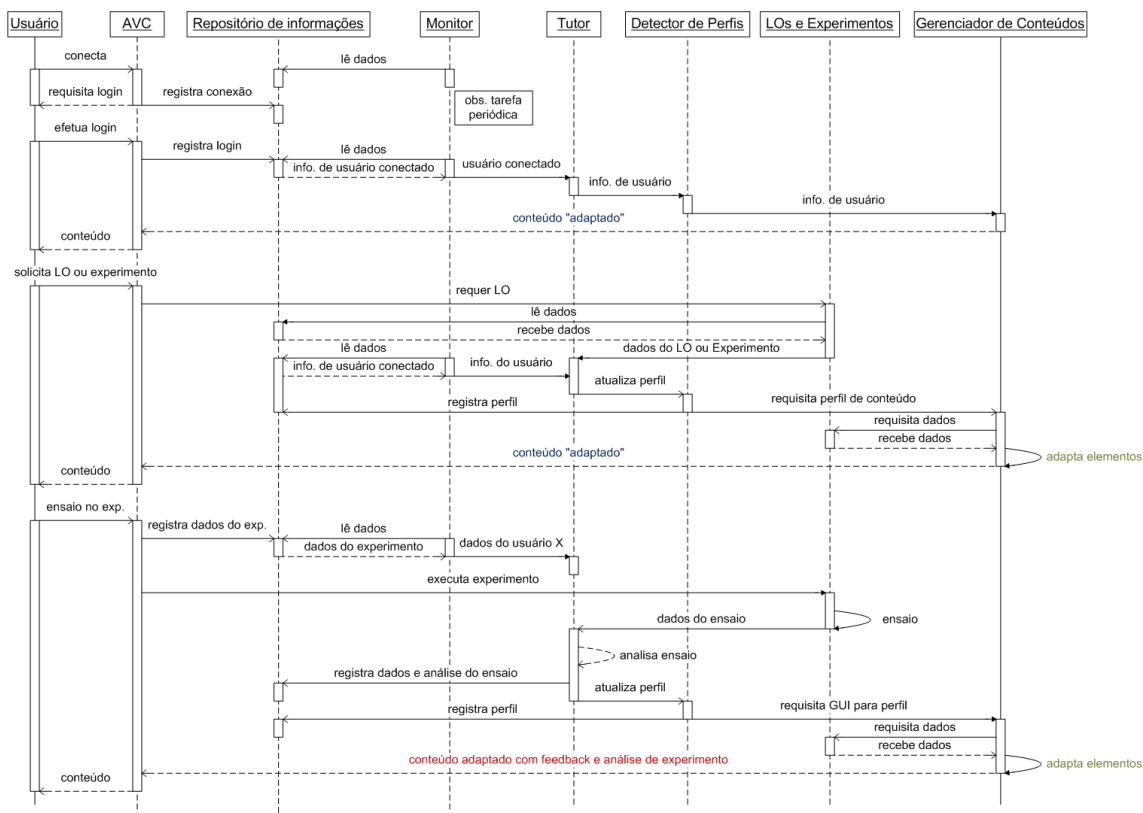


Figura 45: Módulo dos tutores autônomos - diagrama sequência *UML* para as diversas tarefas e agentes envolvidos.

de estudantes demonstraram avanços na avaliação do nível de conhecimento dos estudantes (NOGUEZ; SUCCAR, 2007), mas este trabalho necessita de dados de um questionário prévio o que normalmente não têm a simpatia dos estudantes.

A Fig. 46 ilustra a organização dos módulos do grupo de *feedback* e adaptação de conteúdo como também as interações entre eles. SMA são empregados em todos os módulos deste grupo. O sistema de tutoriamento autônomo inclui conceitos de *ITS* e adaptação de conteúdo, e o módulo monitores emprega técnicas de *data mining* para refinamento de dados em informações úteis. É importante frisar que os módulos deste grupo diferentemente de outros, anteriormente apresentados, não possuem funcionamento independente, pois dependem de outros módulos onde há interação direta com o usuário para a coleta de informações que servem para inferir novos dados de volta (*feedback*) a módulos onde há visualização/interação.

A associação com *ITSs* é nítida, já que a “inteligência” (adaptabilidade e *feedback*) está centrada nos módulos deste grupo, especialmente no sistema de tutoriamento autônomo. A automação também é evidente, já que o AVC com estes módulos possibilita a resposta automática a interações dos usuários sem a necessidade de intervenção do professor/instrutor. Obviamente que a autonomia do sistema é restrita, devido ao fato das ações serem previamente programadas e seguirem uma “tabela” de instruções de acordo com os dados coletados. Inteligência adicional pode ser obtida usando técnicas de inteligência artificial associada aos tutores autônomos. Embora este tema não tenha sido abordado neste trabalho a integração destas técnicas é futuramente desejável, assim como a utilização de modelos de estudante dinâmicos (baseados em probabilidades, lógicas *fuzzy*, etc).

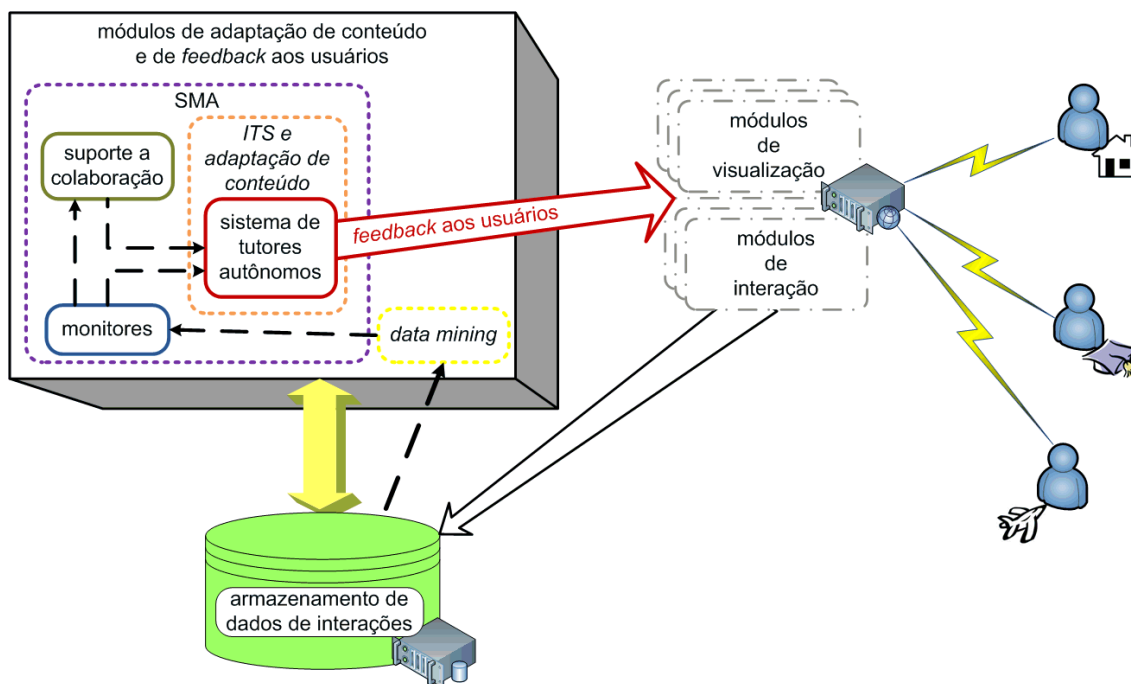


Figura 46: Módulos do grupo de adaptação de conteúdo e de *feedback* aos usuários - diagrama e interações.

4.4.5 Repositório de Informações

A característica do repositório de informações não é exatamente de um módulo na arquitetura, mas sim de um banco de dados de interações onde todas as outras informações pertinentes e comuns aos outros módulos idealizados estão armazenadas. Como explicitado nas seções anteriores, praticamente todos os módulos possuem conexão com o repositório de informações, lendo e/ou escrevendo dados. Por esse motivo o repositório é responsável pela centralização e sincronia dos dados entre módulos. Estes dados, por sua vez, são compostos por informações de registro, os chamados *logs*, e de interação de qualquer natureza.

Comumente, bancos de dados são empregados quando há necessidade de armazenamento, organização e gerenciamento de uma grande quantidade de dados, especialmente quando podem ocorrer leituras ou escritas concorrentes, isto é, computacionalmente simultâneas (“ao mesmo tempo”). Embora, na realidade, se tenha como diferenciar o exato início da leitura e escrita de dados, vale lembrar que estes processos não são atômicos e necessitam um tempo computacional para completar a tarefa. A leitura de um mesmo dado (informação) antes da escrita deste ou a alocação de recursos pode levar a erros que são comuns nessas operações, mas que podem ser devidamente tratados por gerenciamento de banco de dados, que naturalmente possuem mecanismos para evitar tais erros de sincronia.

A fim de manter a capacidade de distribuição da arquitetura o repositório de informações também pode ser composto por vários bancos de dados interconectados e dispersos. Na Fig. 47 está ilustrado um exemplo com módulos da arquitetura que possuem repositórios distribuídos. Nota-se a interconexão de dados e a distribuição de recursos possibilitadas pela modularidade da arquitetura.

Importante notar que o repositório de informações é transparente para o usuário que não interage diretamente com os dados nele armazenados e nem tem conhecimento de sua

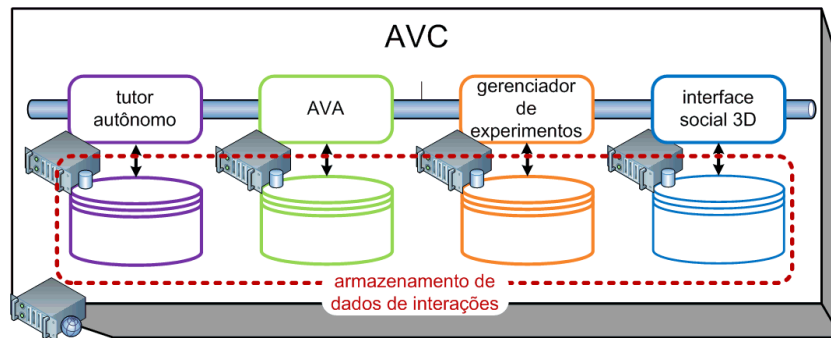


Figura 47: Repositório de Informações - exemplo de distribuição e interconexão.

estruturação.

4.4.6 Middleware

O projeto do *middleware* é talvez o maior desafio na arquitetura proposta, pois é ele o responsável pela interconexão dos módulos. Esta conexão pode ter as mais variadas funcionalidades e envolver qualquer um dos módulos. Entre as conexões podem-se citar tanto as leituras, atualizações e escritas de dados no repositório de informações quanto os comandos de execuções de certas tarefas em módulos distintos. Conexões envolvendo somente a troca de dados entre módulos e o repositório de informações são simples, mas as que envolvem, por exemplo, módulos projetados usando agentes do SMA e o gerenciador de conteúdos ou a interface social 3D envolvem maior complexidade e, por consequência, dificuldade elevada.

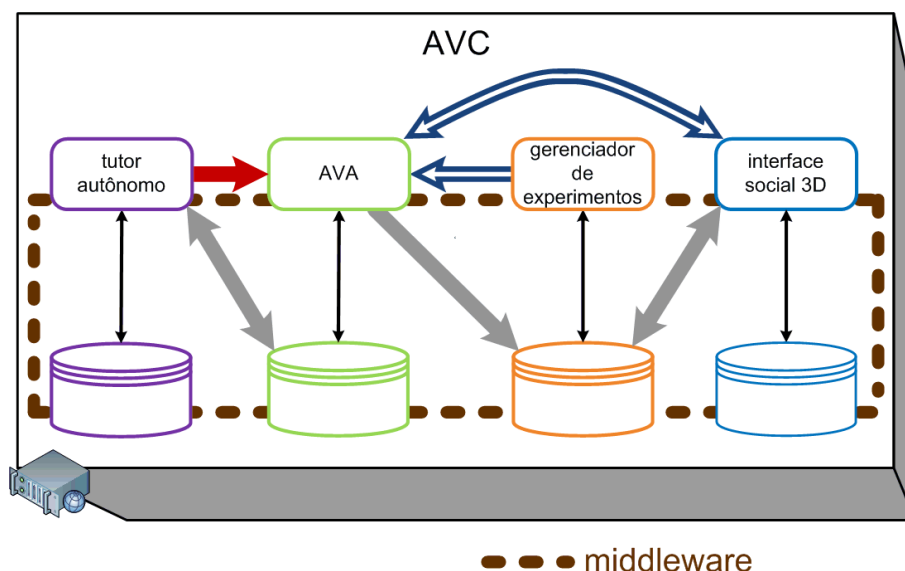


Figura 48: *Middleware* da arquitetura - exemplo de funcionamento com conexões.

A exemplo do repositório de informações, o *middleware* não pode ser considerado como um módulo, não tem funcionamento independente e é transparente ao usuário. Na Fig. 48 estão ilustrados alguns exemplos de conexões possibilitadas através do *middleware*. As flechas entre módulos representam as trocas de informações. Por exemplo, a flecha cheia escura do módulo tutor autônomo ao AVA representa a inferência de mudança de conteúdo no AVA. As flechas cinzentas entre módulos destinadas ao repositório

de módulos diferentes representam a leitura e a escrita de dados que, por fim, causam a mudança de conteúdo do módulo (inferência indireta). As flechas vazadas representam conexões de trocas de conteúdo *Web*.

Há tipos de conexões diferentes de acordo com os dados envolvidos e os módulos. Como apresentado anteriormente, praticamente todos os módulos têm conexão com o repositório de informações direta ou indiretamente. Embora eficiente e naturalmente simplificada, a interconexão através da escrita e leitura de banco de dados exige o uso de tarefas periódicas, pois não há informação da atualização de dados sem a verificação. A Fig. 49 exemplifica as diferentes “interações” entre os módulos AVA e o gerenciador de experimentos. Claramente se nota a escrita aperiódica (sob demanda do usuário) e a leitura periódica (repetidas leituras espaçadas por um tempo constante) do AVA no repositório e do gerenciador de experimentos no repositório respectivamente. Assim, a interação do AVA no gerenciador de experimento é indireta (flecha pontilhada), pois se dá através do repositório e sem sincronia, já que o repositório não “avisa” a mudança de parâmetros ao gerenciador. Este percebe a mudança através de repetidas leituras. Em contraste, a atualização da visualização dos dados do ensaio enviada ao AVA pelo gerenciador de experimento é direta e periódica, pois o *GUI* do experimento está integrado diretamente no AVA e não necessita de leituras repetidas do repositório. Isto exemplifica a relação *pró versus* contra de interações diretas e indiretas usando o *middleware*, pois em um tem-se a facilidade de uso somente de interfaces de comunicação com o repositório de informações e em contrapartida a necessidade de leituras periódicas; e em outro tem-se a necessidade de desenvolvimento de uma interface de comunicação entre módulos específicos (interação direta), mas que elimina a interação indireta e as leituras periódicas ao repositório de informações.

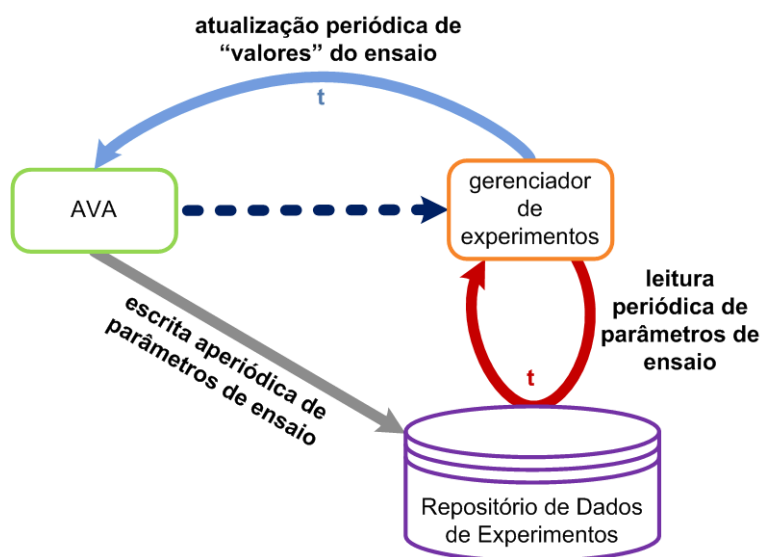


Figura 49: *Middleware* de conexão - exemplo de interações diretas e indiretas.

O balanço correto entre o desenvolvimento de interfaces de comunicação e a qualidade da interação entre os mais variados módulos é a tarefa do *middleware* idealizado. As interfaces de comunicação implicam o uso de topologias diferentes de conexão, e isto impacta diretamente nas características de interoperabilidade e integração de módulos. Por exemplo, módulos que usam a interface de comunicação com bancos de dados *SQL* (*Structured Query Language*) não são inerentemente compatíveis com interfaces de co-

municação que empregam chamadas de procedimento remotas (*RPC - Remote Procedure Call*).

É importante, portanto, possibilitar a maior interoperabilidade possível entre módulos empregando tanto ferramentas de *software* possivelmente compatíveis quanto amplamente abertas, e também protocolos de comunicação vastamente abrangentes que sejam independentes de plataforma e simplificados para o uso via *Internet*. A revisão efetuada na Seção 3 ilustra diversas ferramentas, possível interoperabilidade entre elas e alguns exemplos de projetos corretos de sistemas para integrabilidade via *Web*.

4.5 Arquitetura para AVCEAs

De forma a garantir que usuários dispersos se comuniquem colaborando em prol comum é proposta uma arquitetura para ambientes computacionais de suporte à colaboração voltados ao ensino, ou AVCEAs. Nela estão estruturados o maior número de conceitos e funcionalidades associados a ambientes colaborativos. Características de linhas de pesquisa de *CSCW* (GRUDIN, 1988), *C_SCL* (STAHL; KOSCHMANN; SUTHERS, 2006), entre outras, foram combinadas e organizadas em módulos. Isso traz diversos benefícios para o projeto de AVCEAs, como modularidade, distribuição de recursos, reusabilidade, interoperabilidade, entre outros.

Organizando os módulos (ou grupos) em um diagrama, pode-se ilustrar a arquitetura proposta para AVCEAs com as funcionalidades descritas anteriormente, como na Fig. 50. Nota-se que o *middleware* serve como meio de comunicação entre todos os módulos e que todos podem ter acesso ao repositório de informações que supre a centralização, a sincronia, e o armazenamento de dados. Usuários se conectam ao AVCEA através da infraestrutura da rede mundial de computadores (*Internet*). Estes usuários podem ser professores, tutores, estudantes, colaboradores, ou qualquer outro usuário interessado em usufruir do AVCEA. Através de uma conexão (via *middleware*) os usuários terão acesso à interface social 3D para interação com os conteúdos. A modularidade da arquitetura torna facultativo o uso desta interface pelo usuário, que pode optar por uma interface tradicional *Web* de AVA (incluído no grupo de módulos de espaço compartilhado de mídias educacionais) se desejar. Num mesmo ambiente social, usuários terão suporte à experimentação e colaboração com auxílio de tutores autônomos. Obviamente, como se trata de uma arquitetura modular, a utilização dos módulos é opcional possibilitando diversas combinações com os módulos existentes. Uma exceção de combinação envolve o módulo responsável pela adaptação de conteúdo que é dependente e necessita de ao menos um dos três módulos principais (espaço compartilhado de mídias educacionais, experimentação, ou interface social 3D). Na figura também estão representados alguns conceitos associados aos módulos, como o sentimento de “presença”/“imersão” virtual (*sense of being there*), *metaversos* e *avatars* no módulo de interface social tridimensional; objetos de aprendizagem (*LOs*) e AVAs no grupo de módulos do espaço compartilhado de mídias educacionais; *SBBT*, *haptics* e componentes intercambiáveis no grupo de experimentação; sistemas de tutoriamento “inteligente” (*ITSs*), *data mining* e SMA no grupo de adaptação de conteúdo e *feedback*; e finalmente no *middleware* conceitos voltados à sua funcionalidade como interoperabilidade, expansibilidade e integrabilidade.

O potencial máximo da arquitetura proposta naturalmente é atingido se todos os módulos forem empregados no desenvolvimento do AVC, tendo assim um AVCEA com interface social 3D e suporte à experimentação e colaboração. Esta interface representaria todos os usuários do ambiente virtual como *avatars* e o espaço compartilhado como um

metaverso multiusuário (classificado como *MUVE*).

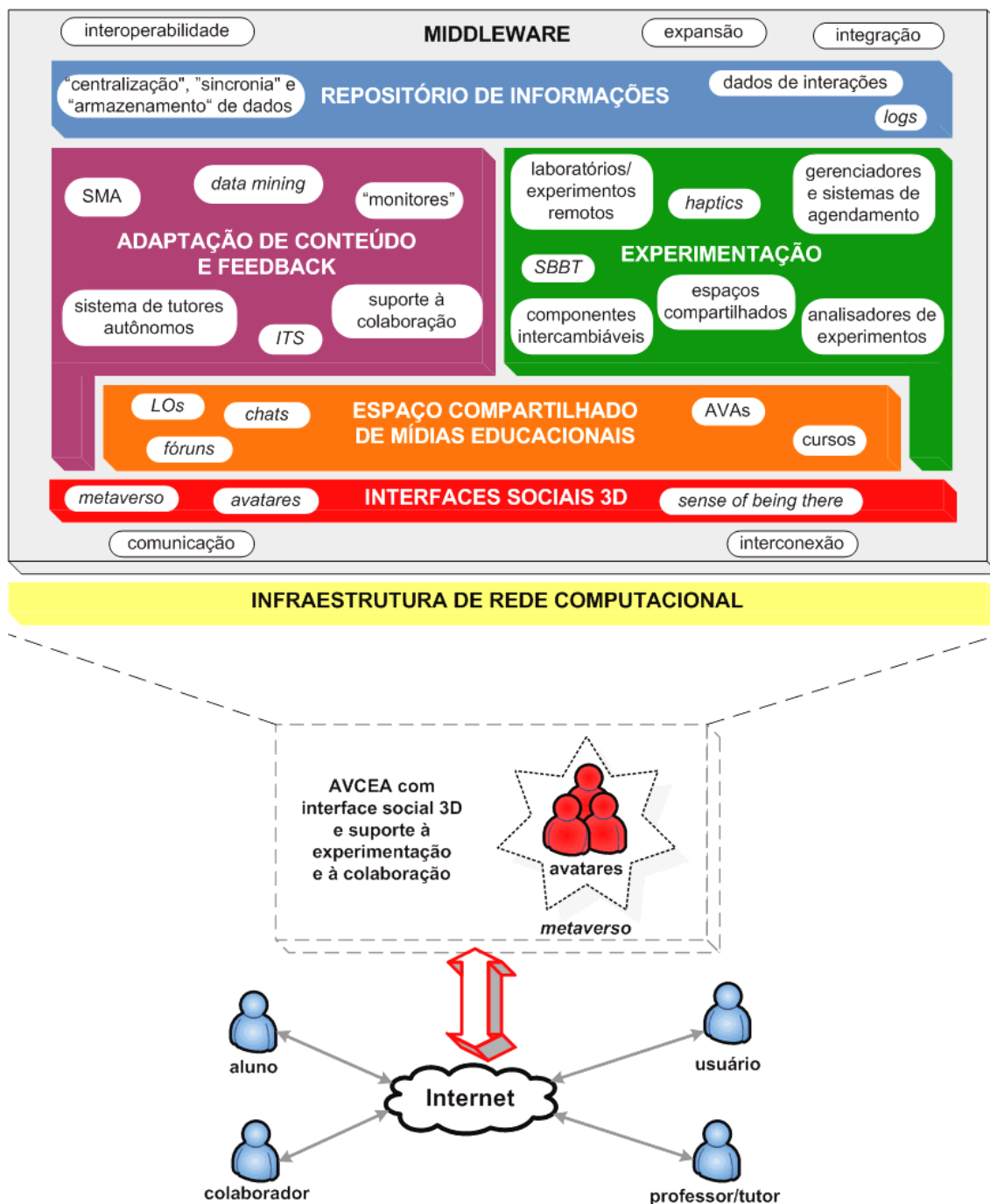


Figura 50: Proposta de arquitetura para AVCEAs - diagrama de módulos e conectividade.

5 IMPLEMENTAÇÃO

5.1 Sinopse

Neste capítulo serão abordadas a praticidade dos conceitos e as técnicas expostas na proposta de arquitetura. Para demonstrar isto serão apresentados os diversos estudos de caso desenvolvidos e para finalizar um protótipo, chamado de *3DAutoSysLab*, que envolve os módulos descritos na arquitetura.

Inicialmente, um estudo das ferramentas e tecnologias atuais será apresentado de maneira a proporcionar um panorama amplo da viabilidade prática de implementações de AVCEAs ligadas às funcionalidades especificadas na arquitetura proposta. Sabe-se, todavia, que implementações alternativas também podem atingir os objetivos propostos.

Muitos dos resultados de implementação estão acessíveis a professores e estudantes dos cursos de Engenharia Elétrica e de Computação na UFRGS. Estes resultados também são parte de projetos e colaborações com o grupo de pesquisa *ArtecLab* da Universidade de Bremen, na Alemanha, com a qual houve intensa cooperação sobretudo durante o estágio de doutorado do autor.

Algumas implementações naturalmente estão ligadas ao trabalho anterior do autor, já que a mesma linha de pesquisa foi mantida (SCHAF, 2006). Estas implementações estão concretizadas e em uso em diversas disciplinas na modalidade aberta (*blended learning*) por alguns professores da UFRGS.

5.2 Panorama das Ferramentas e Tecnologias

A seção de análises e comparações do estado da arte (vide Seção 3.11) deste trabalho já apresentou algumas implementações de trabalhos relacionados. Logo, servirá de primeira base de conhecimento sobre ferramentas que estão disponíveis atualmente para o desenvolvimento de AVCs. Ainda assim foram investigadas ferramentas que melhor se adaptassem às características desejadas e conceituais da arquitetura proposta. Neste quesito, e também devido a trabalhos prévios com algumas ferramentas, foram selecionadas ferramentas *FOSSs* tais que possibilitem a maior flexibilidade para desenvolvimento e modificação e/ou personalização quanto necessário.

Nas próximas subseções cada categoria de ferramentas será apresentada e vantagens e desvantagens descritas sem a divisão ou vinculação com os módulos da arquitetura. Posteriormente, vários estudos de caso serão descritos, cada um ligado a diferentes partes de módulos. Após, eles serão integrados num único protótipo de AVCEA baseado na arquitetura e nos estudos expostos previamente neste trabalho.

5.2.1 Ferramentas para Sistemas de Aprendizagem

Como descrito na Seção 2.7.2, sistemas de aprendizagem via *Web* são compostos comumente por conjuntos de ferramentas que compõem os AVAs e que também podem acumular funções de gerenciadores de cursos (*CMSs*), gerenciadores de objetos de aprendizagem (*LOMs*), sistema instrucional de ensino-aprendizagem (*ILSs*), entre outras. Há na atualidade uma gama muito grande de implementações de AVAs disponíveis (vide Seção 3.3). Muitos destes ainda possuem licenças livres e são de código aberto o que viabiliza a implementação de sistemas de aprendizagem adaptados para as mais diversas instituições.

O Ministério da Educação brasileiro (MEC) adota e indica como sistema de aprendizagem o *MOODLE*, pelo que este possibilita - maior reusabilidade e flexibilidade, além de possuir uma extensa rede de colaboradores/desenvolvedores. Estima-se que este AVA esteja disponível para estudantes de mais de 50.000 instituições pelo mundo. O alinhamento de teoria educacional (construtivismo social) e a adoção deste AVA no consórcio *RExNet* foi de grande valia para o seguimento das implementações deste AVA no âmbito da arquitetura proposta.

O *MOODLE*, como quase todos AVAs analisados, possui ligação com sistemas de banco de dados diversos o que é vantajoso e está de acordo com a arquitetura. Dentre os sistemas de banco de dados compatíveis estão: *PostgreSQL* (POSTGRESQL, 2011), *FOSS*, desenvolvido usando a linguagem de programação *C* e multiplataforma; *MySQL* (MYSQL, 2011), também *FOSS* desenvolvido usando a linguagem de programação *C* e *C++* e multiplataforma, de ampla utilização; *Microsoft SQL Server - MSSQL* (MICROSOFT, 2011b), desenvolvido pela *Microsoft* em *C*, *C++* e *C#* e, por consequência, exclusivo das plataformas *Windows* e de licença proprietária; e *Oracle Database* (ORACLE, 2011b), da empresa *Oracle* escrito em *C* e *C++*, multiplataforma, vastamente utilizado em grandes corporações onde a segurança de dados é extremamente valiosa, também de licença proprietária. Entre estes gerenciadores ou sistemas de banco de dados utilizaremos o *MySQL* por experiências prévias positivas, mas isso não exclui a possibilidade de uso do *PostgreSQL* como alternativa futura.

As informações de usuários, conteúdos, *logs* e todas as interações são armazenadas pelo *MOODLE* em um único banco de dados em diversas tabelas (aproximadamente 198). A organização das informações no banco de dados do *MOODLE* é intrincadamente complexa, mas disposta em alguns grandes grupos:

- configuração - dados de configuração do *MOODLE* e de *plug-ins*;
- usuários e perfis - todos os dados de usuários e perfis;
- tipos de usuários e suas delimitações - informações de papéis (*roles*) de usuários e suas “permissões” (*capabilities*);
- cursos - informações da organização dos módulos dos cursos;
- grupos e agrupamentos - informações de conjunto de usuários;
- *logs* - todas as informações de interações com o AVA;
- eventos - informações de possíveis eventos ligados a datas, etc.;
- estatísticas - geração de estatísticas baseada na tabela de *logs* de usuários;
- notas (*gradebook*) - informações de avaliações de estudantes;
- banco de questões - informações de questionários;
- sistema de mensagem - informação de fóruns, *chats*, etc.;
- rede - informações de serviços de rede;
- atividades - informações de atividades, tarefas, etc.;

- blocos - informações de instâncias de blocos personalizados.

A Fig. 51 ilustra a página inicial típica, com alguns elementos personalizados, do *MOODLE* usado pelo GCAR (GCAR-EAD, 2011), à esquerda, e também no servidor geral da UFRGS (UFRGS-MOODLE, 2011), à direita.

Figura 51: *MOODLE* - sistemas de aprendizado em uso na UFRGS.

A inclusão de funcionalidades novas é facilitada por uma ferramenta de gerenciamento de módulos. Também é bastante simples o desenvolvimento de módulos personalizados, já que o código é aberto e a arquitetura do *MOODLE* é modular e orientada a objetos, o que facilita inclusive a instanciação de diversos módulos. O código *PHP*, apesar de simples, é de grande utilidade e de fácil edição. O *MOODLE* conta também com uma interface de comunicação *XML-RPC* para gerenciamento remoto bastando para isso conhecer os comandos e habilitar esta funcionalidade.

5.2.2 Ferramentas para Componentes Intercambiáveis

Em um trabalho anterior do autor, a estratégia de componentes intercambiáveis foi implementada usando o *framework* de comunicação *OPC-DA* (*OLE for Process Control - Data Access*) (SCHAF, 2006; OPC Foundation, 2011). Todavia esta tecnologia de comunicação se mostrou muito seletista, pois é baseada em interfaces *OLE* (*Object Linking and Embedding*), *COM* (*Component Object Model*) e *DCOM* (*Distributed COM*) sob plataformas *Windows*. Embora atualmente este cenário tenha se modificado com a proposição de um novo modelo *OPC*, o *OPC-UA* (*Unified Architecture*), que desvincula a necessidade do uso de plataformas *Windows* e pode ser implementado em linguagens *Java*, *.NET* ou *C*, o desenvolvimento ainda não segue o modelo concretizado e a comunidade desenvolvedora ainda está adaptando o novo modelo.

Embora no meio industrial o *OPC* seja bastante utilizado, no meio educacional aplicações são ligadas a ferramentas comerciais, o que restringe a interoperabilidade e a integração a novos ambientes. Por este motivo uma nova estrutura para componentes intercambiáveis é proposta baseada em tecnologias abertas e comunicação via *Web* no estilo de *WSs*, o *XML-RPC*. Com este uma simples chamada de funções soluciona grandes problemas de interoperabilidade e de dependências de plataformas. O padrão de comunicação entre componentes está ilustrado na Fig. 52. O gerenciador é responsável pela criação do cenário de ensaio com os componentes escolhidos. Cada componente inicia um servidor *XML-RPC* que aguarda conexões de clientes. O gerenciador inicia a conexão criando um cliente *XML-RPC* de acordo com informações armazenadas num repositório de informações de componentes intercambiáveis. Este repositório também contém dados dos elementos presentes em cada componente: sensores e atuadores nas plantas e tipos de controle dos controladores. Desta maneira é possível direcionar corretamente as entradas e saídas (*I/Os*) de cada componente para se obter o cenário desejado. Naturalmente, o gerenciador pode iniciar tantos clientes quantos forem desejados para unir qualquer dos componentes que podem, inclusive, ser formados por mais de uma planta ou controlador.



Figura 52: Implementação dos componentes intercambiáveis - diagrama de comunicação.

5.2.3 Ferramentas para Sistemas Multiagente

SMA's podem ser empregados nas mais diversas áreas e para cada área há uma topologia ou *framework* de agentes adequado (STONE; VELOSO, 2000). Há atualmente uma vasta gama de *frameworks* de desenvolvimento para SMA's. Inicialmente é necessário o estabelecimento do tipo de agentes: reflexivos (reativos) ou cognitivos. Agentes cognitivos são baseados em organizações sociais humanas como grupos, hierarquias e mercados. Estes possuem memória (histórico) de ações e podem planejar ações futuras. Agentes reflexivos são baseados em modelos de organização biológica ou etológica¹. O modelo de funcionamento de um agente reativo é formado pelo par estímulo-resposta (ação-reação) e não há representação explícita do conhecimento do agente. Este tipo é adequado para a busca de informações específicas em ambientes computacionais. Nota-se a diferença de complexidade dos dois, e por este motivo inicialmente o projeto de um SMA para aplicação em tutores autônomos e também para adaptação de conteúdo se fundamentará em agentes reflexivos.

Os *frameworks* de desenvolvimento de SMA's analisados foram:

¹etologia é o estudo do comportamento social e individual.

- *Jason* (JASON, 2011) - licença *GNU LGPL*, *framework* escrito em *Java* para desenvolvimento de *SMA*s com agentes *BDI* (*Belief-Desire-Intention*) utilizando a linguagem de descrição *AgentSpeak(L)*;
- *Cougaar* (COUGAAR, 2011) (*Cognitive Agent Architecture*) - *FOSS*, baseado totalmente em *Java*;
- *JADE* (*Java Agent DEvelopment framework*) (JADE, 2011) - *FOSS*, para agentes reativos descritos em linguagem de programação *Java*;
- *SemanticAgent* (SEMANTICAGENT, 2011) - baseado no *JADE* com extensão para linguagem de regras de semântica da *Web* (*SWRL - Semantic Web Rule Language*);
- *SeSAM* (SESAM, 2011) (*Shell for Simulated Agent Systems*) - *LGPL*, ambiente para simulação de sistemas multiagente reativos descritos em linguagem declarativa própria (*DECL*);

Para agentes reativos tem-se três opções sendo que uma delas é amplamente utilizada e, inclusive, com padronização internacional. O *JADE* usa um protocolo padronizado pela *FIPA* (*Foundation for Intelligent Physical Agents*) - uma organização da *IEEE* que propõe interoperabilidade entre tecnologias baseadas em agentes - o *ACL* (*Agent Communications Language*). Na Fig. 53 está ilustrada a interface gráfica (*GUI*) do “ambiente” *JADE* com alguns agentes desenvolvidos em operação no momento (à esquerda) e uma ferramenta de visualização de troca de mensagens entre agentes, um *sniffer* (à direita).

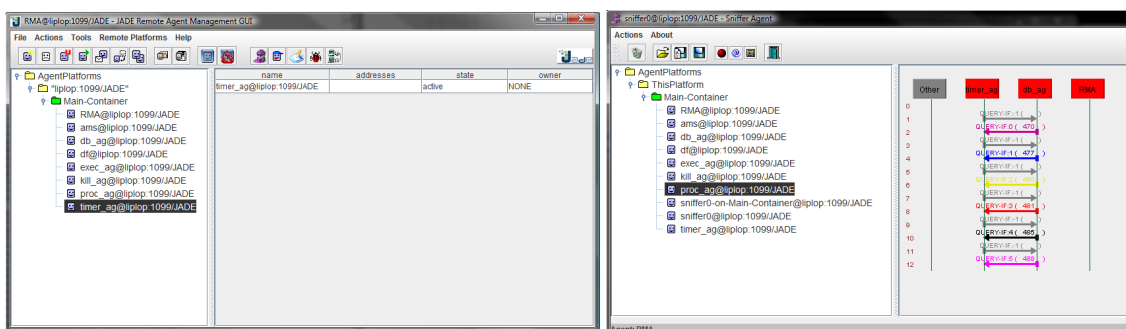


Figura 53: *JADE* - *GUI* e funcionalidade *sniffer*.

O *JADE* engloba o caráter de distribuição de agentes em sua arquitetura, sendo considerado cada nó distribuído um *container*. O *container* principal (mestre) possui uma base de três agentes: o *RMA* (*Remote Monitoring Agent*), responsável pela monitoria de *containers* distribuídos; o *DF* (*Directory Facilitator*), que funciona como páginas amarelas para o descobrimento de serviços de agentes; e o *AMS* (*Agent Management System*) que gerencia todos os agentes. A Fig. 54 ilustra a arquitetura de comunicação do *JADE*. Nota-se que a comunicação entre os *containers* utiliza o *Java RMI* (*Remote Invocation Method*) bastante seguro e de ampla utilização em sistemas de automação.

Por ser de implementação aberta e baseado em *Java*, com padronização específica para garantir interoperabilidade, o *JADE* foi escolhido como ferramenta de desenvolvimento na implementação de estudos de caso com *SMA*s.

5.2.4 Ferramentas para Adaptação de Conteúdo

Para adaptação de conteúdo (vide 4.4.4) é necessário o desenvolvimento de um sistema que acompanhe as interações dos estudantes no (ou com o) *AVCEA*. Como previamente idealizado na arquitetura proposta, *SMA*s são adequados para este acompanhamento. Logo, é necessário o desenvolvimento de agentes-monitores de interações usando

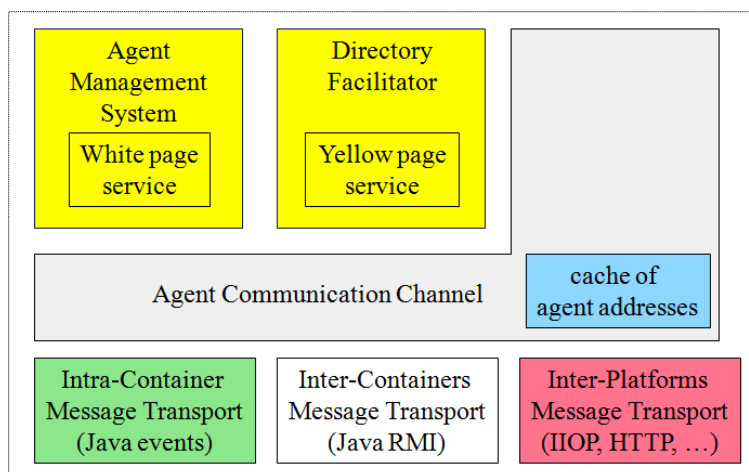


Figura 54: *JADE* - arquitetura de comunicação básica.

o *JADE*, que foi escolhido como *framework* para o desenvolvimento de agentes. As interações, por sua vez, estão/são armazenadas durante a navegação do usuário pelo AVCEA. Inicialmente, estabeleceu-se que o *MOODLE* terá a funcionalidade de AVC e, assim, todas as informações necessárias de interações podem ser acessadas na tabela *logs* (vide Seção 5.2.1) do banco de dados (*MySQL*).

Não existe ferramenta específica para o desenvolvimento de programas computacionais de adaptação de conteúdo, assim, uma associação de agentes *JADE* com o banco de dados *MySQL* e páginas *PHP* servem como ferramentas para a implementação dessa “tarefa”. Posteriormente, a implementação de uma solução para este tema será apresentada em um estudo de caso.

A modelagem de agentes segue os padrões do *JADE*, embora SMAs baseados em agentes reativos geralmente possuam comportamentos (*behaviour*) também padronizados. Entre os comportamentos de agentes podem-se citar: i. categoria de comportamentos simples que incluem: a. comportamento atômico (*OneShot*) e b. comportamento cíclico (*Cyclic*); ii. categoria de comportamentos compostos com: a. máquina de estados finitos (*FSM*), b. sequencial (*sequential*) e c. paralelo (*parallel*). Adicionalmente, ainda existem classes para mais dois comportamentos no *JADE*, são eles: comportamento despertador (*Waker*) e comportamento periódico (*Ticker*). Não é difícil imaginar que para tarefas periódicas o comportamento *Ticker* é o mais adequado e, para tarefas que não envolvam sequenciamento, mas que não sejam atômicas, o comportamento cíclico (aperiódico) é adequado. Com base nestes, pode-se projetar toda a funcionalidade do SMA para adaptação de conteúdo como idealizado no módulo de adaptação da arquitetura proposta.

O sistema de aprendizado *MOODLE* é naturalmente composto por diversos módulos e pode ser enriquecido com recursos externos. A adaptação de conteúdo é natural em *PHP* que são executados pelo servidor e a resposta (quase sempre um simples *HTML*) é enviada ao usuário que requisitou o conteúdo. A adaptação de conteúdo educacional é inerente neste ponto, já que *PHPs* também serão usados para materiais didáticos, *LOs*, etc. e o universo de conteúdos está armazenado no banco de dados *MySQL*.

De maneira a incrementar as regras de adaptação do agente-adaptador do *JADE*, ele pode ainda ser dotado de funcionalidade do *JESS* (*JESS*, 2011), que é máquina de regras também baseada em *Java*.

5.2.5 Ferramentas para Tutoriamento

Da mesma maneira que a adaptação de conteúdo o tutoriamento será efetuado por agentes modelados no *JADE*. Os agentes-monitores também efetuam o trabalho de coleta de informações no banco de dados, e o refinamento destas informações é efetuado por um agente-tutor (personalizado para cada usuário) disparado pelo monitor-mestre. A programação do agente-tutor, todavia, é muito mais complexa que do que do agente-adaptador (de conteúdo), uma vez que este deve também interagir com o usuário. Através de opções oferecidas ao usuário, o tutor remodela respostas e infere diretamente sugerindo material educacional adicional ou alternativo para solucionar problemas de aprendizagem.

O tutoriamento exige extensa programação, modelagem de mapas conceituais, modelos de aprendizado de estudantes, etc. (MAGOULAS; PAPANIKOLAOU; GRIGORIDOU, 2003). No capítulo estado da arte (vide Seção 3.8), foram apresentadas algumas implementações simples de *ITSs*. Esta linha de pesquisa ainda gera muito debate sobre efeitos pedagógicos da automatização do tutoriamento de estudantes.

Análogo à seção anterior, ferramentas como agentes *JADE*, banco de dados *MySQL* e páginas *PHP* são empregadas para implementações simples de tutores autônomos baseados no histórico de interações com o AVA *MOODLE*. Redes neurais, lógica *fuzzy* e redes probabilísticas (bayesianas) também podem incrementar o mapeamento da aprendizagem dos estudantes.

5.2.6 Ferramentas para Suporte à Colaboração

Uma prévia de ferramentas colaborativas foi apresentada na Seção 2.4 com inclusive uma categorização entre elas. Diversas “tecnologias” foram citadas conforme a complexidade e a finalidade em três grupos: ferramentas de comunicação, de conferência e de gerenciamento de colaboração. Normalmente, os AVCEAs proporcionam a maioria das funcionalidades abrangidas, com exceção das que envolvem áudio- e vídeoconferência, sistemas de reuniões eletrônicas e compartilhamento de aplicações, pois normalmente essas funcionalidades estão direcionadas para reuniões virtuais que normalmente não possuem caráter educacional e sim comercial/profissional. Ainda assim, tais funcionalidades também podem ser integradas ao projeto de AVCEAs para o transmissão de aulas virtuais com *softwares* como o *Adobe Connect*, o *OpenMeetings*, o *BigBlueButton* ou até o *Skype* (INC., 2011) (vide Seção 3.11).

A integração de ferramentas de colaboração em AVCEAs entretanto não garante a colaboração entre usuários. A elaboração de agentes cientes de colaboração (*collaboration aware agents*) eleva a complexidade de tutores baseados em simples interações com o AVA, pois neles a relação usuário-usuário também é relevante e necessária. Assim, tabelas do banco de dados que contenham informações de comunicação entre usuários também estão sujeitas à *data mining*. O grupo que contém tabelas de sistemas de mensagens, como *forum posts*, *forum discussions*, *chat messages*, etc. é sujeito de análise. Para isso também é necessário o conhecimento dos grupos de estudantes (tabela de agrupamentos), tabela de cursos, tabela de usuários e perfis, tipos de usuários, etc.

Logo, não é difícil perceber que o suporte à colaboração é tarefa muito complexa e exige uma coleta e tratamento intensos de informações de usuários. Uma implementação simples ainda utiliza agentes de colaboração modelados no *JADE*, banco de dados *MySQL* e páginas *PHP*. Uma semântica relacional também é necessária para obtenção de correlações entre os dados e, conseqüentemente, uma avaliação de pesos para tomadas de decisão da solução ótima de suporte à colaboração.

5.2.7 Interfaces Sociais Tridimensionais e Ambientes Imersivos

Implementações de ferramentas que suprem interface social tridimensional foram amplamente apresentadas no capítulo de estado da arte de ambientes imersivos (vide Seção 3.5). Contando somente as implementações *FOSS* tem-se: o *Open Wonderland*, o *OpenSim*, o *ScienceSim*, o *RealXtend* e o *VastPark*. A maior comunidade desenvolvedora, entretanto, ainda se concentra no *Second Life* (vide Seção 3.5.3 e Seções 3.5.3.2 e 3.5.3.3 para aplicações no ensino), que é uma implementação comercial. Visto que o *OpenSim* serve de base para o *ScienceSim* e para o *RealXtend* e tendo em vista a grande compatibilidade e semelhança com o *Second Life*, adotamos o *OpenSim* como ferramenta de desenvolvimento para a interface social 3D.

Ainda assim, o *VastPark* pode assumir um papel maior no futuro, pois atualmente ainda está evoluindo. O *Open Wonderland* (vide Seção 3.5.2 e Seção 3.5.2.1 para aplicação no ensino) poderia ser, naturalmente, a opção mais simples, pois se baseia em tecnologias *Java*, mas seu futuro ainda é incerto e a comunidade desenvolvedora é menor do que a do *OpenSim*.

O *ScienceSim* oferece materiais educacionais para serem integrados no *OpenSim* voltados às ciências naturais. O *RealXtend* (vide Seção 3.5.5), por sua vez, estende funcionalidades do *OpenSim* e está em pleno desenvolvimento, contando com grandes avanços, tanto na criação de servidores mais robustos e confiáveis quanto no desenvolvimento de clientes mais leves e que possam ser usados em dispositivos com menos recursos computacionais como *PDA*s e *smartphones*.

Como descrito o *OpenSim* (vide Seção 3.5.4) é baseado na linguagem de programação *C#* com diversos protocolos de comunicação para integração com *plug-ins* e módulos externos. O *OpenSim*, naturalmente, suporta o banco de dados *MySQL*, que será usado como repositório, e o protocolo de comunicação *XML-RPC*, na qual a maioria de sua estrutura de conexão está projetada. Interações no *metaverso* não são, naturalmente registradas (previamente programadas para serem registradas), mas esta funcionalidade é facilmente implementada usando *scripts* do próprio *OpenSim* ou *Linden scripts*, já que estes são em quase sua totalidade compatíveis.

A personalização ou customização e desenvolvimento de um mundo no servidor do *OpenSim* é bastante simples e proporciona grandes vantagens de visualização de material educacional uma vez que objetos virtuais serão criados à imagem e modelo de elementos reais. Há, assim, a criação de uma expansão da realidade no mundo virtual.

Devido a compatibilidade da maioria das funções do *Second Life* com o *OpenSim*, o pacote *SLOODLE* (vide Seção 3.5.3.1) pode ser utilizado para ligação com o *MOODLE*. Isso torna possível a visualização de elementos do *MOODLE* como *LO*s e elementos dos cursos, como tarefas e outros, no *metaverso* do *OpenSim*.

Através do *framework Mono* pode-se executar o *OpenSim* também em *SOs Unix*, o que não gera dependências de sistemas específicos. Existe uma infinidade de clientes para conexão/visualização de mundos hospedados no servidor *OpenSim*. Adotaremos o *Hippo Viewer* (HIPPO, 2011) como cliente padrão por se encaixar nos moldes de *FOSS* e por proporcionar todas as funcionalidades desejadas, como interoperabilidade e independência de plataforma. O *Hippo* também pode ser usado para conexão com o servidor *Second Life* e possui uma ferramenta integrada de seleção de servidores.

Há também um conjunto de bibliotecas para desenvolvimento para interações com mundos 3D virtuais, a *LibOpenMetaverse* compatível com protocolo do *Second Life* e do *OpenSim*. Nela podem ser desenvolvidas aplicações para clientes e autômatos (*bots*), ou seja, podem-se “automatizar” *avatares* para efetuarem interações no *metaverso* de acordo

com a programação desenvolvida. A biblioteca está escrita em *C#* e é compatível com o *framework .Net* e com o *Mono*.

5.3 Estudos de Caso

Várias implementações de diferentes módulos produziram estudos de caso suficientes para argumentação e construção de um protótipo. A seguir seis estudos de caso serão descritos que ilustram aspectos presentes na maioria dos módulos idealizados na arquitetura para AVCEAs.

5.3.1 Monitorador de Agendamento

Um primeiro estudo de caso confirmou a possibilidade do uso de SMAs projetados no *JADE* para o desenvolvimento de monitores de informações. Neste teste foi usado um agente, chamado de *db_mon_ag*, capaz de efetuar leituras no banco de dados, que contém tabelas para o armazenamento de informações de um experimento remoto utilizado na disciplina de Sistemas de Controle I, da Engenharia Elétrica da UFRGS. O sistema de agendamento (*booking system*) foi desenvolvido por haver somente uma planta na qual somente um estudante pode fazer ensaios por vez. Ele foi programado em *PHP* e registra seus agendamentos numa tabela chamada *agenda*, no banco de dados *MySQL* do experimento, onde informações do *login* (identificador do usuário), horário e nome são armazenadas. O horário, por sua vez, é composto de dia (d), mês (m), ano (a) e horário do dia (h) no formato 24h, cada um composto por dois dígitos com exceção do ano que tem 4 dígitos, ou seja, *ddmmaaaahh*. O sistema de agendamento está integrado totalmente na instalação do AVA *MOODLE* usado na Engenharia Elétrica, chamado de GCAR-EAD, como pode ser visto na Fig. 55.

O *MOODLE* aceita repasse de informações para recursos externos que neste caso é o agendador. Os dados do usuário transmitidos como parâmetros são o nome, *id* (*login*) e o *e-mail*. Assim o agendador reconhece cada usuário do *MOODLE* e pode efetuar o agendamento no experimento. Nota-se também que o AVA serve como controle de acesso de usuários aos experimentos.

O experimento é uma planta térmica, na qual estudantes devem controlar a temperatura da resistência, que é aquecida pela circulação de corrente elétrica através do circuito. A corrente elétrica é “manipulada” pelo controlador industrial *PID* (modelo N1100 da empresa brasileira *Novus* (NOVUS, 2011)) através de *PWM* da tensão, com valores de 0 a 100% do *duty cycle*. A medição de temperatura é efetuada por um termopar tipo K próximo à resistência que está ligado ao controlador *PID* na entrada universal multisensor. Este experimento está melhor descrito em um trabalho anterior do autor, onde, inclusive, foi implementada a estratégia de componentes intercambiáveis (SCHAF, 2006).

Para evitar problemas com o gerenciador de experimentos (vide Seção 2.8), que foi implementado usando o *software Elipse SCADA* - desenvolvido por uma empresa brasileira com a qual a UFRGS possui algumas licenças devido a um convênio -, ele somente é iniciado no servidor quando há horário agendado por usuários. Isso evita a alocação demasiada de recursos e o aumento excessivo de memória de paginação (problemas intrínsecos do SO *Windows*) que podem causar erros no servidor e inviabilizar a automação do experimento até que um tutor reinicie o processo manualmente no servidor. Assim, para automatizar este processo de inicialização do gerenciador, um agente, chamado *exec_ag*, foi criado e outro agente, chamado *kill_ag*, para finalização do gerenciador. Outro agente, chamado de *proc_ag*, monitora os processos que estão em execução



Grupo de Controle Automação e Robótica

GCAR-EAD > Exp. Rem. > Recursos > Agendar ou utilizar experimento remoto - Planta Térmica

Agendamento de Experimentos

Experimentos disponíveis:

Controle Manual	Controle Automático
Malha Aberta - Variação de Potência Resposta em Frequência	Malha Fechada - PID

Recurso livre para a hora corrente!
Agendar Agora

Agendar experimento para dia:
01 / 01 às 00 horas Agendar

Hora atual do servidor: 04/01/11 18:34

Recursos Agendados:

<< January 2011 >>						
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29
30	31					

Recursos Agendados para o dia 04/01:
13h - 13:59 - Frederico
15h - 15:59 - Teste

O tempo de alocação é de no mínimo uma hora.

Figura 55: Sistema de agendamento - interface integrada no *MOODLE*.

no servidor.

Um esquema do funcionamento do “monitorador de agendamento” está ilustrado na Fig. 56, que é semelhante ao proposto no Fig. 44 para adaptação de conteúdo da arquitetura modular de AVCEAs (vide Seção 4.4.4). Neste diagrama estão presentes o *MOODLE*, como AVA, o gerenciador de agendamentos, escrito em *PHP* e integrado ao *MOODLE* (como ilustrado na Fig. 55), o banco de dados, usando o gerenciador *MySQL*, com a tabela agenda, o gerenciador do experimento, supervisor implementado no *software Eclipse SCADA*, e os vários agentes do SMA, que desempenham o papel principal do “sistema monitorador de agendamentos”.

O agente que tem a funcionalidade principal neste estudo de caso é o `db_mon_ag`. Parte do código de implementação desse agente está descrito logo abaixo. Para a leitura do banco de dados ele emprega a biblioteca *JDBC (Java Database Connectivity)* para usar um *driver* para o banco de dados *MySQL*, chamado de *MySQL connector for Java*, como pode ser visto na linha 12. Esta biblioteca é livre e suporta *drivers* de diversos bancos de dados diferentes. Não é difícil de perceber que o agente que monitora a tabela do banco de dados do experimento, `db_mon_ag`, executa uma tarefa periódica, implementada usando o comportamento `Ticker`, de leitura (linha 5). Ele faz uma busca (*query*) na tabela procurando o valor do campo `hora` para o horário corrente (linha 14). Se houver uma ocorrência positiva (linha 25), ele envia uma mensagem ao `proc_ag` (linha 28 e 29). O comportamento dele é iniciado no momento de sua inicialização, `setup()` (linha

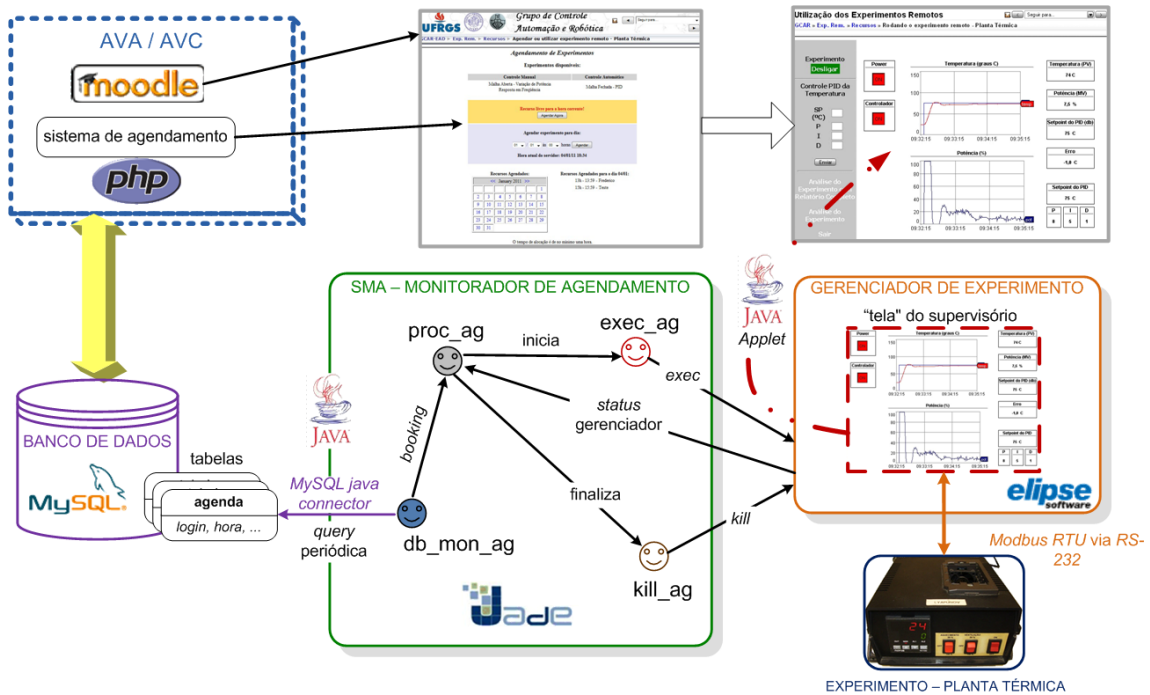


Figura 56: Estudo de caso: monitorador de agendamento de experimento - diagrama de implementação.

4), e a cada período de 500 ms (linha 5) o método `OnTick()` é executado (linha 6). A mensagem padronizada `ACL` do `db_mon_ag` ao `proc_ag` é enviada com a “performativa” de informação, `INFORM`, na linha 26. Obviamente que bibliotecas do `JADE` e do `SQL` devem ser carregadas para a execução desse código (linhas 1 e 2).

```

booking_mon_ag.java (db_mon_ag)
1  import java.sql.*;
2  import jade.core.*;
3  public class booking_mon_ag extends Agent {
4      protected void setup() {
5          loop = new TickerBehaviour(this, 500) {
6              protected void onTick() {
7                  ACLMessage msg = null;
8                  System.out.println("Verifying Status on " + now("ddMMyyyyHH"));
9                  String result = null; Connection conn = null;
10                 String url = "jdbc:mysql://localhost:3306/";
11                 String dbName = "experimento";
12                 String driver = "com.mysql.jdbc.Driver";
13                 String userName = "*****"; String password = "*****";
14                 String query = "SELECT login FROM agenda WHERE hora=" + now("ddMMyyyyHH") + "****";
15                 Statement stmt; ResultSet rs;
16
17                 try {
18                     Class.forName(driver).newInstance();
19                     conn = DriverManager.getConnection(url+dbName,userName,password);
20                     stmt = conn.createStatement();
21                     rs = stmt.executeQuery(query);
22                     while(rs.next()) { result = "booked to " + rs.getString("login"); }
23                     conn.close();
24                 }
25                 catch (Exception e) { e.printStackTrace(); }
26
27                 if (result != null) {
28                     msg = new ACLMessage(ACLMessage.INFORM);
29                     msg.setContent(result);
30                     msg.addReceiver(new AID("proc_ag", AID.ISLOCALNAME));
31                     send(msg);

```

```

30         }
31         else { System.out.println("no booking"); }
32     }
33 };
34     addBehaviour(loop);
35 }
36 }

```

O agente `proc_ag` está modelado com o comportamento cíclico que verifica mensagens com a “performativa” de informação. Caso ele receba uma mensagem de agendamento este executa um comando de informações sobre tarefas, buscando a execução do gerenciador (*Eclipse SCADA*). Caso o processo não esteja ativo ele envia uma mensagem ao `exec_ag` que é responsável pela inicialização do gerenciador. Após passado o horário de agendamento o `db_mon_ag` pode também informar o `proc_ag` sobre a vacância do horário e assim este pode também determinar a finalização do gerenciador enviando uma mensagem ao `kill_ag`.

Os agentes ligados à inicialização, finalização e informações de processos empregam a classe `Runtime` da biblioteca padrão do *Java* (`java.lang.Runtime`), como o método `exec`, que tem como parâmetro único o a linha de comando (arquivo executável e possivelmente os respectivos parâmetros de entrada deste) que se deseja executar. Desta maneira qualquer aplicativo executável pode ser iniciado como se fosse da linha de comando (`shell` no *Linux* e o equivalente `cmd` no *Windows*). Por exemplo, basta substituir o comando na linha

`Process p = Runtime.getRuntime().exec("comando")`, para:

- `"caminho"/termica.app`
para inicializar o gerenciador do experimento, sendo o arquivo `termica.app` o supervisor do *Eclipse SCADA* que gerencia a planta térmica;
- `tasklist /fi "IMAGENAME eq elipse32.exe" /NH`
para obter informações do gerenciador. Este retorna o identificador (id/número) do processo;
- `taskkill /t /im elipse32.exe -F`
para finalizar o gerenciador.

Assim, o SMA usado para automatizar o processo de agendamento e inicialização do gerenciador de experimentos pode ser visualizado na Fig. 57, onde, à esquerda, está representado o *GUI* do *JADE* com os agentes listados e a troca de mensagens de agentes após um agendamento de um usuário capturado pela ferramenta *sniffer* e, à direita, o detalhamento da mensagem *ACL* do `proc_ag` ao `db_mon_ag` (puramente para depuração) indicando que o processo (no caso o gerenciador, ou seja, o `Elipse32.exe`), está ativo com o identificador de número 1048 e consumindo 20.592 Kbytes de memória.

5.3.2 Metaversos como AVCEAs

Usando o *software OpenSim*, foi criado um ambiente colaborativo social baseado em mundos tridimensionais, isto é, um *metaverso* chamado de *TestLab3D*. O estudo de caso foi implementado no modo *stand-alone* do *OpenSim* associado com o sistema de banco de dados *MySQL*. A Fig. 58 expõe o *metaverso* criado, com um *avatar* personalizado de um usuário. Adicionalmente ao terreno virtual foram criados metaobjetos, alguns experimentos virtuais (simulações), painéis de visualização, materiais educacionais e outros elementos usados a fim de familiarizar os usuários com um ambiente de ensino-aprendizagem.

O *metaverso* criado (chamado desenvolvedores do *OpenSim* de *simulação*) é simples, mas expõe a possível funcionalidade desta ferramenta que pode ser usada como interface

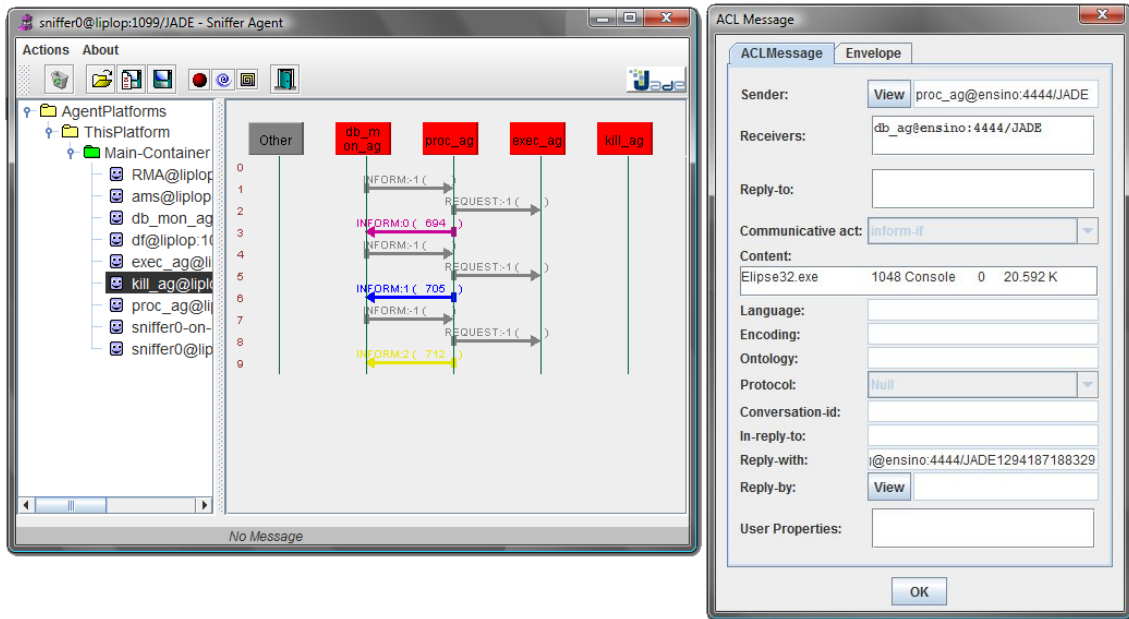


Figura 57: Estudo de caso: monitorador de agendamento de experimento - GUI do JADE e interação de agentes.



Figura 58: Estudo de caso: interface social 3D.

para acesso a AVCEAs tradicionais ou incorporar as funcionalidades destes. A criação de objetos é árdua, pois exige muito tempo de desenvolvimento. Para o *metaverso* do estudo de caso foram gastos aproximadamente 100 horas (homens-hora) de desenvolvimento. Para se adicionar interatividade a elementos (metaobjetos) é necessária a programação de *scripts* associados aos respectivos metaobjetos. A Fig. 59 ilustra a ferramenta de edição de objetos, alguns atributos, a associação com *scripts* (*Linden Scripts*) e um exemplo de *script* no qual, a partir do toque (*touch*) do usuário, uma mensagem de texto é publicada

para todos os usuários presentes num raio prédefinido de distância do objeto. O cliente *Hippo Viewer* foi empregado para visualização do *TestLab3D*.

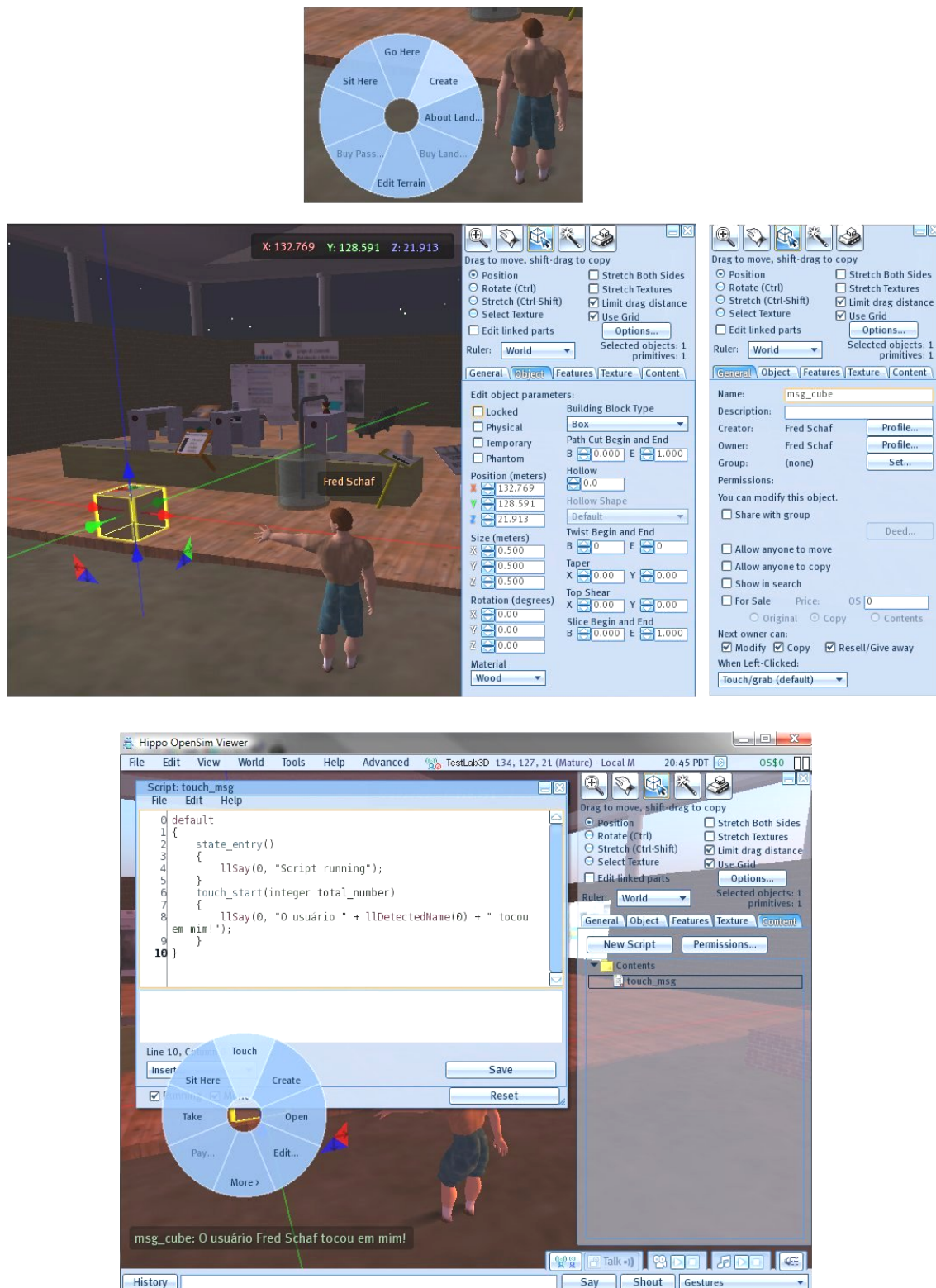


Figura 59: *OpenSim* - ferramenta de edição de objetos do *metaverso*.

O *OpenSim* é compatível com *Linden Scripts*, do *SL*. Há uma grande comunidade que divulga tutoriais e exemplos de *scripts* na *Web*. O *Second Life Wiki* (SLWIKI, 2011) pos-

sui uma vasta gama de exemplos de código-livres, os quais são mantidos pelos próprios usuários (residentes) do *SL* que desejarem colaborar.

O pacote *SLOODLE* (vide Seção 3.5.3.1) foi testado para possibilitar uma ligação com cursos já desenvolvidos no GCAR-EAD. A compatibilidade do *SLOODLE* com o *OpenSim* ainda não é total e algumas mensagens de depuração do *script* (*warnings*), do pacote *SLOODLE*, são mostradas no *metaverso* devido à falta de implementação do método `llGetNextEmail`. A função deste método no código original do pacote *SLOODLE* é receber notificação automática de atualização do conteúdo (hospedado no *MOODLE*), o que pode ser implementado alternativamente por um verificador periódico, que foi implementado. Apesar deste aparente problema, um *link* com uma instalação simples do *MOODLE* foi feita, e alguns *slides* foram visualizados usando o módulo *presenter* do *SLOODLE*, como mostra a Fig. 60. Diversos recursos do *MOODLE* podem, assim, ser acessados e visualizados no *OpenSim*.

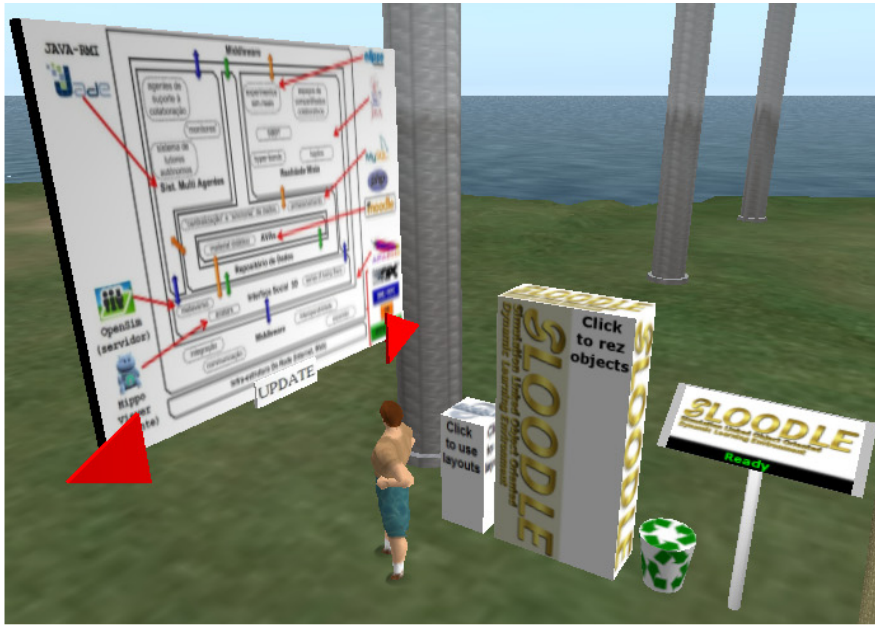
Um teste do *software RealXtend* também foi conduzido e apresentou praticamente os mesmos resultados, mas com algumas vantagens. *Avatares* do *RealXtend* possuem mais realismo, porque é possível, inclusive, a utilização de ferramentas de geração de modelos de rosto através da captura de imagens do usuário. Porém, talvez a maior vantagem do *RealXtend* em relação ao *OpenSim* é a capacidade de importação de modelos tridimensionais no formato *mesh*, o que pode encurtar e facilitar o projeto de objetos virtuais em ferramentas especializadas e com maiores recursos. Mundos desenvolvidos no *OpenSim* podem ser completamente importados para o *RealXtend* sem perdas, já o caminho inverso implica em perdas de funcionalidades, pois o *RealXtend* estende algumas funcionalidade de objetos virtuais que não são interpretados corretamente por visualizadores do *OpenSim* como o *Hippo*.

5.3.3 *Hardware-in-the-Loop* de Baixo Custo para Testes de Realidade Mista

Este estudo de caso testou a operação de controladores reais que automatizam processos simulados em uma planta virtual. Primeiramente, foi desenvolvido uma planta de envase virtual modelada no *OpenSim* com o auxílio de *scripts* para a programação do comportamento do processo². A planta é constituída de três estações de trabalho, cada uma responsável por uma etapa do processo de envase de garrafas. A Fig. 61 ilustra a modelagem virtual tridimensional da planta, seu painel de controle e os sensores dispostos como sinalizadores vermelhos em cada estação no *RealXtend Viewer*. A primeira estação, de enchimento, é composta de uma válvula, que, quando aberta, deixa escoar o líquido usado no enchimento das garrafas; um reservatório com um sensor de nível, onde o líquido é armazenado; e um sensor que detecta a posição da garrafa quando esta se encontra exatamente abaixo da válvula (na estação de enchimento). A segunda estação, de lacre, é composta de um braço mecânico, que, quando acionado, executa a tarefa tarefa de colocação do lacre (tampa); e um sensor que detecta a posição da garrafa quando esta se encontra exatamente abaixo do braço mecânico (na estação de lacre). A terceira e última estação, de rotulagem, é composta de uma rotuladora de impressão, que, quando acionada, executa a tarefa de rotulagem (imprime); e um sensor que detecta a posição da garrafa quando esta se encontra exatamente à frente da rotuladora. A esteira é o autador que transporta a garrafa através das estações de trabalho.

De forma a automatizar o processo, os usuários devem, com base nas informações dos sensores, acionar os atuadores. Este modelo, na sua versão anterior modelado no

²baseado em um trabalho anterior do autor desenvolvido durante estágio de docência na disciplina de Sistemas de Automação da Engenharia Elétrica da UFRGS (SCHAF, 2006).



curso teste

Moodle_MR > teste > SLOODLE Modules > Presenter: test_presenter

Presenter: test_presenter

Module Type: Presenter

View Edit Add slides Upload Many Import Slides

"prot impl"

1 2 3 4

Figura 60: Pacote SLOODLE - teste de uso do no OpenSim.

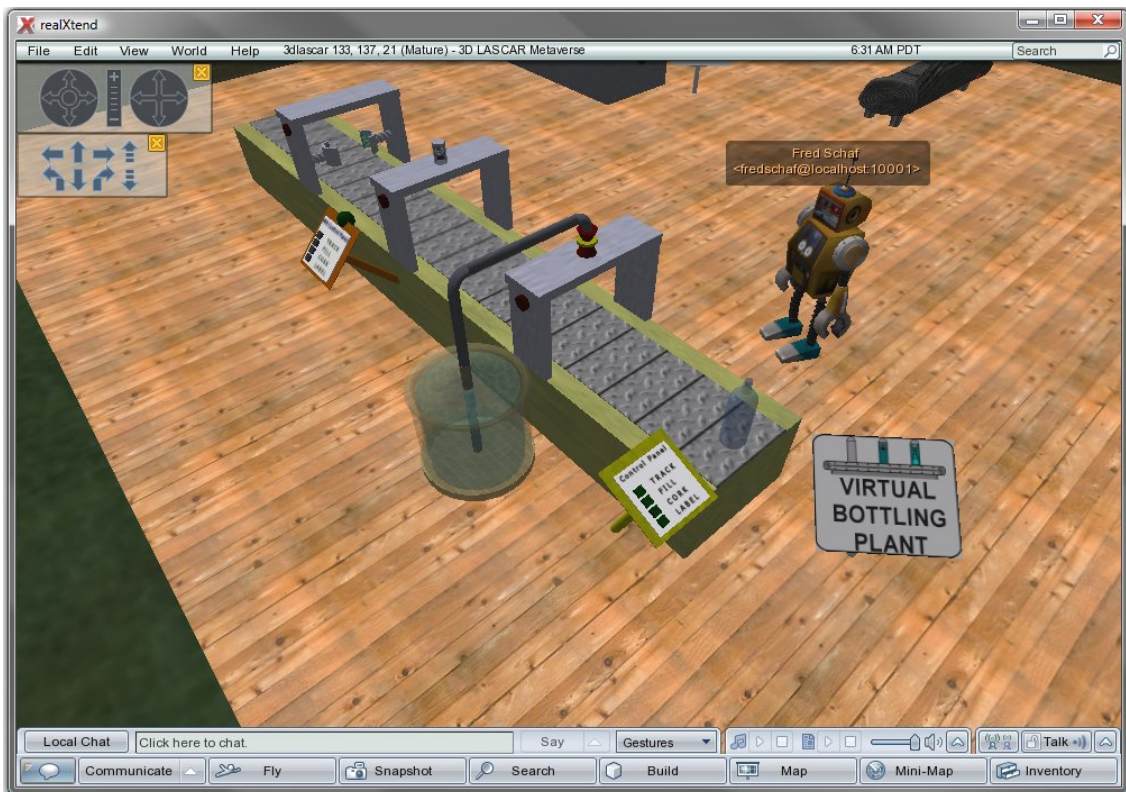


Figura 61: Planta de envase virtual - interface no *RealXtend Viewer*.

ISaGRAF 3.30 (ICS Triplex, 2011), está em utilização há quatro anos em disciplinas de Sistemas de Automação tanto na graduação quanto na pós-graduação em Engenharia Elétrica da UFRGS para ensino-aprendizagem das linguagens de programação padronizadas pela norma IEC 61131-3. A Tabela 4 organiza as informações de sensores e atuadores presentes na planta com seus tipos de dados.

Tabela 4: Informações dos sensores e atuadores da planta de envase virtual.

Sensores			Atuadores		
nome	tipo	observação	nome	tipo	observação
<i>sensor1</i>	booleano (digital)	detecção abaixo da válvula	<i>track</i>	booleano (digital)	acionamento da esteira de transporte
<i>sensor2</i>	booleano (digital)	detecção abaixo da braço de lacre	<i>fill</i>	booleano (digital)	abertura da válvula de enchimento
<i>sensor3</i>	booleano (digital)	detecção a frente da rotuladora	<i>cork</i>	booleano (digital)	acionamento da braço de lacre
<i>reserv</i>	inteiro (analog.)	nível do reservatório de líquido	<i>label</i>	booleano (digital)	acionamento da rotuladora

Para a criação do cenário *HiL* de baixo custo, e assim, testes de realidade mista envolvendo o *OpenSim/RealXtend*, foi desenvolvido com suporte da *LibOpenMetaverse* um comunicador para criar a “hiperconexão” (vide Seção 2.6) com um controlador real. Como controlador foi utilizada uma placa de desenvolvimento *Arduino* (ARDUINO, 2011), modelo 2009, em conjunto com componentes elétricos básicos. A placa *Arduino* é de baixo

custo e de fácil programação usando uma linguagem próxima a *C* convertida em linguagem de máquina para ser executada no microcontrolador *ATmega328* de 32 Kbytes de memória *flash* da fabricante *Atmel*. O *software* de hiperconexão foi implementado/escrito durante estadia na Universidade de Bremen em cooperação com o Prof. Dr. Dieter Müller (MÜLLER; SCHAF, 2009). A integração da hiperconexão com a planta de envase virtual segue o diagrama da Fig. 62.

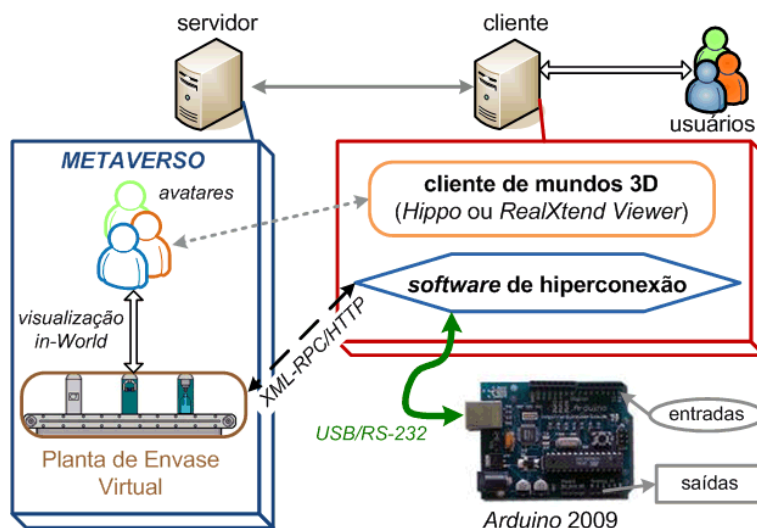


Figura 62: Estudo de caso: diagrama de *HiL* com o *Arduino*.

A demonstração da implementação deste estudo de caso está ilustrada na Fig. 63. Embora simples, essa interconexão possibilita que uma infinidade de elementos reais possam ser integrados. A hiperconexão desenvolvida usa interfaces de comunicação comuns, como: *XML-RPC*, para conexão com o *metaverso*, e a *serial (USB/RS-232)*, com a placa de desenvolvimento *Arduino*. Estas interfaces são de simples implementação em qualquer linguagem de programação. A comunicação do *software* de hiperconexão com a planta se dá *in-world* através de mensagens em canais reservados a *scripts*. Um painel de visualização de controle *HiL* também foi desenvolvido para monitoramento dos atuadores. O controle da planta embora representado por botões, pode naturalmente ser programado automaticamente no microcontrolador. Os dados dos sensores também são representados por *LEDs* ligados na placa de desenvolvimento *Arduino*. Há, portanto, o fluxo de informações bidirecionais entre os componentes. O acionamento e a visualização podem também ser efetuados no *hardware* conectado ao *metaverso*, embora esta visualização seja demasiado simples. É importante notar que o controlador também se conecta ao *metaverso* como um cliente, possibilitando, assim, a distribuição de recursos.

5.3.4 Componentes intercambiáveis com *Processing* e *Arduino*

A partir do estudo de caso anterior, foram desenvolvidos simulações e programas para incorporar a estratégia de componentes intercambiáveis usando a placa de desenvolvimento *Arduino*. O *software Processing* (desenvolvido por uma equipe do *MIT*), baseado em *Java*, foi empregado para a criação de componentes virtuais por facilitar a escrita de código e proporcionar bibliotecas gráficas pré-selecionadas para visualizações simples. Outros *softwares* também poderiam ser igualmente utilizados, como o *LabVIEW*, o *MatLab*, o *EasyJava*, etc.

A ideia é criar componentes intercambiáveis que sejam acessados por um gerenciador

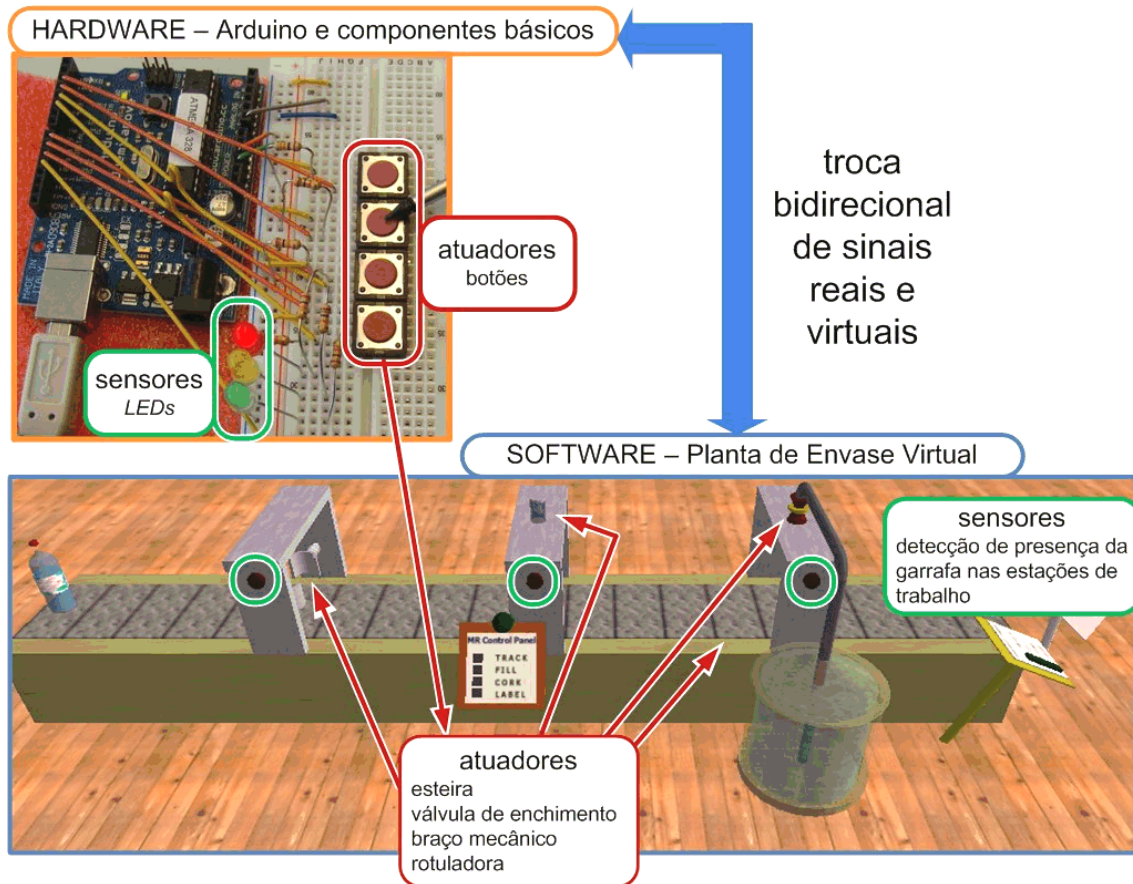


Figura 63: Estudo de caso: *HiL* com o *Arduino* e a planta de envase virtual no *OpenSim*.

que efetua conexões (*XML-RPC*) com cada um dos componentes (servidores) como anteriormente definido no diagrama da Fig. 52. A Fig. 64 ilustra o uso do *Arduino* como controlador e a planta agora modelada no *Processing*. O conexão se dá por três programas (ambos desenvolvidos usando o *Processing*): um cria um servidor *XML-RPC* que simula o comportamento da planta de envase (pode ser visualizado no *GUI*); outro cria outro servidor que se conecta via *USB/RS-232* com o controlador (no caso *Arduino*); e o último interconecta os *I/Os* dos componentes através da criação de dois clientes (cada um se conecta com um servidor).

Logicamente, que a arquitetura pode ser expandida para qualquer controlador e planta. A complexidade, todavia, aumenta de acordo com os elementos envolvidos. Por exemplo, pode-se modelar um controlador *PID* no *MatLab* e este ser usado para controlar a planta térmica (vide Fig. 56), mas para isso devem-se desenvolver dois servidores *XML-RPC* conectado a cada um dos componentes. Neste caso o servidor *XML-RPC* no *MatLab* é tarefa mais simples do que tornar o experimento acessível via *XML-RPC*, pois esse usa o protocolo *Modbus RTU* com uma biblioteca específica para o *Eclipse SCADA*.

Nota-se que o importante é a padronização do meio de comunicação (*XML-RPC* via *HTTP*) e do “protocolo”. A interação via *XML-RPC* não é feita por mensagens e sim por chamadas a procedimentos remotos. Assim, se quisermos ler as entradas de um determinado componente, por exemplo, a planta virtual de envase (*VirtualBottlePlant*), deve-se usar algo como: `VirtualBottlePlant.ReadSensors()`. A chamada sem parâmetros retornaria um vetor de valores representando os valores dos sensores. Analogamente, a escrita de um atuador é dada por:

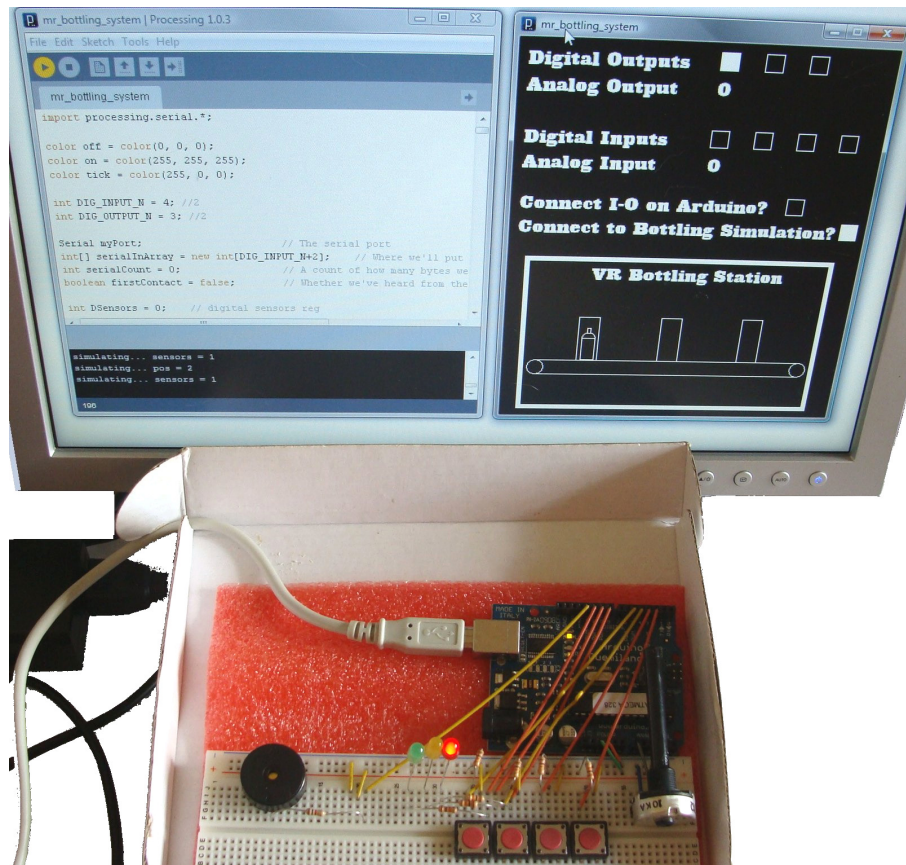


Figura 64: Estudo de caso: *HiL* com o *Arduino* e a planta de envase virtual no *Processing*.

```
VirtualBottlePlant.WriteActuator("nome/número", "valor").
```

Para facilitar a padronização de chamadas, cada componente possui um procedimento `InfoReq()` no qual são listados as chamadas implementadas e os respectivos parâmetros associados às entradas e saídas de cada um. Desta forma pode-se criar um serviço para cada componente, facilitando a interoperabilidade e a reusabilidade de cada componente. O laboratório remoto *ACT* possui algo do gênero de padronização de comunicação com experimentos usando modelos do *MatLab* (vide Seção 3.4.3). Já o laboratório da Universidade de Deusto (vide Seção 3.5.3.3) usa *WSs* implementados em *AJAX* em seus elementos, o que facilita em muito a configuração e arquitetura de conexão dos experimentos. Diferentemente do *OPC-DA* implementado no trabalho anterior do autor (SCHAF, 2006), o *XML-RPC* é independente de *OS* e linguagem de programação e possibilita interconexão via *Web*.

5.3.5 Sistema Tutor Autônomo Simples

A implementação completa de um sistema tutor é uma tarefa muito complexa que deriva do foco deste trabalho. Por esse motivo foi proposto um sistema simplificado baseado em informações de interações e representações de *feedback* tanto no *MOODLE* quanto no *OpenSim*. A base para a elaboração do *feedback* em ambos os casos, entretanto, é a mesma. Um sistema de coleta de informações baseado em agentes do *SMA* foi desenvolvido. Foi projetado um concentrador de informações, chamado de *ITS_server*, para tornar essa tarefa integrável e interoperável com outros módulos. O “servidor” possui interface *XML-RPC*. A Fig. 65 elucida o funcionamento do presente estudo de caso tanto

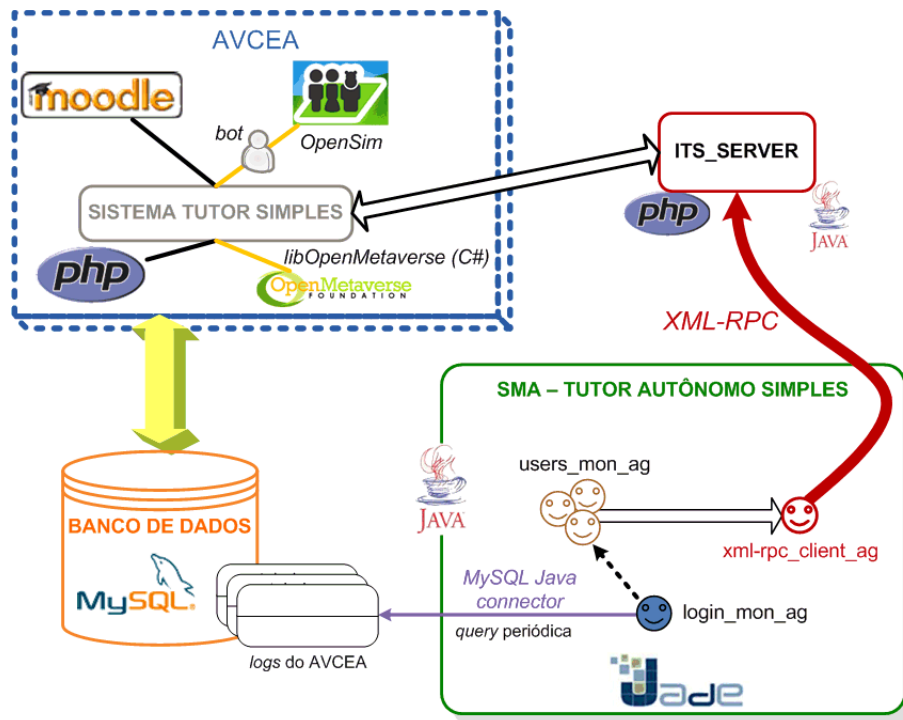


Figura 65: Estudo de caso: diagrama do sistema tutor autônomo genérico.

para tutores com resposta no AVA *MOODLE* quanto no *metaverso* do *OpenSim*.

O funcionamento do tutor autônomo inicia com a monitoração de conexão de usuários, através do agente `login_mon_ag` que semelhantemente ao monitorador de agendamento (vide Seção 5.3.1) vasculha o banco de dados procurando uma informação de login desta vez na tabela `logs` do *MOODLE*, por *default* nomeada de `mdl_log`. A tabela contém os seguintes dados: identificador do dado (*id*), tempo (*time*), número identificador do usuário (*userid*), endereço *IP* do cliente (*ip*), curso (*course*), módulo (*module*), identificação do comando (*cmid*), ação (*action*), localizador de recursos (*url*) e nota adicional (*info*). Naturalmente, pode-se ligar o identificador do usuário ao nome através da relação da tabela de usuários e analogamente o nome do curso também. As ações usadas são simples: visualização (*view*), *login*, *logout*, tentativa de reposta de questionários (*attempt*), adicionar recurso (*add*), assinalar tarefa (*assign*), etc. Logo, o agente `login_mon_ag` lê periodicamente a `mdl_log` buscando uma ação de *login*. Assim que este agente a encontra um outro agente é criado, chamado de "user" `_mon_ag`, onde `user` é o nome usuário que se conectou. Esse último agente tem a tarefa de buscar informações (passadas e presentes) do usuário que se conectou e informá-las ao `ITS_server` via cliente *XML-RPC*, executando do procedimento `Inform(timestamp, usuário, dado)`. Esta operação pode ser ilustrada pelo código-fonte do `login_mon_ag` mostrado abaixo.

```

login_mon_ag.java
1  import java.sql.*;
2  import jade.core.*;
3  import jade.domain.*;
4  import jade.lang.acl.ACLMessage;
5  import jade.wrapper.*;
6  public class login_mon_ag extends Agent {
7      protected void setup() {
8          loop = new TickerBehaviour(this, 500) {

```

```

9         protected void onTick() {
10             ACLMessage msg = null;
11             System.out.println("Verifying Status on " + now("ddMMyyyyHH"));
12             String url = "jdbc:mysql://localhost:3306/";
13             String dbName = "moodle";
14             String driver = "com.mysql.jdbc.Driver";
15             String userName = "****"; String password = "****";
16             String query = "SELECT * FROM 'mdl_log' WHERE action = 'login' ";
17             Statement stmt; ResultSet rs;
18
19             try {
20                 Class.forName(driver).newInstance();
21                 conn = DriverManager.getConnection(url+dbName,userName,password);
22                 stmt = conn.createStatement();
23                 rs = stmt.executeQuery(query);
24                 while(rs.next()) { name = rs.getString("userid"); }
25                 conn.close();
26             }
27             catch (Exception e) { e.printStackTrace(); }
28
29             if (result != null) {
30                 String ag_class = "agents.UserDataCollectorAgent";
31
32                 try {
33                     AgentController agc;
34                     agc = getContainerController().createNewAgent(name+"_mon_ag",ag_class,null);
35                     agc.start();
36                     System.out.println("iniciado monitor para o usuário " + name);
37                 }
38                 catch (Exception e) { e.printStackTrace(); }
39             }
40         }
41     }

```

Nota-se no código que as bibliotecas associadas a criação dinâmica de agentes foram carregadas (linhas 3 a 5). Se detectado o *login* do usuário (linha 27) de nome *name*, um agente para monitorá-lo e para coletar informações é criado e iniciado (linhas 29 a 35). Esse agente é identificado através da associação com o usuário em questão ("usuário" *_mon_ag*) (linha 33) e seu comportamento é controlado pela classe do agente *UserDataCollectorAgent* (linha 28). Esta classe se encontra descrita no código abaixo.

UserDataCollectorAgent.java	
1	import java.sql.*;
2	import jade.core.*;
3	import java.util.Vector;
4	import org.apache.xmlrpc.XmlRpcClient;
5	public class UserDataCollectorAgent extends Agent {
6	protected void setup() {
7	loop = new TickerBehaviour(this, 500) {
8	protected void onTick() {
9	semelhante ao do login_mon_ag entre as linhas 10 e 26 com exceção da busca (query)
10	query = "SELECT action FROM 'mdl_log' WHERE userid == 'usuário em questão'";
11	String function = null; String value = null; Object its_result = null;
12	try {
13	XmlRpcClient server = new XmlRpcClient("http://localhost:7777");
14	Vector params = new Vector();
15	params.addElement(new Integer(now()));
16	params.addElement(new String(name));
17	params.addElement(new String(dado));
18	its_result = server.execute("its_server.Inform", params);
19	System.out.println("> " + (String) its_result);
20	}

```

21         catch (Exception e) { System.err.println("XML-RPC Java Client: "+ e.printStackTrace()); }
22     }
23     });
24     addBehaviour(loop);
25 }
26 }

```

Observa-se que nesse código está carregada a biblioteca do *XML-RPC* (linha 4). E que a chamada *XML-RPC* (linhas 12 a 21) transporta o dado coletado (*result*) ao *ITS_Server* via chamada remota do procedimento *Inform*.

O procedimento de coleta de informações de usuários é considerado um exemplo de *data mining*. Se o usuário, por exemplo, ainda não efetuou a “entrega” de uma tarefa ou ainda não executou a experimentação dentro do prazo estipulado pelo tutor, este agente pode inferir mensagens em blocos de *feedback* no AVA, como “você ainda não efetuou a tarefa designada”. A adaptação de conteúdo pode ser exemplificada simplesmente pela alteração de um *URL*, ou de uma informação do banco de dados de acordo com as ações do estudante.

Figura 66: Estudo de caso: bloco de tutoriamento integrado ao *MOODLE*.

Para a representação inicial do *feedback* do sistema tutor no AVA (interface *Web* tradicional) foi desenvolvido um bloco (módulo) escrito em *PHP* que segue a arquitetura do *MOODLE*. A Fig. 66 ilustra o tutoriamento autônomo integrado ao *MOODLE*, no caso do “curso” experimentos remotos, chamado de monitor de experimentos remotos. Obviamente, o tutor neste caso é praticamente neutro, pois não força a interação e, sim, apenas sugere uma informação ao usuário. Cabe ao usuário aceitar ou não a sugestão e acatá-lo ou não.

Noutra representação pode-se personificar o tutor através de um *avatar* autômato, isto é, um *bot*, chamando o tutor de *tutor_bot*, implementado através do desenvolvimento de um programa-cliente que usa o conjunto de bibliotecas da *LibOpenMetaverse*. A Fig. 67 ilustra a representação deste *bot* interagindo com um usuário que está buscando informações sobre o funcionamento de um experimento no *metaverso*, no caso, um simulador de *pick-and-place* numa estação modular de produção (*MPS*). Vale lembrar que o *tutor_bot* interage com o usuário de forma simples (tabela de interações pré-programadas) e não é dotado de qualquer inteligência, já que para isso é necessário o uso

de ferramentas de inteligência artificial, reconhecimento de texto/fala, etc. Com base no conhecimento das interações do usuário com o ambiente (*logs*) ele pode oferecer ajuda em forma de texto (visualizado no *chat box*) com opções e se movimentar para próximo do usuário ou de objetos.

Tanto o bloco de *feedback* do *MOODLE* quanto o *bot* no *OpenSim* podem estar interconectados usando o pacote *SLOODLE* que oferece sincronização de mensagens de chat. Esta ferramenta é útil para representar ações do *ITS_Server* para adaptação de conteúdo no *AVCEA*, embora esta adaptação seja bastante simples e altere muito pouco no “ambiente”.



Figura 67: Estudo de caso: representação do sistema tutor (*bot*) no *OpenSim*.

5.3.6 Colaboração no AVCEA

Embora intrínseca, a colaboração no AVCEA via *MOODLE* ou *OpenSim* deve ser incentivada. Esse incentivo é alcançado de maneira análoga à inferência do sistema tutor autônomo via *ITS_Server*. A coleta de informações também ocorre, o que muda é a base de conhecimento para a inferência de colaboração. Ou seja, quando mais de um usuário está presente no AVCEA o sistema de apoio à colaboração busca informações relevantes dos usuários para possível colaboração. Estes estudo de caso é meramente teórico e ilustrativo, uma vez que foi provado anteriormente que o sistema de tutoriamento autônomo é factível, esse também deve ser.

Assim, imagina-se o cenário onde o usuário “A”, que não efetuou ainda a tarefa de experimentação, e outro usuário “B”, que já efetuou esta tarefa com êxito, se encontram no AVCEA. Logo, pode-se cruzar as informações dos usuários e inferir que o usuário “A” interaja com o “B” para obter informações sobre a tarefa, ou analogamente, inferir que o usuário “B” interaja com o usuário “A” para oferecer ajuda. A extrapolação desse cenário para mais de dois usuários é evidente, assim como a interação *offline* (assíncrona) entre usuários quando alguma das partes não estiver *online* no momento.

5.4 Protótipo 3DAutoSysLab

Para validar experimentalmente a arquitetura proposta foi desenvolvido um protótipo de laboratório virtual tridimensional com suporte à realidade mista e à colaboração dos usuários, chamado de *3DAutoSysLab*³. Como permitido pela arquitetura, existem duas “interfaces” para visualização do ambiente, o AVA, que é o *MOODLE*, e o *OpenSim*, que é o *software* gerenciador de mundos 3D virtuais.

O controle e o gerenciamento de usuários é interligado através do uso do mesmo repositório de dados, neste caso o banco de dados *MySQL*. A maioria das ferramentas de *software* empregadas no protótipo são livres, algumas até de código aberto, para propósitos não comerciais. Uma exceção são os gerenciadores de experimentos da planta didática *Foundation Fieldbus* e da planta térmica, que foram desenvolvidos anteriormente (SCHAF; PEREIRA, 2009) e utilizam o *Eclipse SCADA*, que tem licença comercial.

O módulo de realidade mista foi desenvolvido em sua maior parte por programas escritos em linguagens de programação *PHP* e *Java*. Esses funcionam como gerenciadores de experimentos. O gerenciador de experimento previamente desenvolvido, que emprega o *Eclipse SCADA*, também foi integrado. Todos os “gerenciadores” possuem interface de comunicação *HTTP*, comunicação com repositório de dados (*DB*), e alguns oferecem *WSs*, como *XML-RPC*, e *sockets* no modelo cliente-servidor.

Assim como nos estudos de caso apresentados, no módulo de suporte à adaptação de conteúdo e *feedback* foi empregado o *JADE* como *framework* para desenvolvimento dos agentes que é de código aberto e livre. Como especificado pela arquitetura, os agentes coletam informações de interações armazenadas no repositório dos usuários com o sistema. A partir destes dados inferem-se modificações do conteúdo do sistema e sugestões aos usuários. Tutores autônomos e suporte à colaboração estão programados usando o *JADE*, *Java* e *PHP*.

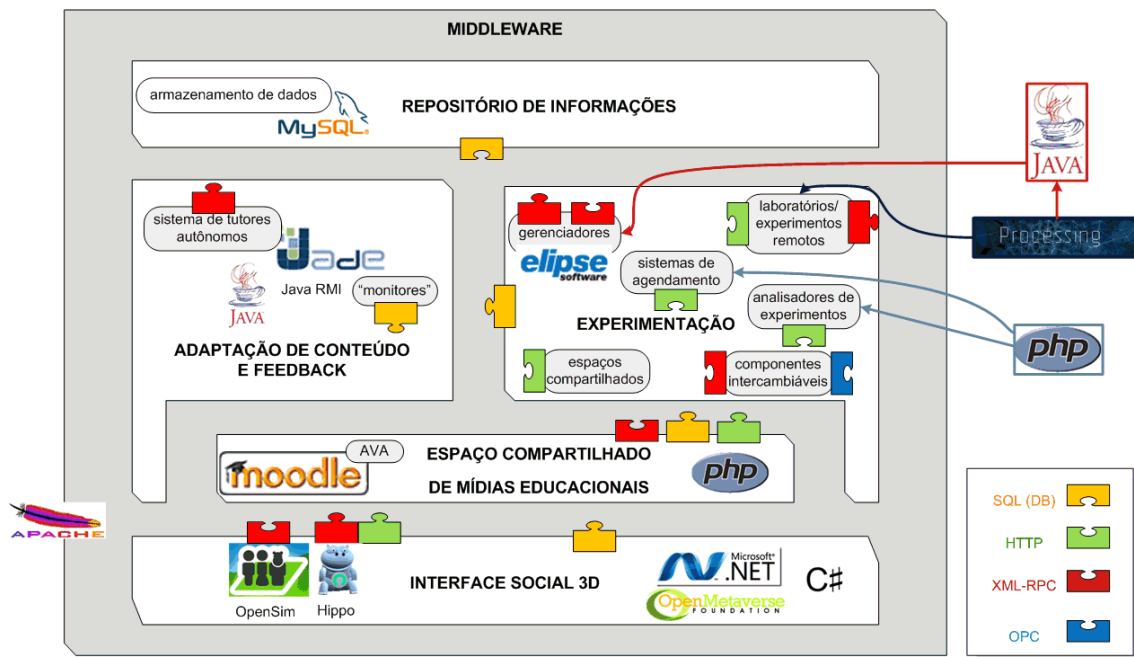


Figura 68: Protótipo - arquitetura de implementação com ferramentas e interfaces de comunicação.

³em alguns artigos do autor o protótipo foi referenciado como *MRCS-CARLab3D*.

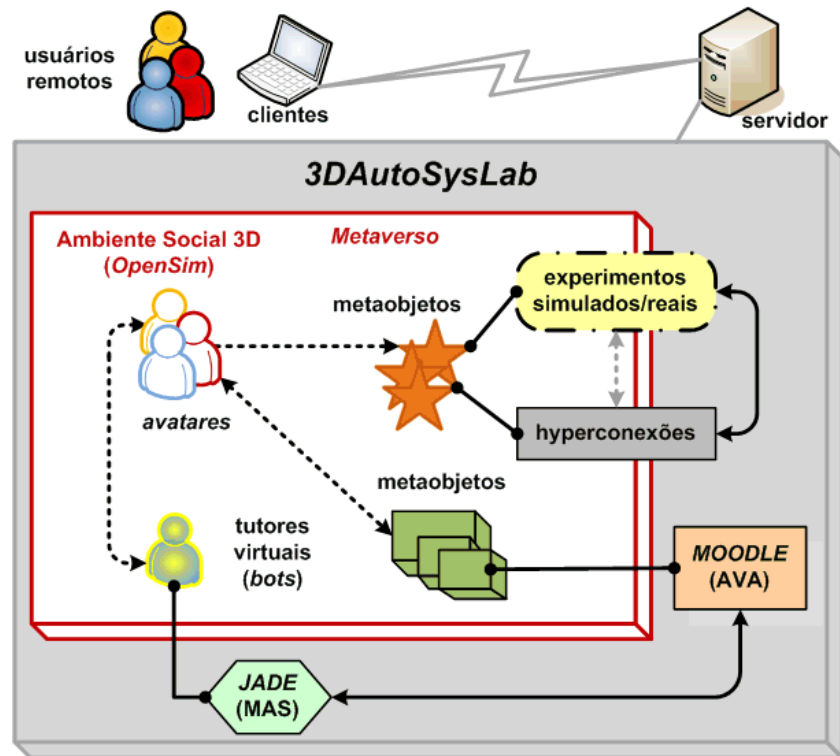


Figura 69: Protótipo - diagrama de implementação.

Assim, a arquitetura pode ser implementada usando o conjunto de ferramentas ilustrado na Fig. 68. Nota-se também a especificação das interfaces de comunicação entre os módulos. Linguagens de programação empregadas no desenvolvimento dos programas (módulos) também estão representadas. Foram suprimidos os conceitos envolvidos em cada módulo (isto está implícito).



Figura 70: Protótipo 3DAutoSysLab - experimentos, colaboração, visualização, salas virtuais, etc.

Integrando-se todas as funcionalidades desenvolvidas nos estudos de caso, pode-se

imaginar o protótipo com o seguinte diagrama simplificado da Fig. 69. Nele foi idealizada a interface social 3D com o *OpenSim/RealXtend* onde toda infraestrutura do AVCEA está disponível. Objetos virtuais são interconectados a experimentos, a objetos de aprendizagem e a elementos do *MOODLE* via pacote *SLOODLE* e também interface *Web*. O tutoriamento autônomo (*bots*) é controlado pelo SMA modelado no *JADE*, que também está a par das interações no *MOODLE* via monitoração do banco de dados em comum. Desta maneira usuários, experimentos, objetos virtuais, tutores, e materiais educacionais são agrupados em um único ambiente (*metaverso*) com representação similar aos presentes no dia-a-dia real. Podem-se, inclusive, adicionar elementos normalmente impossíveis na realidade, como o tutoriamento automatizado 24h.

A capacidade de acompanhamento de aulas presenciais também foi incluída usando um *script* livremente distribuído na *Web* para visualização de imagens e vídeos, o *Free-View*. Usando uma câmera capaz de transcodificar o vídeo capturado no formato *MPEG-2* com codificação *H.264* e disponibilizá-lo pelo protocolo *RTSP* (*open RTSP*), é possível visualizar e ouvir a aula de modo online. Antes essa funcionalidade normalmente era possível usando *softwares* de conferência como *Skype* e o *Adobe Connect*. A Fig. 70 ilustra esta alternativa. Vale ressaltar que isso somente é possível graças à interoperabilidade do *OpenSim/RealXtend* com o *plug-in* do *Apple Quicktime* (APPLE, 2011) que deve ser instalado no servidor.

6 VALIDAÇÃO

Sabe-se que para validar uma proposta deve-se estipular primeiro a metodologia de avaliação na qual sejam definidos pontos-chaves de conformidade. Neste capítulo serão abordadas três metodologias: avaliação da conformidade dos objetivos do trabalho; avaliação do AVCEA criado (protótipo), segundo critérios de *software* educacional; e avaliação da resposta dos usuários. Cada avaliação será então abordada nas próximas subseções.

6.1 Avaliação da Conformidade dos Objetivos do Trabalho

Dentro do esperado, os objetivos da proposta foram alcançados, já que a proposta de arquitetura atende à maioria dos problemas apontados no ensino, sobretudo na parte experimental (prática com experimentos) e no EaD. Podem-se comentar os resultados obtidos na integração das funcionalidades inicialmente tidas como objetivos, da seguinte maneira: [✓] para concretizado e com engajamento pessoal; [●] para concretizado; e [○] para necessitando concretização.

- ✓ materiais didáticos organizados e gerenciados por AVAs
 - A proposta é factível; colaborações dos professores da instituição aumentaram em quantidade e qualidade os materiais desenvolvidos. Há também o interesse dos estudantes nesta organização que facilita a estruturação do curso e o planejamento de ensino-aprendizagem tanto dos estudantes quanto dos professores.
- suporte à colaboração e à cooperação no aprendizado e em tarefas didáticas
 - Embora ainda minimizado em tarefas práticas, a colaboração a nível de interações nos AVAs ocorre dentro do esperado o que facilita a disseminação do conhecimento por construtivismo social.
- suporte à interação social
 - Usuários normalmente alteram perfis e criam grupos, o que são indica que há interações sociais e, conseqüentemente, bases para uma colaboração sólida entre usuários.
- interfaces imersivas com representações tridimensionais
 - O baixo uso ainda é reflexo da pouca experiência e novidade da “ferramenta”. Vale lembrar também que a infraestrutura ainda é precária para a utilização em larga escala dessa alternativa.
- características pseudolúdicas (*serious gaming*)
 - Esta característica ainda é absolutamente nova e a criação de cenários educacionais envolvidos com competitividade e estruturação de jogos computacionais ainda é uma tarefa muito complexa para desenvolvimento em larga escala. Apesar de os apelativos desta alternativa serem cada vez mais presentes no dia-a-dia de usuários modernos e a resposta muitas vezes ser positiva (BELLOTTI et al., 2009).

✓ experimentação remota

A prática com *SBBT* é positiva e estudantes aprovam o uso destes experimentos pela flexibilidade temporal e espacial dessa alternativa.

- características de realidade mista

Esta mistura é altamente benéfica e possibilita flexibilidade de criação de cenários e maior didática na ilustração de conceitos teóricos.

- suporte a tutores e facilitadores de colaboração autônomos

Este conceito ainda é muito pouco utilizado e polêmico, mas futuramente será uma alternativa real e efetivamente viável.

- características modulares e interoperáveis

Toda estrutura da arquitetura pesa nestes pilares que possibilitam a integrabilidade e distribuição de recursos. O efeito destas características possibilita a criação de uma ferramenta efetivamente colaborativa em todos os níveis.

6.2 Avaliação do Protótipo

É muito difícil estabelecer métricas para avaliar soluções de *software* voltados para o ensino-aprendizagem, já que os parâmetros de comparação muitas vezes são muito vagos. Há diversos trabalhos que propõem uma comparação entre laboratórios reais e remotos (CORTER et al., 2004), ou laboratórios remotos e simulações (MA; NICKERSON, 2006), ou avaliações de *software* colaborativos (NEALE; CARROLL; ROSSON, 2004), mas em nenhum desses há um consenso ou uma teoria concreta dos critérios de avaliação ou comparação entre soluções.

Logo, prefere-se estabelecer os acertos e erros como critérios de avaliação no projeto de *software* educacional. Assim, um método baseado no TICESE - Técnica de Inspeção Conformidade Ergonômica de *Software* Educacional - com o *ErgoList* (ERGOLIST, 2011) foi utilizado para avaliar protótipo. A Tabela 5 mostra os resultados obtidos pela resposta de mais de 100 perguntas subdivididas em 18 grupos.

A avaliação do questionário *Ergolist*, portanto, apontou para uma conformidade total próxima a 70%, o que para um protótipo é tido como uma boa validação teórica. Alguns grupos no entanto não foram atendidos corretamente, como o caso dos erros que ainda não são tratados de maneira adequada no protótipo.

Certamente pode-se também comparar com implementações similares presentes na comunidade científica. Alguns trabalhos apresentados no capítulo de estado da arte (vide Capítulo 3), como o *3D Physics Lab*, o *Second Lab*, o *Magee Campus* e o *JDoc* que possuem representação tridimensional e ainda características semelhantes ao protótipo. Outros ainda, como o *Mixed Reality Mechatronics Lab*, o *ACT*, o *CoLab*, o *VCLab* e o *HTI Solar Energy e-Learning Lab* possuem representação diferente, mas o mesmo propósito. Assim, apresenta-se a Tabela 6 na qual características desejadas na arquitetura proposta são usadas como quesitos de avaliação. Nela nota-se que muitas implementações de ambientes virtuais voltados ao ensino-aprendizagem, embora muitas vezes disponibilizem materiais educacionais, não estão organizados por um AVA, ou ferramenta similar, integrados. A diferenciação também pode ser observada no quesito colaboração onde nem todos suportam a modalidade de ensino teórico e prático. Já no quesito tutoriamento e *feedback* somente o protótipo desenvolvido possui um tutor autônomo simples para *feedback*. O quesito interoperabilidade ressalta a possibilidade de expansão e reusabilidade de módulos em outros projetos. Nele, vale ressaltar, o *3D Physics Lab*, o *Second Lab* e o protótipo, que foram desenvolvidos com esta preocupação, em contraste com o *DeriveServer*,

Tabela 5: Resultados da avaliação TICESE.

Grupo <i>versus</i> questões	conformes	não conformes	não aplicáveis	%
Concisão	4	1	9	80,0
Mensagens de erro	5	3	1	62,5
Flexibilidade	1	2	0	33,3
Legibilidade	8	1	18	88,9
Significados	8	0	4	100,0
Proteção contra erros	1	3	3	25,0
Agrupamento por formato	10	3	4	77,0
Experiência do usuário	2	3	1	40,0
Presteza	9	3	5	75,0
Controle do usuário	2	1	1	66,7
Correção de erros	0	2	3	0,0
Consistência	8	0	3	100,0
Agrupamento por localização	4	0	3	100,0
Densidade informacional	3	2	4	60,0
Feedback	4	3	5	57,0
Compatibilidade	11	2	8	84,6
Ações explícitas	2	1	1	66,7
Ações mínimas	3	2	0	60,0
Total	85	32	77	72,6

o *JDoc* e também o *Magee Campus*. Embora estes três últimos possibilitem a integração de funcionalidades extras, a arquitetura de comunicação e os protocolos “escolhidos” dificultam em muito esta tarefa. Logo, avaliando as funcionalidades, pode-se apontar que o *MRCSE 3DAutoSysLab* é possivelmente superior para o uso efetivo de AVCEAs.

Também foram testados critérios associados a qualidade de serviço (*QoS - Quality of Service*), como o tempo de resposta, a latência de rede. A estabilidade do protótipo para variadas situações com número crescente de usuários foi igualmente analisada. A conversação de usuários no *OpenSim* é dependente da velocidade de conexão de ambos os usuários, do mesmo modo que ferramentas consagradas como o *Skype*, ou seja são tempos de rede e não de computação (ou do servidor). O *OpenSim* suporta o *FreeSwitch* (FREESWITCH, 2011) que proporciona um serviço de *VoIP* integrado (*in-World*). Esta implementação é ligeiramente pior que a conversação via *Skype* nas mesmas condições. A simulação *in-World* usando somente *scripts* para programação de interatividade de objetos acrescenta um tempo de processamento intrínseco do *OpenSim* para qualquer *script* que normalmente é de 1 segundo. Assim, para objetos que estejam interconectados com recursos externos soma-se este tempo de processamento ao tempo de latência da rede. Para chamadas *HTTP* o tempo médio de resposta é de 2 segundos. Já se o usuário desejar visualizar elementos *Web* em objetos virtuais (aplicação de textura dinâmica), o tempo de resposta sobe para 5 segundos em média para cada mudança de “arquivo”. Portanto, a visualização de *slides* e elementos gráficos *Web* fora do navegador (o *Hippo* possui um *Web Browser* integrado) exige paciência, pois a transição de um *slide* para o outro leva 5 segundos. Quando, porém, o elemento a ser visualizado é um vídeo, o tempo de resposta é igual ao tempo de carregamento (5s) mais o de “bufferização” (dependendo da conexão do usuário). Já para elementos que exigem conexão com servidor *XML-RPC*, o tempo de resposta é de 2 segundos, 1s para o *script* e outro para a inicialização do

Tabela 6: Comparação de implementações de ambientes virtuais voltados ao ensino-aprendizagem com a arquitetura e o protótipo proposto.

“-” não suporta e “X” suporta.

x	AVA	experim.	colab.	tutores/ <i>feedback</i>	interop.	interface
VCLab	-	virtual	-	-	-	<i>Web VRML</i> +
ACT	-	misto	-	-	<i>MatLab</i>	<i>Web</i>
CoLab	-	misto	som. exp.	-	-	<i>Java 2D</i>
HTI SEL	X	real	som. AVA	<i>quizz</i>	não espec.	<i>Web</i>
DeriveServer	X	elementos mistos	X	-	difícil	<i>Web VRML</i> +
3D Physics Lab	-	misto	X	-	possível, <i>Java</i> , não espec.	Mundo 3D
Second Lab	-	real	X	-	possível, variada	Mundo 3D
Magee Campus	-	misto	X	-	não espec.	Mundo 3D
JDoc	-	-	não espec.	não espec.	não espec.	Jogo 3D
Arquitetura Proposta	X	misto com compon. intercamb.	X	X	X	Mundo 3D e <i>Web</i>
MRCSE 3DAutoSys Lab	X	misto com compon. intercamb.	X	simples	<i>XML-RPC, DB, HTTP, OPC-DA</i>	Mundo 3D e <i>Web</i>

cliente *XML-RPC*. O tempo requerido para conexão do *Hippo* com o *OpenSim* é de aproximadamente 10 segundos. O *metaverso* é estável para as mais variadas interações com número crescente de usuários, conexões e desconexões, mas notou-se um problema na simulação física de objetos interativos. Objetos marcados com o atributo “físico” são simulados por uma máquina física associada ao *OpenSim*, por padrão, o *ODE* (*ODE*, 2011) - (*Open Dynamics Engine*), que, em algumas testes, ocasionou a falha total do servidor (*server crash*).

A validação estaria completa com o emprego efetivo do protótipo desenvolvido no ensino, mas esta possibilidade ainda envolve muitos recursos computacionais e de rede o que dificulta a utilização. Vale lembrar também que isso requer muito tempo, no mínimo 2 semestres, e muitos materiais de ensino associados, o que ainda não é uma realidade.

6.3 Avaliação dos Usuários

Como mencionado anteriormente, o protótipo ainda não foi testado por estudantes, mas alguns usuários ajudaram a testar o sistema com avaliação positiva. A Fig. 71 ilustra algumas capturas de interações no *metaverso* criado. Nota-se que o sentimento de presença é evidente, pois cada usuário vê o *avatar* do outro.

Mesmo que o protótipo não tenha sido testado alguns elementos integrantes dele foram. A integração do MOODLE com experimentos remotos, componentes intercambiá-



Figura 71: Protótipo 3DAutoSysLab - interações de usuários.

veis e um *feedback* de experiência simples foram testados no curso de Engenharia Elétrica na UFRGS, mais precisamente, na disciplina de Sistemas de Controle I. Diversos materiais educacionais, tarefas e experimentos estão em uso atualmente. Notou-se um aumento da motivação dos estudantes que se refletiu na queda do índice de desistência (reprovação) (SCHAF; PEREIRA; HENRIQUES, 2008). Para se ter uma noção da aceitação do ambiente, chamado então de GCAR-EAD, foi aplicado um questionário respondido por 53 estudantes. A Tabela 7 ilustra as questões e as respostas dos estudantes em porcentagens do total.

Tabela 7: Resultado da pesquisa de opinião dos estudantes do GCAR-EAD.

Fonte: (SCHAF; PEREIRA; HENRIQUES, 2008).

Na sua opinião, qual a sua avaliação do sistema GCAR-EAD?				
excelente	bom	regular	ruim	
43%	43%	0%	14%	
Na sua opinião, qual a sua avaliação dos experimentos oferecidos no GCAR-EAD?				
excelente	bom	regular	ruim	
50%	43%	0%	7%	
Na sua opinião, o que é mais didático, simulações ou experimentos reais?				
simulações	experimentos reais	ambos	a combinação dos dois	
0%	14%	36%	50%	
Na sua opinião, quais das seguintes características são pontos fortes do GCAR-EAD?				
flexibilidade temporal	flexibilidade espacial	material didático integrado	ambiente colaborativo	integração com mecanismos de busca
65%	50%	14%	57%	43%

Observou-se que as opiniões foram bastante positivas, com mais de 90% entre excelente e bom, tanto para o ambiente quanto para os experimentos empregados no ensino. Também é interessante a opinião dos estudantes quanto a simulações: nenhum acha que simulações têm valia didática e a metade acredita que a combinação entre simulação e realidade é a alternativa pedagógica mais vantajosa. A “a combinação dos dois” aqui implica na experimentação onde elementos simulados são interconectados com reais em um único experimento, enquanto que na opção “ambos” há dois experimentos distintos, um simulado e outro real, que ilustram o mesmo processo experimental com diferenças ligadas às características intrínsecas da modelagem ou do fenômeno físico. Os pontos fortes do GCAR-EAD mais lembrados foram: as flexibilidades espacial e temporal (resposta esperada) e o ambiente colaborativo. Analisando os registros do AVA e dos experimentos notou-se a preferência por horários à noite e a intensa interação no fórum do curso, justificando o caráter colaborativo do ambiente.

Espera-se, em um futuro próximo, empregar um questionário semelhante aos usuários do *3DAutoSysLab* e comparar resultados de as ambas interfaces: 2D e 3D. Também é importante fazer uma análise melhor dos dados e referenciá-los a informações como grau de instrução, idade, semestre letivo, etc.

7 CONCLUSÕES

Conforme apresentado, o trabalho demonstra uma base, conceitos e um complemento tecnológico para incrementar o ensino dos professores e a aprendizagem dos estudantes de automação e controle. Esta base é fundamentada numa arquitetura modular para o projeto de ambientes virtuais voltados ao ensino e à aprendizagem. A partir das características apontadas como vitais para o projeto destes ambientes, através do estudo do estado da arte, módulos funcionais foram idealizados e suas características descritas.

Uma das grandes vantagens dos ambientes virtuais de qualquer natureza, que possuam ferramentas de interação entre usuários, é a colaboração. Unir o processo de ensino-aprendizagem com colaboração entre usuários é a chave para a descentralização da instrução (ensino). A colaboração aliada à educação é defendida pela maioria dos pesquisadores do ensino (LEHTINEN, 2003). Deste modo teorias de construtivismo social podem ser aplicadas ao ensino-aprendizagem em ambientes virtuais com auxílio de computadores. É inegável o avanço da tecnologia em todas as áreas, e a educação e o uso de ferramentas computacionais pelos estudantes independente das aulas é evidente. Logo, unir o útil ao agradável, também na educação, motiva estudantes e aumenta a efetividade do processo de ensino-aprendizagem.

Enquanto *softwares* são desenvolvidos para colaboração de usuários, é difícil, tanto haver interações efetivas sem que se permita a formação de relações sociais, como incentivo à interação social sem o suporte a trabalhos compartilhados ou com co-autoria. Assim, a interface, idealizada na proposta de arquitetura, utiliza mundos tridimensionais visando facilitar e motivar a interação social promovendo a colaboração. A interface também facilita a navegação e o sentimento de presença dos usuários, dando a eles representações corporais físicas, ou seja, *avatars*. Isso pode parecer pouco útil, mas capta a atenção e permite melhor familiarização com o ambiente, além de afetar o psicológico dos usuários favorecendo a imersão.

Para a caracterização do ambiente de ensino é necessário que materiais de aprendizagem estejam presentes e que estratégias e objetivos sejam claros para os usuários. Portanto uma organização é necessária. Logo, a presença de uma ferramenta de gerenciamento de objetos de aprendizagem é crucial. Dentro da arquitetura, este módulo é representado pelo grupo dos AVAs, que são implementações comuns e crescentemente utilizadas nas instituições de ensino nos mais variados níveis. A adoção pelo MEC destas ferramentas somente intensifica o potencial envolvido em tais soluções para o ensino.

Nas áreas da engenharia e ciências da computação é sempre necessária a prática laboratorial. Como os custos de produção, manutenção e instalação de laboratórios com equipamentos didáticos é muito alto e inviável para muitas instituições de ensino, a alternativa virtual é quase sempre empregada. A utilização, todavia, de equipamentos e experimentos reais é vital para a preparação de profissionais. Para tanto, alternativas foram apresenta-

das para unir vantagens, baixar custos, e proporcionar flexibilidade de criação de cenários educacionais. Dentre elas estão os componentes intercambiáveis (SCHAF; PEREIRA, 2009) e os *kits* de construção (SCHAF; MÜLLER, 2010). A distribuição desses componentes permite ainda a cooperação entre instituições que podem viabilizar experimentos entre si.

Conceber respostas a estudantes em tempo de interação em ambientes virtuais é uma tarefa inviável para tutores tradicionais. A automação de *feedback* em AVCEAs é, assim, importante para sinalização de acompanhamento dos estudantes. Embora tornar este “serviço” autônomo seja uma tarefa muito complexa, tutores projetados usando tecnologia de SMAs foram idealizados e descritos para cumprir as mais variadas tarefas de tutoramento. Um módulo somente com esta funcionalidade foi especificado na arquitetura para suprir tarefas de acompanhamento, auxílio e *feedback* interativos.

De maneira a criar uma solução que possa ser reutilizável ou integrada, a interoperabilidade também foi englobada no projeto dos módulos. Assim, modularidade, expansibilidade, integrabilidade e interoperabilidade devem ser incorporadas no desenvolvimento dos módulos, pois funcionalidades estáticas e indissociáveis são facilmente ultrapassadas e podem se tornar elefantes brancos.

A escolha das tecnologias, ferramentas, conceitos ou métodos foi cuidadosamente estudada para a implementação de um protótipo capaz de desempenhar funções e características da arquitetura idealizada. Ferramentas livres e tecnologias abertas foram empregadas no projeto de módulos que devem ainda ser independentes de plataformas e sistemas operacionais para não restringir seu uso. Estão entre os destaques destas escolhas: o *Java*, o *XML-RPC*, bancos de dados *SQL*, *C#*, e o SMA (esta última segue padronização de comunicação e estruturação). Foram empregadas no desenvolvimento do protótipo implementações consagradas, entre elas: o *MOODLE* como AVA, o *JADE* para desenvolvimento de agentes, o *OpenSim/RealXtend* como gerenciadores de mundos 3D e o *Hippo/RealXtend Viewer* como visualizador destes mundos. Este protótipo, em fase de validação educacional, apresentou bons resultados em testes primários.

É notório, todavia que a simples escolha de ferramentas não garante a interação entre elas, por isso e um estudo meticuloso de um *middleware* foi conduzido para permitir a interação entre os módulos/ferramentas de forma transparente, eficiente e possivelmente padronizada. Este foi talvez o maior desafio tanto do projeto da arquitetura quanto da implementação.

Embora seja muito difícil provar a eficácia de qualquer estratégia onde se emprega tecnologia na educação (PINELLE; GUTWIN; GREENBERG, 2003; NEALE; CARROLL; ROSSON, 2004) em combinação com o ensino tradicional ou somente na modalidade EaD, alguns estudos incentivam e apontam para um futuro promissor do chamado *blended learning* ou ensino híbrido (*hybrid learning*), ou ainda ensino aberto.

Também é de conhecimento que a infraestrutura de rede atual não permite grandes taxas de transmissão (para todos os usuários) capazes de controlar tais experimentos sem atrasos consideráveis. Para nosso estudo foi constatado um atraso médio de 2s para operações de controle com realidade mista, e de 1s para o controle totalmente simulado. Para cenários onde somente equipamentos reais são empregados nos experimentos (presentes somente no servidor), a manipulação dos atuadores (desempenhada pelo controlador) não foi afetada pelo atraso, somente a visualização e as operações de controle feitas durante o processo (*runtime*) pelos usuários. Isso garante que o controle ocorra sem problemas, pois os processos dos experimentos desenvolvidos possuem uma dinâmica considerada “lenta”.

O protótipo ainda precisa ser aprimorado e é planejado o emprego em algumas disciplinas dos cursos de Engenharias Elétrica e de Automação. Futuras métricas e estatísticas de uso do sistema serão coletadas, e análises indicarão a efetividade real do sistema no ensino. Anteriormente, versões simplificadas do protótipo foram utilizadas, sem a interface social 3D e apresentaram resultados satisfatórios principalmente no aumento do interesse dos estudantes (SCHAF; PEREIRA; HENRIQUES, 2008).

Embora a área de aplicação deste trabalho seja ainda pouco utilizada no Brasil, publicações do autor em periódicos de renome internacional e o debate em conferências, tanto nacionais quanto internacionais, apontam para o impacto desta pesquisa na comunidade científica. A colaboração com instituições internacionais com a realização de projetos situou o grupo de trabalho no mapa através de colaborações com pesquisadores da América Latina e Europa. Também é importante ressaltar que a proposta deste trabalho gerou e gerará interesse entre pesquisadores, apesar de convicto de que ela não passa de mais uma gota de contribuição para a evolução das metodologias e ferramentas de ensino no Brasil.

REFERÊNCIAS

- 3DXPLORER. **3DXplorer - 3D Web accessible to all Website**. Disponível em: <<http://www.3dexplorer.com>>. Acesso em: 18 fev. 2011.
- ACCESS GRID (AG). **Access Grid Website**. Disponível em: <<http://www.accessgrid.org>>. Acesso em: 19 fev. 2011.
- ACT. **Automatic Control Telelab Website**. Disponível em: <<http://www.dii.unisi.it/~control/act>>. Acesso em: 12 fev. 2011.
- ACTIVE WORLDS (AW). **Active Worlds Website**. Disponível em: <<http://www.activeworlds.com>>. Acesso em: 13 fev. 2011.
- ALBRECHTSEN, H. Web-based Collaboratories - from centres without walls to collaboratories in use. **Journal of Digital Information Management**, [S.l.], v.2, n.1, p.2–3, 2004.
- ADVANCED DISTRIBUTED LEARNING (ADLNET). **ADL NET Website**. Disponível em: <<http://www.adlnet.org>>. Acesso em: 15 dez. 2010.
- ALVES, G. R. et al. E. Remote Experimentation Network Yielding an Inter-University Peer-to-Peer e-Service. In: IEEE INTERNATIONAL CONFERENCE ON EMERGING TECHNOLOGIES AND FACTORY AUTOMATION (ETF A), 2005, Catania, Italy. **Proceedings...** [S.l.]:IEEE, 2005. p.1023–1030.
- ALVES, G. R. et al. Remote Experimentation Network Yielding an Inter-University Peer-to-Peer e-Service. In: IEEE INTERNATIONAL CONFERENCE ON EMERGING TECHNOLOGIES AND FACTORY AUTOMATION (ETF A), 2005, Catania, Italy. **Proceedings...**[S.l.]:IEEE, 2005. p.1023–1030.
- AMI@WORK. **AMI@Work on-line Communities Wiki Website**. Disponível em: <<http://www.ami-communities.eu>>. Acesso em: 19 fev. 2011.
- ARDUINO. **Open-Source Electronics Prototyping Platform Website**. Disponível em: <<http://www.arduino.cc>>. Acesso em: 10 mar. 2011.
- ATHENA. **ATHENA Project Website**. Disponível em: <<http://www.athena-ip.org>>. Acesso em: 19 fev. 2011.

ATKAN, B. et al. Distance Learning Applied to Control Engineering Laboratories. **IEEE Transactions on Education**, New York, USA, v.39, n.3, p.320–326, Aug. 1996.

AUER, M. et al. Distributed Virtual and Remote Labs in Engineering. In: IEEE INTERNATIONAL CONFERENCE ON INDUSTRIAL TECHNOLOGY, 2003, Maribor, Slovenia. **Proceedings...** [S.l.]:IEEE, 2003. p.1208–1213.

AUTODESK. **3DSMax at Autodesk Website**. Disponível em: <<http://www.autodesk.com/3dsmax>>. Acesso em: 17 fev. 2011

AVOURIS, N. et al. ModellingSpace: interaction design and architecture of a collaborative modeling environment. In: INTERNATIONAL CONFERENCE ON COMPUTER-BASED LEARNING IN SCIENCE, 2003, Nicosia, Cyprus. **Proceedings...** [S.l.:s.n.], 2003. p.993–1004.

BAFOUTSOU, G.; MENTZAS, G. Review and Functional Classification of Collaborative Systems. **International Journal of Information Management**, Amsterdam, Netherlands, v.22, n.4, p.281–305, Aug. 2002.

BATUR, C. et al. Remote Tuning of a PID Position Controller via Internet. In: AMERICAN CONTROL CONFERENCE, 2000, Chicago, USA. **Proceedings...** [S.l.]:ACM, 2000. p.4403–4406.

BELLOTTI, F. et al. Enhancing the Educational Value of Video Games. **ACM Computers in Entertainment**, New York, USA, v.7, n.2, p.23:1–23:18, June 2009.

BENDER, E. Rules of the Collaboratory Game: virtual collaborations for sharing data and insights are increasingly key to scientific success. When they work, that is. **MIT Technology Review - Business**, Cambridge, USA, November 2004. Disponível em: <<http://www.technologyreview.com/business/13899>>. Acesso em: 20 nov. 2010.

BENEDETTELLI, D. et al. A LEGO Mindstorms Experimental Setup for Multi-Agent Systems. In: IEEE CONTROL APPLICATIONS AND INTELLIGENT CONTROL, 2009, St. Petersburg, Russia. **Proceedings...** [S.l.]:IEEE, 2009. p.1230–1235.

BEREITER, C. **Education and Mind in the Knowledge Age**. Mahwah, USA: Lawrence Erlbaum Associates, 2002. 544p.

BIG BLUE BUTTON (BBB). **Big Blue Button Software Website**. Disponível em: <<http://bigbluebutton.org>>. Acesso em: 20 fev. 2011.

BILLINGHURST, M.; KATO, H. Collaborative Mixed Reality. In: INTERNATIONAL SYMPOSIUM ON MIXED REALITY (ISMR), 1999, Yokohama, Japan. **Proceedings...** [S.l.: s.n.], 1999. p.261–284.

BLACK BOARD (BB). **Blackboard International Website**. Disponível em: <<http://www.blackboard.com>>. Acesso em: 9 dez. 2010.

BLENDER. **Blender - free open source 3D content creation suite Website**. Disponível em: <<http://www.blender.org>>. Acesso em: 17 fev. 2011.

BLOOM, B. S. **Taxonomy of Educational Objectives**. Boston, USA: Addison Wesley Publishing Company, 1956.

BOCKHOLT, U. et al. Augmented Reality for Enhancement of Endoscopic Interventions. In: IEEE VIRTUAL REALITY, 2003, Los Angeles, USA. **Proceedings...** [S.l.]:IEEE, 2003. p.97–101.

BOEDKER, S. **Through the Interface**: a human activity approach to user interface design. Hillsdale, USA: Lawrence Erlbaum, 1991. 169p.

BOHL, O. et al. The sharable content object reference model (SCORM) - a critical review. In: INTERNATIONAL CONFERENCE ON COMPUTERS IN EDUCATION, 2002, Auckland, New Zealand. **Proceedings...**[S.l.: s.n.], 2002. p.950–951.

BRUCE, B. C.; RUBIN, A. **Electronic Quills**: a situated evaluation of using computers for writing in classrooms. Hillsdale, USA: Erlbaum, 1993. 232p.

BRUNER, J. **A Study of Thinking**. Piscataway, USA: Transaction Publishers, 1956. 350p.

BRUNS, F. W. Hyper-Bonds - Distributed Collaboration in Mixed Reality. **Annual Reviews in Control**, Rotterdam, Netherlands, v.29, n.1, p.117–123, Mar. 2005.

CALLAGHAN, M. J. et al. Intelligent User Support in Autonomous Remote Experimentation Environments. **IEEE Transactions on Industrial Electronics**, New York, USA, v.55, n.6, p.2355–2367, June 2008.

CALLAGHAN, M. J. et al. Hybrid Remote/Virtual Laboratories with Virtual Learning Environment Integration. In: REMOTE ENGINEERING AND VIRTUAL INSTRUMENTATION CONFERENCE (REV2010), 2010, Stockholm, Sweden. **Proceedings...** [S.l.]:Universität Kassel, 2010. p.238–245.

CARRERAS, M. A. M. et al. Colab: a platform design for collaborative learning in virtual laboratories. **International Federation for Information Processing (IFIP)**, Tokyo, Japan, v.177, n.12, p.95–109, 2005.

CARROLL, J. M. et al. Notification and Awareness: synchronizing task-oriented collaborative activity. **International Journal of Human-Computer Studies**, Netherlands, v.58, n.5, p.605–632, May 2003.

CASINI, M. et al. **A Matlab-based Remote Lab for Multi-Robot Experiments**. [S.l.: s.n.], 2009. Disponível em: <<http://www.dii.unisi.it/~anto/papers/ACE09.pdf>>. Acesso em: 3 dez. 2010.

CASINI, M.; PRATTICHIZZO, D.; VICINO, A. The Automatic Control Telelab: a web-based technology for distance learning. **IEEE Control Systems Magazine**, New York, USA, v.24, n.3, p.36–44, June 2004.

CASINI, M.; PRATTICHIZZO, D.; VICINO, A. A Student Control Competition through a Remote Robotics Lab. **IEEE Control Systems Magazine**, New York, USA, v.25, n.1, p.56–59, Feb. 2005.

CHASTINE, J. W.; ZHU, Y.; PRESTON, J. A. A Framework for Inter-referential Awareness in Collaborative Environments. In: INTERNATIONAL CONFERENCE ON COLLABORATIVE COMPUTING: NETWORKING, APPLICATIONS AND WORKSHARING (COLCOM), 2006, Atlanta, USA. **Proceedings...** [S.l.: s.n.], 2006. p.1–5.

CHILDERS, L. et al. Designing Experiment Agnostic Remote Laboratories. In: INTERNATIONAL IMMERSIVE PROJECTION TECHNOLOGY WORKSHOP, 4., 2000, Ames, USA. **Proceedings...** [S.l.: s.n.], 2000. p.1–9.

CHOMSKY, N. **Language and Mind**. Cambridge, USA: Cambridge University Press, 2006. 208p.

CLAROLINE.NET. **Claroline - Open Source eLearning and eWorking platform Website**. Disponível em: <<http://www.claroline.net>>. Acesso em: 9 dez. 2010.

COLLABORATIVE LABORATORIES (COLAB). **CoLab Website**. Disponível em: <<http://colab.edte.utwente.nl>>. Acesso em: 12 fev. 2011.

COLE, M. **Cultural Psychology: a once and future discipline**. Cambridge, USA: Belknap Press of Harvard University Press, 1998. 416p.

COMEDI. **Comedi** - linux control and measurement device interface website. Disponível em: <<http://www.comedi.org>>. Acesso em: 10 mar. 2011.

ADOBE SYSTEMS INCORPORATED. **Adobe Connect Software Website**. Disponível em: <<http://www.adobe.com/de/products/connect>>. Acesso em: 20 fev. 2011.

COOPER, M. The Challenge of Practical Work in a eUniversity - Real, Virtual and Remote Experiments. In: INFORMATION SOCIETY TECHNOLOGIES CONFERENCE, 2000, Nice, France. **Proceedings...** [S.l.]:IST, 2000. p. 34–40.

CORTER, J. E. et al. Remote versus Hands-on Labs: a comparative study. In: ANNUAL FRONTIERS IN EDUCATION, 34., 2004, Savannah, USA. **Proceedings...** [S.l.: s.n.], 2004. p.17–21.

COUGAAR. **Cougaar Agent Architecture Website**. Disponível em: <<http://cougaar.org>>. Acesso em: 9 fev. 2011.

PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO TECGRAF (CSGRID). **CSGrid Website**. Disponível em: <<http://www.tecgraf.puc-rio.br/csbase/csgrid>>. Acesso em: 19 fev. 2011.

DEWEY, J. **Experience and Education**. New York, USA: Free Press, 1938. 96p.

DOUGIAMAS, M.; TAYLOR, P. MOODLE: using learning communities to create an open source course management system. In: WORLD CONFERENCE ON EDUCATIONAL MULTIMEDIA, HYPERMEDIA AND TELECOMMUNICATIONS, 2003, Chesapeake, Australia. **Proceedings...** [S.l.: s.n.], 2003. p.171–178.

- DYNAMIT. **Dynamische Systeme Website**. Disponível em: <<http://www.esr.ruhr-uni-bochum.de/uvm>>. Acesso em: 14 jul. 2010.
- ECOSPACE. **ECOSPACE IP Project - eProfessional Collaborative Workspace Website**. Disponível em: <<http://www.ip-ecospace.org>>. Acesso em: 18 fev. 2011.
- EASY JAVA SIMULATIONS (EJS). **Easy Java Simulations Website**. Disponível em: <<http://www.um.es/fem/Ejs>>. Acesso em: 15 fev. 2011.
- ELIPSE SOFTWARE. **Elipse Software Website**. Disponível em: <<http://www.elipse.com.br>>. Acesso em: 10 mar. 2011.
- ELLIS, C. A.; GIBBS, S. J.; REIN, G. L. Groupware - Some Issues and Experiences. **Communications of the ACM**, New York, USA, v.34, n.1, p.38–58, Jan. 1991.
- ELVESÆTER, B. et al. **Interoperability of Enterprise Software and Applications**. London, UK: Springer Verlag, 2006. p.409–420.
- ENLUND, N. Being Virtually There - reality and presence in mediated learning. In: INTERNATIONAL CONFERENCE ON TELECOMMUNICATIONS FOR EDUCATION AND TRAINING, 2001, Prague, Czech Republic. **Proceedings...** [S.l.: s.n.], 2001. p.iv–ix.
- ENTROPIAUNIVERSE. **Entropia Universe Website**. Disponível em: <<http://www.entropiauniverse.com>>. Acesso em: 13 fev. 2011.
- ERGOLIST. **ErgoList Website**. Disponível em: <<http://www.labiutil.inf.ufsc.br/ergolist>>. Acesso em: 12 fev. 2011.
- ESPINDOLA, D. B. et al. Using Mixed Reality in the Visualization of Maintenance Processes. In: IFACWORKSHOP ON ADVANCED MAINTENANCE ENGINEERING, SERVICES AND TECHNOLOGY, 1., 2010, Lisboa, Portugal. **Proceedings...** [S.l.]:IFAC, 2010. p.35–40.
- ESQUEMBRE, F. Easy Java Simulations: a software tool to create scientific simulations in java. **Computer Physics Communications**, Amsterdam, Netherlands, v.156, n.2, p.199–204, Jan. 2004.
- FACEBOOK. **Facebook Social Network Website**. Disponível em: <<http://www.facebook.com>>. Acesso em: 10 fev. 2011.
- FAUST, M.; BRUNS, F. W. Mixed Reality Web Service: air through the internet. In: INTERNATIONAL CONFERENCE ON TECHNOLOGY-ENHANCED LEARNING, 2003, Milano, Italy. **Proceedings...** [S.l.: s.n.], 2003.
- FAUST, M.; YOO, Y.-H. Haptic Feedback in Pervasive Games. In: INTERNATIONAL WORKSHOP ON PERSVASIVE GAMING APPLICATIONS, 3., 2006, Dublin, Ireland. **Proceedings...** [S.l.: s.n.], 2006.

FEISEL, D. L.; ROSA, A. J. The Role of the Laboratory in the Undergraduate Engineering Education. **Journal of Engineering Education**, Washington, USA, v.94, n.1, p.121–130, Jan. 2005.

FESTO. **FESTO Didactic Website**. Disponível em: <<http://www.festo-didactic.com>>. Acesso em: 10 mar. 2011.

FJELD, M. et al. Physical and Virtual Tools: activity theory applied to the design of groupware. **Special Issue of Computer Supported Cooperative Work (CSCW): Activity Theory and the Practice of Design**, Netherlands, v.11, n.1-2, p.153–180, 2002.

MICROSOFT CORPORATOIN. **Microsoft's Flight Simulator Website**. Disponível em: <<http://www.microsoft.com/games/flightimulatorx>>. Acesso em: 15 fev. 2011.

FRANK, G. A.; HELMS, R. F.; VOOR, D. Determining the Right Mix of Live, Virtual, and Constructive Training. In: INTERSERVICE/INDUSTRY TRAINING SYSTEMS AND EDUCATION CONFERENCE, 2000, Orlando, USA. **Proceedings...** [S.l.: s.n.], 2000.

FREESWITCH. **FreeSwitch - Cross-Platform Scalable Free Multi-Protocol Soft Switch Website**. Disponível em: <<http://www.freeswitch.org>>. Acesso em: 23 fev. 2011.

GARCÍA-ZUBÍA, J. et al. Developing a Second-Life-based Remote Lab over the WebLab-Deusto architecture. In: REMOTE ENGINEERING AND VIRTUAL INSTRUMENTATION CONFERENCE (REV2010), 2010, Stockholm, Sweden. **Proceedings...** [S.l.]:Universitat Kassel, 2010. p.171–176.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL. Grupo de Controle Automaao e Robotica (GCAR-EAD). **GCAR-EAD MOODLE da Engenharia Eletrica da UFRGS**. Disponível em: <<http://ensino.ece.ufrgs.br/moodle>>. Acesso em: 12 fev. 2011.

GNU OPERATING SYSTEM. **GNU ('GNU is Not Unix') Free Operating System General Public License Website**. Disponível em: <<http://www.gnu.org/licenses/gpl.html>>. Acesso em: 10 mar. 2011.

GOMES, L.; GARCÍA-ZUBÍA, J. **Advances on Remote Laboratories and e-Learning Experiences**. Bilbao, Spain: University of Deusto, 2007. 309p.

GREER, J. E.; MCCALLA, G. I. **Student Modelling: the key to individualized knowledge-based instruction**. Amsterdam, Netherlands: Springer Verlag, 1994. 383p.

GROUPWARE@LES (AULANET). AulaNet Groupware Website. Disponível em: <<http://groupware.les.inf.puc-rio.br>>. Acesso em: 9 dez. 2010.

GRUDIN, J. Why CSCW fail? Problems in Design and Evaluation of Organizational Interfaces. In: CONFERENCE ON COMPUTER SUPPORTED COOPERATIVE WORK, 2., 1988, Portland, USA. **Proceedings...** [S.l.: s.n.], 1988. p.85–93.

- GRUDIN, J. Groupware and Social Dynamics: eight challenges for developers. **Communications of the ACM**, New York, USA, v.37, n.1, p.92–105, Jan. 1994.
- GUTIERREZ, M.; OTT, R.; THALMANN, D.; VEXO, F. Mediators: virtual haptic interfaces for tele-operated robots. In: IEEE INTERNATIONAL WORKSHOP ON ROBOT AND HUMAN INTERACTIVE COMMUNICATION, 13., 2004, Okayama, Japan. **Proceedings...** [S.l.]:IEEE, 2004. p.515–520.
- GUTWIN, C.; GREENBERG, S. A Descriptive Framework of Workspace Awareness for Real-Time Groupware. **Computer Supported Cooperative Work**, Norwell, USA, v.11, n.3-4, p.411–446, Sep. 2002.
- HALL, T. et al. The Visitor as Virtual Archaeologist: explorations in mixed reality technology to enhance educational and social interaction in the museum. In: CONFERENCE ON VIRTUAL REALITY, ARCHEOLOGY, AND CULTURAL HERITAGE, 2001, Glyfada, Greece. **Proceedings...** [S.l.: s.n.], 2001. p.91–96.
- HENDAOU, A.; LIMAYEM, M.; THOMPSON, C. W. 3D Social Virtual Worlds: research issues and challenges. **IEEE Internet Computing Magazine**, New York, USA, v.12, n.1, p.88–92, Jan./Feb. 2008.
- HINE, N. A. et al. Institutional Factors Governing the Deployment of Remote Experiments: lessons from the rexnet project. In: REMOTE ENGINEERING AND VIRTUAL INSTRUMENTATION CONFERENCE (REV2007), 2007, Porto, Portugal. **Proceedings...** [S.l.]:Universität Kassel, 2007. p.1–5.
- HIPPO OPENSIM VIEWER. **Hippo OpenSim Viewer Website**. Disponível em: <<http://mjm-labs.com/viewer>>. Acesso em: 8 fev. 2011.
- HUIJUN, L.; AIGUO, S. Virtual-Environment Modeling and Correction for Force-Reflecting Teleoperation With Time Delay. **IEEE Transactions on Industrial Electronics**, New York, USA, v.54, n.2, p.1227–1233, Apr. 2007.
- ILABS. **MIT iLabs: internet access to real labs - anywhere, anytime website**. Disponível em: <<http://icampus.mit.edu/ilabs>>. Acesso em: 14 fev. 2011.
- INSTANT MESSAGING VIRTUAL UNIVERSE (IMVU). **Instant Messaging Virtual Universe Website**. Disponível em: <<http://www.imvu.com>>. Acesso em: 13 fev. 2011.
- ICS TRIPLEX ISAGRAF INC. **ICS Triplex ISaGRAF Inc. - leading IEC 61131 and IEC 61499 Software Website**. Disponível em: <<http://www.isagraf.com>>. Acesso em: 12 fev. 2011.
- JAVA AGENT DEVELOPMENT FRAMEWORK (JADE). **Java Agent Development Framework Website**. Disponível em: <<http://jade.tilab.com>>. Acesso em: 18 fev. 2011.
- JASON. **JASON - a Java-based interpreter for an extended version of AgentSpeak Website**. Disponível em:

<<http://jason.sourceforge.net/Jason/Jason.html>>. Acesso em: 9 fev. 2011.

JENNINGS, N. R. On Agent-Base Software Engineering. **Artificial Intelligence**, Amsterdam, Netherlands, v.117, n.2, p.277–296, Mar. 2000.

JESS. **JESS - the Rule Engine for the Java Platform Website**. Disponível em: <<http://www.jessrules.com>>. Acesso em: 21 fev. 2011.

JOOLINGEN, W. R. van; JONG, T. de. **Authoring Tools for Advanced Technology Educational Software**: toward cost-effective production of adaptive, interactive, and intelligent educational software. Dordrecht, Netherlands: Kluwer Academic Publishers, 2003. p.1–31.

JVOICEBRIDGE. **Java Voice Bridge Website**. Disponível em: <<http://jvoicebridge.dev.java.net>>. Acesso em: 14 fev. 2011.

KANEVA. **Kaneva Website**. Disponível em: <<http://www.kaneva.com>>. Acesso em: 13 fev. 2011.

KAPPE, F.; GUETL, C. Enhancements of the RealXtend Framework to Build a Virtual Conference Room for Knowledge Transfer and Learning Purposes. In: **WORLD CONFERENCE ON EDUCATIONAL MULTIMEDIA, HYPERMEDIA AND TELECOMMUNICATIONS (ED-MEDIA)**, 2009, Honolulu, Hawaii, USA. **Proceedings...** [S.l.: s.n.], 2009.

KAPTELININ, V. **Context and Consciousness**. Cambridge, USA: MIT Press, 1995. p.45–68.

KARNOPP, D. C.; ROSENBERG, R. C. **System Dynamics**: a unified approach. New York, USA: Wiley-Interscience, 1990. 528p.

KEMP, J.; KABUMPO, S. J. Putting a Second Life 'Metaverse' Skin on Learning Management Systems. In: **SECOND LIFE EDUCATION WORKSHOP AT SECOND LIFE COMMUNITY CONVENTION**, 2006, San Francisco, USA. **Proceedings...** [S.l.: s.n.], 2006. p.13–18.

KHEIR, N. A. et al. Control Systems Engineering Education. **Automatica**, Amsterdam, Netherlands, v.32, n.2, p.147–166, Feb. 1996.

KIRRIEMUIR, J. The Second Life of UK Academics. **Ariadne Magazine**, Bath, UK, n.53, Oct. 2007.

KIRSCHNER, P. A.; SWELLER, J.; CLARK, R. E. Why Minimal Guidance During Instruction Does Not Work: an analysis of the failure of constructivist, discovery, problem-based, experiential, and inquiry-based teaching. **Educational Psychologist**, Philadelphia, USA, v.41, n.2, p.75–86, 2006.

KOKU, A. B.; KAYNAK, O. An Internet-assisted Experimental Environment Suitable for the Reinforcement of Undergraduate Teaching of Advanced Control Techniques. **IEEE Transactions on Education**, New York, USA, v.44, n.1, p.24–28, Feb. 2001.

- KON, F. et al. 2K: a distributed operating system for dynamic heterogeneous environments. In: IEEE INTERNATIONAL SYMPOSIUM ON HIGH PERFORMANCE DISTRIBUTED COMPUTING, 9., 2000, Pittsburgh, USA. **Proceedings...** [S.l.]:IEEE, 2000. p.201–208.
- KON, F. et al. Monitoring, Security, and Dynamic Configuration with the DynamicTAO Reflective ORB. In: IFIP/ACM INTERNATIONAL CONFERENCE ON DISTRIBUTED SYSTEMS PLATFORMS, 2000, Secaucus, USA. **Proceedings...** [S.l.]:ACM, 2000. p.121–143.
- KREIJNS, K.; KIRSCHNER, P. A.; JOCHEMS, W. Identifying the Pitfalls for Social Interaction in Computer-Supported Collaborative Learning Environments: a review of the research. **Computer in Human Behavior**, Amsterdam, Netherlands, v.19, n.3, p.335–353, May 2003.
- KUMAR, V. S. Computer-Supported Collaborative Learning: issues for research. In: ANNUAL GRADUATE SYMPOSIUM ON COMPUTER SCIENCE, 8., 1996, Saskatoon, Canada. **Proceedings...** [S.l.: s.n.], 1996. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1.50.6693&rep=rep1&type=pdf>>. Acesso em: 13 dez. 2010.
- NATIONAL INSTRUMENTS. **National Instruments LabVIEW Software Website**. Disponível em: <<http://www.ni.com/labview>>. Acesso em: 10 mar. 2011.
- LASO-BALLESTEROS, I.; KARLSSON, L. **CWE'06 Conference Report**. [S.l.: s.n.], 2006. Disponível em: <http://www.difac.net/Download/CWE06_Report_Final_eBook.pdf>. Acesso em: 23 nov. 2010.
- LEE, J.-S.; HSU, P.-L. Remote Supervisory Control of the Human-in-the-Loop System by Using Petri Nets and Java. **IEEE Transactions on Industrial Electronics**, New York, USA, v.50, n.3, p.431–439, June 2003.
- LEHTINEN, E. **Unraveling Basic Components and Dimensions of Powerful Learning Environments**. Amsterdam, Netherlands: Elsevier, 2003. p.35–53.
- LEIDNER, D.; JARVENPAA, S. The Use of Information Technology to Enhance Management School Education: a theoretical view. **Management Information Systems (MIS) Quarterly**, Minneapolis, USA, p.265–292, Sept. 1995.
- LESSER, V. Cooperative Multi-Agent Systems: a personal view of the state of the art. **IEEE Transactions on Knowledge and Data Engineering**, New York, USA, v.11, n.1, p.133–142, Jan./Feb. 1999.
- OPEN METAVERSE FOUNDATION. **LibOpenMetaverse at Open Metaverse Foundation Website**. Disponível em: <<http://www.openmetaverse.org/projects/libopenmetaverse>>. Acesso em: 23 fev. 2011.
- LIMA, M. J. de et al. **CSGrid: um sistema para integração de aplicações em grades computacionais**. [S.l.: s.n.], 2005. Disponível em: <<http://www.tecgraf.puc->

rio.br/publications/artigo_2005_csgrid_integracao_aplicacoes.pdf>. Acesso em: 3 jul. 2010.

LINKEDIN. LinkedIn Professional Network Website. Disponível em: <<http://www.linkedin.com>>. Acesso em: 10 fev. 2011.

LUA. The Programming Language Lua Website. Disponível em: <<http://www.lua.org>>. Acesso em: 20 fev. 2011.

LUM, W. Y.; LAU, F. A Context-Aware Decision Engine for Content Adaptation. **IEEE Pervasive Computing**, IEEE, New York, USA, v.1, n.3, p.41–49, Dec. 2002.

MA, J.; NICKERSON, J. V. Hands-on, Simulated, and Remote Laboratories: a comparative literature review. **ACM Computing Surveys**, New York, USA, v.38, n.3, Sep. 2006.

MAGEE CAMPUS. Magee Campus Second Life URL. Disponível em: <<http://slurl.com/secondlife/University%20Ulster%20Magee/249/161/38>>. Acesso em: 16 fev. 2011.

MAGOULAS, G. D.; PAPANIKOLAOU, Y.; GRIGORIADOU, M. Adaptive Web-based Learning: accommodating individual differences through system's adaptation. **British Journal of Educational Technology (BJET)**, London, UK, v.34, n.4, p.511–527, Sep. 2003.

MAPLE SOFTWARE. Maple Website. Disponível em: <<http://www.maplesoft.com>>. Acesso em: 10 mar. 2011.

MARTYN, M. Clickers in the Classroom: an active learning approach. **Educause Quarterly**, Washinton, USA, n.2, p.71–74, 2007.

MARVEL. Virtual laboratory in Mechatronics Access to Remote and Virtual e-Learning Website. Disponível em: <<http://www.marvel.uni-bremen.de>>. Acesso em: 15 jul. 2010.

MASLOW, A. A Theory of Human Motivation. **Psychological Review**, Washington, USA, v.50, n.4, p.370–396, 1943.

MATHWORKS INC. MathWorks - MATLAB and Simulink for Technical Computing Website. Disponível em: <<http://www.mathworks.com>>. Acesso em: 10 mar. 2011.

MAXIMA SOFTWARE. Maxima Website. Disponível em: <<http://maxima.sourceforge.net>>. Acesso em: 10 mar. 2011.

MAYO, M. J. Games for Science and Engineering Education. **ACM Communications**, New York, USA, v.50, n.7, p.30–35, July 2007.

MEASUREMENT COMPUTING (MC). USB Data Acquisition Device at Measurement Computing Website. Disponível em:

<<http://www.mccdaq.com/usb-data-acquisition/USB-1208FS.aspx>>. Acesso em: 10 mar. 2011.

MICHAEL, D. R.; CHEN, S. L. **Serious Games: games that educate, train, and inform.** [S.l.]: Thomson Course Technology, 2005. 287p.

MICHAELIDES, I.; ELEFTHREIOU, P.; MÜLLER, D. A Remotely Accessible Solar Energy Laboratory - A Distributed Learning Experience. In: INTERNATIONAL CONFERENCE ON REMOTE ENGINEERING AND VIRTUAL INSTRUMENTATION, 2004, Vienna, Austria. **Proceedings...** [S.l.]:Universität Kassel, 2004.

MILGRAM, P. et al. Merging Real and Virtual Worlds. In: IMAGINA, 1995, Montecarlo. **Proceedings...**[S.l.: s.n.], 1995. p.221–230.

MILGRAM, P.; KISHINO, F. A Taxonomy of Mixed-reality Visual Displays. **Institute of Electronics, Information and Communication Engineers (IECE) Transactions on Information and Systems**, Tokyo, Japan, v.E77-D, n.12, p.1321–1329, 1994.

MIYAKE, N. Constructive Interaction and the Iterative Process of Understanding. **Cognitive Science Journal**, Amsterdan, Netherlands, v.10, n.2, p.151–177, Apr./June 1986.

MÜLLER, D. et al. Mixed Reality Learning Spaces for Collaborative Experimentation: a challenge for engineering education and training. **International Journal of Online Engineering (iJOE)**, [S.l.], v.3, n.4, p.15–19, 2007.

MÜLLER, D.; FERREIRA, J. M. MARVEL: a mixed-reality learning environment for vocational training in mechatronics. In: TECHNOLOGY ENHANCED LEARNING INTERNATIONAL CONFERENCE, 2004, Milano, Italy. **Proceedings...**[S.l.: s.n.], 2004. p.65–72.

MÜLLER, D.; SCHAF, F. M. A Low Cost Learning Environment for Collaborative Engineering. In: INTERNATIONAL CONFERENCE ON REMOTE ENGINEERING AND VIRTUAL INSTRUMENTATION (REV 2009), 2009, Bridgeport, USA. **Proceedings...** [S.l.]:Universität Kassel, 2009. p.304–308.

MONO FRAMEWORK. **Mono Framework Website.** Disponível em: <<http://www.mono-project.com>>. Acesso em: 16 fev. 2011.

MOODLE. **MOODLE - open-source community-based tools for learning Website.** Disponível em: <<http://www.moodle.org>>. Acesso em: 9 dez. 2010.

MOSAIC. **MOSAIC Network Project Website.** Disponível em: <<http://www.mosaic-network.org>>. Acesso em: 19 fev. 2011.

MODELLINGSPACE (MS). **ModellingSpace Website.** Disponível em: <<http://www.modellingspace.net>>. Acesso em: 12 fev. 2011.

MICROSOFT SQL SERVER (MSSQL). **Microsoft's SQL Server Website.** Disponível em: <<http://www.microsoft.com/sqlserver>>. Acesso em: 12 fev. 2011.

MYSQL. MySQL open-source Database Website. Disponível em:
<<http://www.mysql.com>>. Acesso em: 12 fev. 2011.

NEALE, D. C.; CARROLL, J. M.; ROSSON, M. B. Evaluating Computer-Supported Cooperative Work: models and frameworks. In: ACM CONFERENCE ON COMPUTER SUPPORTED COOPERATIVE WORK, 2004, Chicago, USA. **Proceedings...** [S.l.]:ACM, 2004. p.112–121.

NOGUEZ, J.; SUCCAR, L. E. A Student Model based on Probabilistic Relational Models. **Lectures Notes in Artificial Intelligence**, Berlin, Germany, v.4511, p.303–308, June 2007.

NOVUS. **Novus - Produtos Eletrônicos Website.** Disponível em:
<<http://www.novus.com.br>>. Acesso em: 7 fev. 2011.

OPEN DYNAMICS ENGINE (ODE). **Open Dynamics Engine Project Website.** Disponível em: <<http://www.ode.org>>. Acesso em: 23 fev. 2011.

OGRE3D. **OGRE - Open Source 3D Graphics Engine Website.** Disponível em:
<<http://www.ogre3d.org>>. Acesso em: 17 fev. 2011.

OPC FOUNDATION. **OPC Foundation Website.** Disponível em:
<<http://www.opcfoundation.org>>. Acesso em: 9 fev. 2011.

OPENID. **OpenID Foundation Website.** Disponível em: <<http://openid.net>>. Acesso em: 17 fev. 2011.

OPENMEETINGS SOFTWARE. **OpenMeetings Website.** Disponível em:
<<http://openmeetings.net>>. Acesso em: 20 fev. 2011.

OPEN WONDERLAND. **Open Wonderland - open source toolkit for creating collaborative 3D virtual worlds Website.** Disponível em:
<<http://www.openwonderland.org>>. Acesso em: 14 fev. 2011.

ORACLE. **MPK20: Sun's virtual workplace project website.** Disponível em:
<<http://labs.oracle.com/projects/mc/mpk20.html>>. Acesso em: 10 jan. 2011.

ORACLE DB. **Oracle Database Website.** Disponível em:
<<http://www.oracle.com/us/products/database>>. Acesso em: 12 fev. 2011.

ORBITEAM. **OrbiTeam BSCW Website.** Disponível em:
<<http://www.bscw.de>>. Acesso em: 19 fev. 2011.

ORDUÑA, P. et al. Designing Experiment Agnostic Remote Laboratories. In: REMOTE ENGINEERING AND VIRTUAL INSTRUMENTATION (REV), 2009, Bridgeport, USA. **Proceedings...** [S.l.]:Universität Kassel, 2009.

ORKUT. **Google Orkut Social Network Website.** Disponível em:
<<http://www.orkut.com>>. Acesso em: 10 fev. 2011.

OPEN SIMULATOR (OS). **Open Simulator Project Website**. Disponível em: <<http://www.opensimulator.org>>. Acesso em: 16 fev. 2011.

OTSUKA, J. L.; ROCHA, H. V. da; BEDER, D. M. A Multi-Agent Formative Assessment Support Model for Learning Management Systems. In: IEEE INTERNATIONAL CONFERENCE ON ADVANCED LEARNING TECHNOLOGIES (ICALT), 7., 2007, Niigata, Japan. **Proceedings...** [S.l.]:IEEE, 2007. p.85–89.

PAYNTER, H. M. **Analysis and Design of Engineering Systems**. Cambridge, USA:MIT Press, 1960.

PEARL. **Practical Experimentation by Accessible Remote Learning Website**. Disponível em: <<http://iet.open.ac.uk/pearl>>. Acesso em: 15 jul. 2010.

PEROZZO, R. F. **Framework para Construção de Sistemas Supervisórios em Dispositivos Móveis**. 2007. 107p. Dissertação (Mestrado em Engenharia) – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2007. Disponível em: <<http://www.lume.ufrgs.br/handle/10183/10700>>. Acesso em: 10 dez. 2010.

PIAGET, J. **Psychology and Epistemology**: towards a theory of knowledge. London, UK: Penguin, 1972.

PIMENTEL, M. et al. Modelo 3C de colaboração para o desenvolvimento de sistemas colaborativos. In: SIMPÓSIO BRASILEIRO DE SISTEMAS COLABORATIVOS, 3., 2006, Natal, Brasil. **Anais...** [S.l.: s.n.], 2006. p.58–67.

PINELLE, D.; GUTWIN, C.; GREENBERG, S. Task Analysis for Groupware Usability Evaluation: modeling shared-workspace tasks with the mechanics of collaboration. **ACM Transactions on Computer-Human Interaction**, New York, USA, v.10, n.4, p.281–311, Dec. 2003.

PLATÃO. **Mênnon**. Texto estabelecido e anotado por John Burnet com Tradução de Maura Iglésias. Rio de Janeiro: PUC-Rio, 2001. 117p.

POHL, M. **Learning to Think, Thinking to Learn**: models and strategies to develop a classroom culture of thinking. Cheltenham, Australia: Hawker Brownlow Education, 1999.

PORTAL MEC. **Repositório de Objetos Educacionais do Ministério da Educação**. Disponível em: <<http://objetoseducacionais2.mec.gov.br>>. Acesso em: 13 dez. 2010.

POSTGRESQL SOFTWARE. **PostgreSQL open-source Database Website**. Disponível em: <<http://www.postgresql.org>>. Acesso em: 12 fev. 2011.

PRINZ, W. et al. ECOSPACE - Towards an Integrated Collaboration Space for eProfessionals. In: INTERNATIONAL CONFERENCE ON COLLABORATIVE

COMPUTING: NETWORKING, APPLICATIONS ANDWORKSHARING (COLCOM), 2006, Atlanta, USA. **Proceedings...** [S.l.: s.n.], 2006. p.1–7.

PROCESSING SOFTWARE. **Processing Website**. Disponível em: <<http://www.processing.org>>. Acesso em: 15 fev. 2011.

QSIMS WIKI. **QSims Wiki Website**. Disponível em: <<http://wiki.qnap.com/wiki/Q-Sims>>. Acesso em: 12 nov. 2010.

QUEMADA, J. et al. A Collaborative Environment Integration Layer for Activity Orientation. In: INTERNATIONAL CONFERENCE ON COLLABORATIVE COMPUTING: NETWORKING, APPLICATIONS ANDWORKSHARING (COLCOM), 2006, Atlanta, USA. **Proceedings...** [S.l.: s.n.], 2006. p.1–6.

APPLE INC. **Apple Quicktime Software Website**. Disponível em: <<http://www.apple.com/quicktime>>. Acesso em: 12 fev. 2011.

RAMAKRISHNAN, V. et al. Development of a Web- Based Control Experiment for a Coupled Tank Apparatus. In: AMERICAN CONTROL CONFERENCE, 2000, Chicago, USA. **Proceedings...** [S.l.]:ACM, 2000. p.4409–4413.

REALXTEND SOFTWARE. **RealXtend - Open Source Platform for interconnected virtual worlds Website**. Disponível em: <<http://www.realxtend.org>>. Acesso em: 17 fev. 2011.

REDDWARF SERVER. **RedDwarf Server Website**. Disponível em: <<http://www.reddwarfserver.org>>. Acesso em: 14 fev. 2011.

REGENBRECHT, H.; BARATOFF, G.; WILKE, W. Augmented Reality Projects in the Automotive and Aerospace Industries. **IEEE Computer Graphics and Applications**, New York, USA, v.25, n.6, p.48–56, Nov./Dec. 2005.

REIS, L. P. **Coordenação em Sistemas Multi-Agente: aplicações na gestão universitária e futebol robótico**. 2003. 451p. Tese (Doutorado em Engenharia) — Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, 2003. Disponível em: <<http://paginas.fe.up.pt/~lpreis/Research.htm>>. Acesso em 27 dez. 2010.

REMOTE EXPERIMENTATION CONSORTIUM LABORATORIES (REXLAB). **Remote Experimentation Consortium Laboratories Website**. Disponível em: <<http://www.rexlab.net>>. Acesso em: 15 jul. 2010.

RFC2326. **RFC 2326 - Real Time Streaming Protocol (RTSP), IETF Specification Website**. Disponível em: <<http://tools.ietf.org/html/rfc2326>>. Acesso em: 10 mar. 2011.

RICHEY, R. C. Reflections on the 2008 AECT Definitions of the Field. **TechTrends**, [S.l.], v.52, n.1, p.24–25, Feb. 2008.

RITZEMA, T.; HARRIS, B. The Use of Second Life for Distance Education. **Journal of Computing Sciences in Colleges**, New York, USA, v.23, n.6, p.110–116, June 2008.

ROGERS, C. Freedom to Learn: a view of what education might become. **Interchange**, Amsterdam, Netherlands, v.1, n.4, p.111–114, 1970.

ROMÁN, M. et al. Gaia: a middleware infrastructure to enable active spaces. **IEEE Pervasive Computing Magazine**, New York, USA, v.1, n.4, p.74–83, Oct./Dec. 2001.

ROMÁN, M.; KON, F.; CAMPBELL, R. Reflective Middleware: from your desktop to your hand. **IEEE Distributed Systems Online, Special Issue on Reflective Middleware**, New York, USA, v.2, n.5, May 2001.

RUMETSHOFER, H.; WÖß, W. XML-based Adaptation Framework for Psychological-driven E-learning Systems. **IEEE Pervasive Computing**, IEEE, New York, USA, v.6, n.4, p.18–29, 2003.

SCARDAMALIA, M.; BEREITER, C. **CSCL: theory and practice of an emerging paradigm**. Hillsdale, USA: Lawrence Erlbaum Associates, 1996. p.249–268.

SCHAF, F. M. **Arquitetura para Ambiente de Ensino de Controle e Automação utilizando Experimentos Remotos de Realidade Mista**. 2006. 207p. Dissertação (Mestrado em Engenharia) – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2006. Disponível em: <<http://www.lume.ufrgs.br/handle/10183/10320>>. Acesso em: 12 jan. 2011.

SCHAF, F. M.; MÜLLER, D. Integrating Tangible and Virtual Construction Kits for Teaching Mechatronics Design. In: **WORKSHOP IN BE-GREIFBARE INTERAKTIONEN**, 2010, Berlin, Germany. **Proceedings...** [S.l.: s.n.], 2010.

SCHAF, F. M.; PEREIRA, C. E. Integrating Mixed Reality Remote Experiments into Virtual Learning Environments using Interchangeable Components. **IEEE Transactions on Industrial Electronics**, New York, USA, v.56, n.12, p.4776–4783, Dec. 2009.

SCHAF, F. M. et al. Collaborative Learning Environment using Distributed Mixed Reality Experiment for Teaching Mechatronics. In: **IFAC SYMPOSIUM ON COST ORIENTED AUTOMATION**, 8., 2007, Habana, Cuba. **Proceedings...** [S.l.]:IFAC, 2007.

SCHAF, F. M.; PEREIRA, C. E.; HENRIQUES, R. V. B. Blended Learning using GCAR-EAD Environment: experiences and application results. In: **IFAC WORLD CONGRESS**, 17., 2008, Seoul, Korea. **Proceedings...** [S.l.]:IFAC, 2008. p.12637–12642.

SCHAFFERS, H. et al. **Mobile Virtual Work: a new paradigm?** London, UK: Springer Verlag, 2006. p.343–367.

SCHEUCHER, T. et al. Collaborative Virtual 3D Environment for Internet-Accessible Physics Experiments. **International Journal of Online Engineering (iJOE)**, Villach, Austria, v.5, n.1, p.65–71, Aug. 2009.

SCHMID, C. Virtual Control Laboratories and Remote Experimentation in Control Engineering. In: **EAEIE ANNUAL CONFERENCE ON INNOVATIONS IN**

EDUCATION FOR ELECTRICAL AND INFORMATION ENGINEERING (EIE), 11., 2000, Ulm, Germany. **Proceedings...** [S.l.:s.n.], 2000.

SCHMIDT, D. C.; LEVINE, D. L.; MUNGEE, S. The Design of the TAO Real-Time Object Request Broker. **Computer Communications**, Amsterdam, Netherlands, v.21, n.4, p.294–324, Apr. 1998.

SCHROEDER, R.; HUXOR, A.; SMITH, A. Activeworlds: geography and social interaction in virtual reality. **Futures**, Rotterdam, Netherlands, v.33, n.7, p.569–587, Sep. 2001.

SCICOSLAB SOFTWARE. **ScicosLab Website**. Disponível em:
<<http://www.scicoslab.com>>. Acesso em: 10 mar. 2011.

SCIENCESIM. **ScienceSim Website**. Disponível em:
<<http://www.sciencesim.com>>. Acesso em: 17 fev. 2011.

SHARABLE CONTENT OBJECT REFERENCE MODEL (SCORM). **SCORM dot Com Technical Explanation on SCORM**. Disponível em:
<<http://scorm.com/scorm-explained/technical-scorm>>. Acesso em: 15 dez. 2010.

SCUTELNICU, A. et al. Integrating JADE Agents into MOODLE. In: INTERNATIONAL WORKSHOP ON INTELLIGENT AND ADAPTIVE WEB-BASED EDUCATIONAL SYSTEMS, 2007, Hiroshima, Japan. **Proceedings...** [S.l.: s.n.], 2007. p.215–220.

SEMANTIC AGENT. **SemanticAgent - Next Generation of Cognitive Agents based on SemanticWeb TechnologiesWebsite**. Disponível em:
<<http://code.google.com/p/semanticagent>>. Acesso em: 9 fev. 2011.

SERCE, F. C.; YILDIRIM, S. A Web-Based Synchronous Collaborative Review Tool: a case study of an on-line graduate course. **Journal of Educational Technology and Society**, Athabasca, Canada, v.9, n.2, p.166–177, Apr. 2006.

SESAM. **SeSAM - Multi-Agent Simulation Environment Website**. Disponível em:
<<http://www.simsesam.de>>. Acesso em: 9 fev. 2011.

SIMQUEST. **SimQuest: an alternative way of learning website**. Disponível em:
<<http://www.simquest.nl>>. Acesso em: 14 dez. 2010.

SIMQUEST WIKI. **SimQuest Edutech Wiki Website**. Disponível em:
<<http://edutechwiki.unige.ch/en/SimQuest>>. Acesso em: 14 dez. 2010.

SKINNER, B. F. Is it Behaviorism? **Behavioral and Brain Sciences**, Cambridge, USA, v.9, n.4, p.716–716, 1986.

SKYPE SOFTWARE. **Skype - Internet Free Calls Website**. Disponível em:
<<http://www.skype.com>>. Acesso em: 8 fev. 2011.

LINDEN RESEARCH INC. **Second Life Website**. Disponível em:
<<http://www.secondlife.com>>. Acesso em: 15 fev. 2011.

SLINEY, A.; MURPHY, D. JDoc: a serious game for medical learning. In: INTERNATIONAL CONFERENCE ON ADVANCES IN COMPUTER-HUMAN INTERACTION (ACHI), 1., 2008, Sainte Luce, Martinique. **Proceedings...** [S.l.: s.n.], 2008. p.131–136.

SLOODLE. **SLOODLE - Simulation Linked Object Oriented Dynamic Learning Environment Website**. Disponível em: <<http://www.sloodle.org>>. Acesso em: 15 fev. 2011.

SECOND LIFE WIKI. **Second Life Wiki Website**. Disponível em:
<<http://wiki.secondlife.com/wiki>>. Acesso em: 23 fev. 2011.

SMALL WORLDS SOFTWARE. **Small Worlds Website**. Disponível em:
<<http://www.smallworlds.com>>. Acesso em: 13 fev. 2011.

SMITH, K. A. The Craft of Teaching Cooperative Learning: an active learning strategy. In: CONFERENCE FRONTIERS IN EDUCATION, 1989, Pittsburg, USA. **Proceedings...** [S.l.: s.n.], 1989. p.188–193.

UNIVERSITY OF MICHIGAN. **Science of Collaboratories at University of Michigan Website**. Disponível em:
<<http://www.scienceofcollaboratories.org>>. Acesso em: 17 jan. 2011.

STAHL, G.; KOSCHMANN, T.; SUTHERS, D. **The Cambridge Handbook of the Learning Sciences**. Cambridge, UK: Cambridge University Press, 2006. p.209–452.

STEFFE, L. P.; GALE, J. **Constructivism in Education**. Hillsdale, USA: Lawrence Erlbaum, 1995. 575p.

STEPHENSON, N. **Snow Crash**. New York, USA: Spectra, 1992. 440p.

STONE, P.; VELOSO, M. Multiagent Systems: a survey from a machine learning perspective. **Autonomous Robots**, Netherlands, v.8, n.3, p.345–383, June 2000.

STUDIP. **Stud.ip Portal**. Disponível em: <<http://www.studip.de>>. Acesso em: 9 dez. 2010.

MASSACHUSETTS INSTITUTE OF TECHNOLOGY. **TEAL Tour and TEALsim at MIT Website**. Disponível em:
<http://web.mit.edu/8.02t/www/802TEAL3D/teal_tour.htm>. Acesso em: 14 fev. 2011.

TELEDUC SOFTWARE. **TELEDUC - Educação à Distância Website**. Disponível em: <<http://www.teleduc.org.br>>. Acesso em: 9 dez. 2010.

TELEPLACE SOFTWARE. **Teleplace - Virtual Worlds Collaboration Solutions Website**. Disponível em: <<http://www.teleplace.com>>. Acesso em: 18 fev. 2011.

THESIMS. **Electronic Arts The Sims Website**. Disponível em: <<http://thesims.ea.com>>. Acesso em: 15 fev. 2011.

TIMM, M. I. **Elaboração de Projetos como Estratégia Pedagógica para o Ensino de Engenharia (Curso à Distância de Projeto no Modelo E-learning-by-doing)**. 2005. 313p. Tese (Doutorado em Informática na Educação) – Programa de Pós-Graduação em Informática na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2005. Disponível em: <<http://hdl.handle.net/10183/13747>>. Acesso em: 18 mar. 2011.

TUTTAS, J.; WAGNER, B. Distributed Online Laboratories. In: INTERNATIONAL CONFERENCE ON ENGINEERING EDUCATION, 2001, Oslo, Norway. **Proceedings..** . [S.l.]:Universitat Kassel, 2001. p.8D1–7–8D1–11.

TWITTER. **Twitter Short Blog Service Website**. Disponível em: <<http://www.twitter.com>>. Acesso em: 10 fev. 2011.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL. **MOODLE Institucional da UFRGS**. Disponível em: <<https://moodleinstitucional.ufrgs.br>>. Acesso em: 12 fev. 2011.

VASTPARK SOFTWARE. **VastPark - Virtual Enterprise Software Website**. Disponível em: <<http://www.vastpark.com>>. Acesso em: 18 fev. 2011.

VCLAB. **Virtual Control Lab Website**. Disponível em: <<http://www.atp.ruhr-uni-bochum.de/VCLab>>. Acesso em: 13 dez. 2010.

VYGOTSKY, L. S. **Mind in Society: the development of higher psychological processes**. Cambridge, USA: Harvard University Press, 1978. 159p.

WATSON, J. B. Psychology as the Behaviorist Views it. **Psychological Review**, Washington, USA, v.20, p.158–177, 1913.

WATSON, K. Utilization of Active and Cooperative Learning in EE courses: three classes and the results. In: CONFERENCE FRONTIERS IN EDUCATION, 1995, Atlanta, USA. **Proceedings..** . [S.l.: s.n.], 1995. v.2, p.3c2.1 – 3c2.6.

WEB 3D CONSORTIUM. **Web 3D Consortium - X3D for Developers Website**. Disponível em: <<http://www.web3d.org/x3d>>. Acesso em: 10 mar. 2011.

WEB ASSIGN INC. **WebAssign - Online Homework and Grading Website**. Disponível em: <<http://www.webassign.net>>. Acesso em: 9 dez. 2010.

WEBAULA. **WebAula - Educao sem fronteiras Website**. Disponível em: <<http://www.webaula.com.br>>. Acesso em: 9 dez. 2010.

WEBDAV. **Web-based Distributed Authoring and Versioning Resources Website**. Disponível em: <<http://webdav.org>>. Acesso em: 17 fev. 2011.

WALLENBERG GLOBAL LEARNING NETWORK. **Wallenberg Global Learning Network Website**. Disponível em: <<http://www.wgln.org>>. Acesso em: 15 dez. 2010.

WIKIPEDIA. **Berkeley Software Distribution (BSD) License at Wikipedia**. Disponível em: <http://en.wikipedia.org/wiki/BSD_licenses>. Acesso em: 10 mar. 2011.

WIKIPEDIA. **Wikipedia article on 3D Modelling Software**. Disponível em: <http://en.wikipedia.org/wiki/3d_Modelling_Software>. Acesso em: 15 fev. 2011.

WILEY, D. A. **The Instructional Use of Learning Objects**. Salt Lake City, USA: University of Utah, 2000. v.2830, n.435, p.1–35. Disponível em: <<http://reusability.org/read/chapters/wiley.doc>>. Acesso em: 22 dez. 2010.

WOOLDRIDGE, M. **An Introduction to Multi-Agent Systems**. Hoboken, USA: John Wiley and Sons, 2002. 348p.

WORLD OF WARCRAFT WIKI. **World of Warcraft Wiki Website**. Disponível em: <http://www.wowpedia.org/World_of_Warcraft>. Acesso em: 15 fev. 2011.

XML-RPC.COM. **XML-RPC Specification at XML-RPC.com Website**. Disponível em: <<http://www.xmlrpc.com/spec>>. Acesso em: 10 mar. 2011.

YANKELOVICH, N. et al. Porta-Person: telepresence for the connected conference room. In: CONFERENCE ON HUMAN-COMPUTER INTERACTION, 2007, San Jose, USA. **Proceedings**. . . [S.l.: s.n.], 2007.

YOO, Y.-H.; BRUNS, F. W. Realtime Collaborative Mixed Reality Environment with Force Feedback. In: IFAC SYMPOSIUM ON COST ORIENTED AUTOMATION, 7., 2004, Gatineau, Canada. **Proceedings**. . . [S.l.]:IFAC, 2004. p.153–158.

ZEILMANN, R. P. et al. Web-based Control Experiment on a Foundation Fieldbus Pilot Plant. In: IFAC INTERNATIONAL CONFERENCE ON FIELDBUS SYSTEMS AND THEIR APPLICATIONS, 5., 2003, Aveiro, Portugal. **Proceedings**. . . [S.l.]:IFAC, 2003. p.325–330.

ZHOU, Y.; EVENS, M. W. A Practical Student Model in an Intelligent Tutoring System. In: IEEE INTERNATIONAL CONFERENCE ON TOOLS WITH ARTIFICIAL INTELLIGENCE, 11., 1999, Chicago, USA. **Proceedings**. . . [S.l.]:IEEE, 1999. p.13–18.

ZYDA, M. From Visual Simulation to Virtual Reality to Games. **IEEE Computer**, New York, USA, v.38, n.9, p.25–32, Sep. 2005.