

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

**Um Sistema de Alertas Inteligentes  
para Ambientes de Ensino na Internet**

por

DANIELA LEAL MUSA

Dissertação submetida à avaliação, como requisito parcial para  
a obtenção do grau Mestre em  
Ciência da Computação

Prof. Dr. José Palazzo Moreira de Oliveira  
Orientador

Porto Alegre, maio de 2001.

**CIP – CATALOGAÇÃO NA PUBLICAÇÃO**

Musa, Daniela Leal

Um Sistema de Alertas Inteligentes para Ambientes de Ensino na Internet/por Daniela Leal Musa – Porto Alegre: PPGC da UFRGS, 2001.

116 f.: il.

Dissertação (Mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR – RS, 2001. Orientador: Oliveira, José Palazzo Moreira de.

1. Educação a distância. 2. Alertas Inteligentes 3. Avaliação Remota 4. Monitoramento de atividades na Web 5. Agentes I.Oliveira, José Palazzo Moreira de.  
II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Profa. Dra. Wrana Panizzi

Pró-Reitor de Ensino: Prof. José Carlos Ferraz Hennemann

Pró-Reitor Adjunto de Pós-Graduação: Prof. Philippe Olivier Alexandre Navaux

Diretor do Instituto de Informática: Prof. Dr. Philippe Olivier Alexandre Navaux

Coordenador do PPGC: Prof. Dr. Carlos Alberto Heuser

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

## Agradecimentos

*Meus agradecimentos a todas as pessoas que contribuíram para o desenvolvimento deste trabalho.*

*Em especial agradeço:*

*Ao professor Dr. José Palazzo Moreira de Oliveira pelos seus valiosos ensinamentos e atenção recebidos em todas as fases do mestrado.*

*À CAPES pelo suporte financeiro que permitiu a dedicação à este trabalho.*

*Aos professores e funcionários do Instituto de Informática da Universidade Federal do Rio Grande do Sul pelo conhecimento que me passaram ao longo desta etapa e dedicação com que atendem e auxiliam a todos os acadêmicos.*

*Aos pesquisadores do Projeto Tapejara que prestaram apoio fundamental à elaboração deste trabalho. Especialmente destaco, Francine Bica, Adriana Soares Pereira e Grupo de Pesquisa em Processos Cognitivos da Unisinos (Prof. Milton Madeira, Regina, Eduardo, Ricardo).*

*Que lega*

*Ao Juliano Moraes, cujo auxílio foi fundamental.*

*A todos os amigos e colegas do mestrado, pelo prazer da convivência, pelo enriquecimento através de conversas e trocas de conhecimentos, pela felicidade proporcionada nas horas de descontração. Em especial à grande amiga de todos os momentos, Graciana Simoni Fischer.*

*A minha mãe que sempre me motivou e torceu pelo êxito final deste trabalho, dando-me forças a continuar.*

*As minhas irmãs, Alexandra, Patrícia e Luciana pela amizade, companheirismo, estímulo constante, apoio e preocupação.*

*Ao meu namorado, Lisandro Zambenedetti Granville, por ter estado presente nos momentos finais desta etapa, estimulando com sua dedicação e carinho, auxiliando nas dificuldades, contribuindo para o pleno sucesso deste trabalho.*

## Sumário

<b>Lista de Abreviaturas.....</b>	<b>7</b>
<b>Lista de Figuras .....</b>	<b>8</b>
<b>Lista de Tabelas.....</b>	<b>9</b>
<b>Resumo .....</b>	<b>10</b>
<b>Abstract.....</b>	<b>11</b>
<b>1 Introdução.....</b>	<b>12</b>
1.1 Motivação.....	13
1.2 Objetivos.....	14
1.3 Organização .....	14
<b>2 Definição de Sistema de Alertas.....</b>	<b>16</b>
2.1 Sistema de Alertas no Gerenciamento de Projetos.....	20
2.1.1 Sistemas Comercias para Gerenciamento de Projetos .....	23
2.2 Sistema de Alertas na Área Médica.....	26
2.3 Sistema de Alertas no Gerenciamento de Redes.....	30
2.4 Sistema de Alertas em Biblioteca Digital.....	32
<b>3 Processo de Avaliação nos Ambientes de Ensino a Distância via <i>Web</i></b> .....	<b>35</b>
3.1 WebCT .....	35
3.1.1 Avaliação no <i>WebCT</i> .....	36
3.2 Virtual-U .....	37
3.3 <i>TopClass</i> .....	38
3.3.1 Processo de Avaliação no TopClass .....	39
3.4 <i>Web Course in a Box</i> .....	40
3.4.1 Avaliação no WCB .....	40
3.5 <i>Learning Space</i> .....	40
3.5.1 Avaliação no Learning Space .....	41
3.6 AulaNet.....	41
3.6.1 Avaliação no AulaNet.....	43
3.7 <i>CyberQ</i> .....	45
3.8 <i>Carnegie Mellon On-Line</i> .....	46
3.9 <i>CADAL Quiz</i> .....	48
3.10 SMART.....	49

<b>3.11 QuizSite .....</b>	<b>50</b>
<b>3.12 QUIZIT .....</b>	<b>50</b>
<b>3.13 Hyperlearning Meter System .....</b>	<b>50</b>
<b>3.14 Comparação dos Processos de Avaliação nos Ambientes de Ensino a Distância via Web.....</b>	<b>51</b>
<b>4 Técnicas Propostas para Avaliação de Alunos em Cursos na Web ...</b>	<b>54</b>
<b>4.1 AnalogVB - Ferramenta de Análise de Atividades de Alunos de Sistemas de Ensino via Internet .....</b>	<b>54</b>
<b>4.2 Access Miner – Ferramenta de Mineração de dados na descoberta do comportamento do usuário da Web .....</b>	<b>55</b>
<b>4.3 Utilizando Agentes no Suporte à Avaliação Informal no Ambiente de Instrução baseada na Web - AulaNet.....</b>	<b>56</b>
<b>4.4 Modelo de Avaliação segundo Maythe Vasconcelos.....</b>	<b>57</b>
<b>4.5 Modelo de Avaliação segundo Luciano Hack .....</b>	<b>58</b>
<b>4.6 Proposta para um Ambiente de Avaliação Pedagógica - AmAPed .....</b>	<b>59</b>
<b>4.7 Ambiente Hipermídia Integrado para o Desenvolvimento de Cursos a Distância .....</b>	<b>60</b>
<b>5 Descrição do Sistema de Alertas .....</b>	<b>64</b>
<b>5.1 O Sistema de Alertas .....</b>	<b>65</b>
5.1.1 O Modelo Especificado.....	66
5.1.2 Integração do Sistema de Alertas com o projeto Tapejara.....	67
5.1.3 Dados de Entrada do Sistema.....	68
<b>5.2 Mecanismo de Rastreamento.....</b>	<b>71</b>
<b>5.3 Base de Conhecimento .....</b>	<b>77</b>
5.3.1 As Regras de Produção .....	77
<b>5.4 Comunicação entre o Sistema de Alertas e Agentes do Tapejara.....</b>	<b>78</b>
<b>5.5 Diagrama de Classes do Protótipo Implementado .....</b>	<b>79</b>
5.5.1 Classes Alerta e Condição.....	80
5.5.2 Classes CondiçãoTempo, CondiçãoMensagem e CondiçãoTempoMensagem ....	80
5.5.3 Classe Observador.....	81
<b>5.6 Funcionamento do Sistema de Alertas .....</b>	<b>82</b>
5.6.1 Processo de Execução do Sistema de Alertas .....	82
5.6.2 Eventos Percebidos .....	83
5.6.3 Tipos de Mensagens.....	84
<b>5.7 Implementação do Protótipo .....</b>	<b>85</b>
5.7.1 Requisitos para Implementação do Sistema de Alertas .....	85

5.7.2 Ambiente de Implementação do Sistema de Alertas .....	86
5.7.3 A Interface .....	88
5.7.4 Operações do Sistema de Alertas .....	89
5.7.5 Exemplos de Alertas gerados pelo Sistema de Alertas .....	93
<b>6 Conclusão .....</b>	<b>97</b>
<b>6.1 Trabalhos Futuros .....</b>	<b>98</b>
<b>Anexo 1 Alertas gerados pelo Sistema.....</b>	<b>100</b>
<b>Anexo 2 Exemplo de mensagens utilizadas pelo Sistema de Alertas.....</b>	<b>110</b>
<b>Anexo 3 Gramática.....</b>	<b>114</b>
<b>Bibliografia.....</b>	<b>115</b>

## Lista de Abreviaturas

<b>AME-A</b>	Ambiente Multiagente de Ensino-Aprendizagem
<b>API</b>	Aplication Programming Interface
<b>BD</b>	Banco de Dados
<b>CADAL QUIZ</b>	Computer Aided Dynamic Assessment & Learning Quiz
<b>CGI</b>	Common Gateway Interface
<b>COM</b>	Component Object Model
<b>DBA</b>	Administrador de banco de dados
<b>DW</b>	Data warehouse
<b>EAD</b>	Educação a Distância
<b>EDI</b>	Electronic Data Interchange
<b>HTML</b>	Hypertext Markup Language
<b>HTTP</b>	HyperText Transfer Protocol
<b>IBW</b>	Instrução baseada na Web
<b>JDBC</b>	Java Database Connectivity
<b>KQML</b>	Knowledge Query and Manipulation Language
<b>RMR</b>	Regenstrief Medical Record
<b>SGBD</b>	Sistema de Gerenciamento de Banco de Dados
<b>SMART</b>	Special Multimedia Arenas for Refining Thinking
<b>SMTP</b>	Simple Mail Transfer Protocol
<b>SQL</b>	Structured Query Language
<b>SSL</b>	Secure Sockets Layer
<b>UFRGS</b>	Universidade Federal do Rio Grande do Sul
<b>UML</b>	Unified Modeling Language
<b>URL</b>	Uniform Resorce Locator
<b>WBC</b>	Web Course in a Box
<b>WBT</b>	Web-Based Training
<b>WWW</b>	World Wide Web
<b>XML</b>	Extensible Markup Language

## Lista de Figuras

FIGURA 2.1- Mecanismo de Alerta em banco de dados .....	17
FIGURA 2.2 - Arquitetura de um Sistema de Alertas de Negócios.....	23
FIGURA 2.3 - Arquitetura do Sistema Categoric Alerts.....	25
FIGURA 2.4- Arquitetura do Serviço de Alertas em Biblioteca Digital.....	34
FIGURA 3.1 - Arquitetura do Carnegie Mellon On-Line .....	48
FIGURA 4.1 - Estrutura para avaliação do aluno .....	59
FIGURA 5.1- Arquitetura Multiagente Projeto Tapejara.....	64
FIGURA 5.2 - Arquitetura do Sistema de Alertas.....	67
FIGURA 5.3– Integração do Sistema de Alertas com o projeto Tapejara .....	68
FIGURA 5.4– Modelo Entidade-Relacionamento .....	76
FIGURA 5.5- Regra de Produção.....	77
FIGURA 5.6 - As classes Alerta e Condição .....	80
FIGURA 5.7- As classes CondiçãoTempo, CondiçãoMensagem e CondiçãoTempoMensagem .....	81
FIGURA 5.8- A classe Observador .....	82
FIGURA 5.9 - Processo de Execução do Sistema de Alertas .....	82
FIGURA 5.10 - Interface do Editor de Alertas .....	88
FIGURA 5.11– Interface do Editor de Alertas (continuação).....	89
FIGURA 5.12 – Código Java mostrando acesso JDBC .....	90
FIGURA 5.13– Código Java mostrando operação de registro dos alertas .....	90
FIGURA 5.14– Código Java mostrando acesso à base de conhecimento .....	91
FIGURA 5.15– Código Java mostrando operação de envio de <i>e-mail</i> .....	92
FIGURA 5.16– Exemplo de alerta “Retorno rápido ao curso” .....	93
FIGURA 5.17- Exemplo de alerta “Pulo Exercício”.....	94
FIGURA 5.18– Exemplo de alerta “Parada Brusca”.....	94
FIGURA 5.19 – Mensagem de alerta enviada ao professor, por <i>e-mail</i> .....	95
FIGURA 5.20 – Mensagem de alerta enviada ao aluno por <i>e-mail</i> .....	96
FIGURA 5.21 – Exemplo de mensagem enviada ao agente .....	96



## Lista de Tabelas

TABELA 2.1 - Correlação entre condição de alerta e critério .....	27
TABELA 3.1 - Comparativo entre os modos de avaliação .....	52
TABELA 5.1 – Tabela Curso .....	71
TABELA 5.2 – Tabela Bibliografia .....	71
TABELA 5.3 – Tabela Modulo .....	72
TABELA 5.4 – Tabela Tatica.....	72
TABELA 5.5 – Tabela URL .....	72
TABELA 5.6 – Tabela Pessoa.....	72
TABELA 5.7 – Tabela AlunoCurso .....	73
TABELA 5.8 – Tabela Autor .....	73
TABELA 5.9 – Tabela LOG .....	73
TABELA 5.10 – Tabela Estilo .....	73
TABELA 5.11 – Tabela Topico .....	73
TABELA 5.12 – Tabela Avaliacao .....	74
TABELA 5.13 - Tabela Estratégia .....	74
TABELA 5.14 – Tabela Exercícios.....	74
TABELA 5.15- Classe KQMLMessage.....	91
TABELA 5.16 – Classe ContentMessage .....	91

## Resumo

A difusão da educação baseada na *Web* está trazendo uma série de mudanças nesta área. Uma dessas mudanças está na forma de como se avaliar as atividades dos alunos remotos, não só através de tarefas tradicionais como testes, mas verificando, em tempo-real, as ações dos alunos e assim possibilitando ao professor um acompanhamento mais completo das atividades dos estudantes.

De acordo com os recursos computacionais existentes, a utilização de um Sistema de Alertas é a opção que melhor se adequa a estas finalidades, pois com este tipo de sistema é possível acompanhar as atividades dos alunos em cursos a distância.

O objetivo deste trabalho é apresentar um Sistema de Alertas Inteligentes para apoio ao ensino, que detecta problemas nas atividades dos alunos em cursos na *Web* e realiza ações corretivas adequadas. Este sistema está parcialmente integrado ao ambiente Tapejara do Instituto de Informática da UFRGS – Sistemas Inteligentes de Ensino na Web - que consiste em um sistema de construção e acompanhamento de cursos disponibilizados via Internet.

A principal característica do Sistema de Alertas Inteligentes é a busca de situações críticas como, por exemplo: aluno apresenta baixo desempenho nos exercícios, a estratégia de ensino não corresponde ao perfil do estudante, aluno não está comparecendo às atividades do curso, etc. Com isto, este sistema pode auxiliar o professor (tutor virtual) a ter um acompanhamento mais preciso sobre as atividades realizadas pelo estudante e assim, adaptar as aulas às características do aluno, sem, com isto, acarretar numa sobrecarga de trabalho.

**Palavras-Chaves:** Educação a distância, Alertas Inteligentes, Sistema de Alertas, Avaliação Remota, Monitoramento de atividades na *Web*, Agentes

**TITLE: "AN INTELLIGENT ALERTS SYSTEM TO INTERNET-BASED TEACHING ENVIRONMENTS "****Abstract**

The dissemination of *Web*-based education strategies brings several changes to this area. One of these changes is related to the way in which tasks performed by remote students are evaluated, not only through traditional mechanisms as standard tests, but also verifying students' actions in real-time, and thus allowing a more complete student support, mainly related to the students' tasks.

Based on the current available computational resources, the use of an Alerts System is the option that best helps to achieve these goals, since with this kind of system it's possible to follow up students' activities in long distance courses.

The goal of this work is to present an Intelligent Alerts System to support teaching that detects student task problems in *Web*-based courses and proceeds with properly solving actions. This system is partially integrated to the Tapejara environment of the Informatics Institutes at UFRGS – Intelligent *Web*-based Teaching Systems. Tapejara environment is a system to build and support courses deployed on the Internet.

The main feature of the Intelligent Alerts System is the ability to find critical learning situations, such as: student with low performance on exercises, the teaching strategy does not correspond to the student profile, student is absent in course activities, etc. With this feature, the system can help teachers (virtual tutors) having more precise information about student tasks, and thus allowing teachers to adapt classes to the student profile, without having an overflow in the amount of work normally required.

**Keywords:** Distance learning, Intelligent Alerts, Alerts System, remote assessment, *Web*-based activities monitoring, Agents

# 1 Introdução

Nessa última década, o aumento da capacidade de processamento dos computadores pessoais, a tecnologia das redes e a capacidade de colocar à disposição a um número cada vez maior de pessoas todos estes instrumentos da computação originaram o interesse em uma área que vem sendo pesquisada há mais de um século – a área de Educação a Distância (EAD).

A Educação a Distância permite que professor e aluno, apesar de fisicamente separados, possam interagir com o auxílio de algum tipo de tecnologia. A *World Wide Web* se apresenta como uma tecnologia capaz de atender as expectativas dos pesquisadores da área de ensino/aprendizagem a distância, proporcionando soluções para o problema do oferecimento de educação e treinamento em larga escala, a custos mais acessíveis do que os atuais, permitindo a publicação de material didático, aplicação de tutoriais, aplicação de provas e testes, comunicação com os estudantes e apresentação de aulas a distância (conferência multimídia).

Empresas e instituições de ensino têm investido grandes recursos em pesquisas relacionadas à utilização de computadores em ambientes de ensino a distância como solução para o atendimento a uma demanda crescente, oferecendo novas oportunidades educacionais.

Para o desenvolvimento de cursos de ensino a distância, o docente não terá uma presença tão efetiva como nos cursos presenciais. Frente a isso, uma das preocupações apresentadas pelos professores é o processo de avaliação dos esforços individuais dos estudantes, principalmente quando envolve um grande número de participantes, assim como a obtenção de resultados que permitam a avaliação da classe como um todo, pois, neste método de ensino a ausência do professor dificulta a avaliação.

Para verificar a qualidade e eficácia de um aprendizado, os professores, ao avaliarem o aluno de forma presencial, utilizam mecanismos formais, como testes, trabalhos, exercícios; e mecanismos complementares, como a repetição oral de alguma idéia passada, registro da linguagem e expressão facial dos alunos, etc. A avaliação, utilizando mecanismos complementares, permite ao professor fazer alguns ajustes nos seus procedimentos de ensino, diminuindo/aumentando o ritmo da aula, fazendo revisões.

No processo de avaliação em EAD o que normalmente tem-se visto é a utilização de provas e testes [RIT 2000]. Desse modo, esses ambientes são desprovidos de mecanismos de avaliação informal. Os professores estão buscando alternativas, para que, mesmo à distância, possam obter dos alunos informações complementares, que trarão um melhor entendimento sobre a real situação do aprendizado [ROD 98]. Por esta razão, é muito importante a existência de ferramentas apropriadas que possibilitem o monitoramento do que está sendo feito por cada estudante, detectando situações de exceção e avisando ao professor (através de alertas) os problemas que estão ocorrendo com alguns alunos. Com isso, o professor terá acesso às peculiaridades de cada um, e ser-lhe-á possibilitado adequar as aulas às individualidades podendo, ele, dessa forma, exercer um acompanhamento mais próximo, estimulando e orientando o aluno, sem que isso acarrete um ônus maior.

## 1.1 Motivação

Segundo BERCH, a avaliação pedagógica tem uma importância fundamental, pois pode ser vista como a base para a tomada de decisões para que um professor adote atividades de reforço, modifique suas posturas com o aluno, forneça ajudas simples, melhores explicações, exemplos e situações, aprofunde questões, proporcione desafios, desenvolva episódios para a aprendizagem e, inclusive, considere o aluno apto em relação ao conteúdo ou à habilidade trabalhados [BER 97].

Um dos problemas encontrados em avaliação do ensino a distância é que, em ambientes tradicionais de ensino, o processo de avaliação realizado pelos professores com relação aos seus alunos não se restringe à utilização de métodos formais, tais como provas e exercícios. Eles utilizam-se de um conjunto de métodos informais, decorrentes do *feedback* presencial recebido, que permite avaliar o processo de ensino-aprendizagem [HAY 97]. Uma das principais características dos ambientes de ensino a distância é a ausência desse aspecto presencial encontrado nos ambientes tradicionais de ensino, caracterizados pela existência do contato direto entre professor e alunos. Por essa razão, ambientes de ensino a distância são desprovidos de mecanismos de avaliação informal [HAY 97].

Alguns ambientes de ensino a distância têm implementado o processo de avaliação do ensino-aprendizagem, utilizando técnicas formais para apurar os resultados [ZAN 99]. No que diz respeito à implementação de técnicas informais, existem modelos de avaliação que descrevem a técnica, como em [VAV 98].

A avaliação do aprendizado do estudante é um dos principais aliados no processo de ensino-aprendizagem, pois é através dela que consegue-se identificar os principais problemas na aprendizagem, tentando, a seguir, solucioná-los através de estratégias de ensino diferenciadas para cada perfil de estudante.

O processo de avaliação deve representar o resultado do esforço aplicado durante todo o curso. Se o processo não considerar aspectos como o perfil do aluno, estilo cognitivo, estratégias e táticas utilizadas, não será possível avaliar o estudante de forma correta e completa. Por isso, o processo de avaliação deve estimular o aluno durante todo o percurso, fazendo com que o mesmo se sinta avaliado continuamente [MEN 98].

Com o processo de avaliação sendo feito a todo momento, há a possibilidade de evitar o não aproveitamento do curso de forma satisfatória, modificando a estratégia para que o esforço aplicado não seja desperdiçado devido a uma má estratégia empregada.

Em ambientes de ensino via Internet, como não há a interação em sala de aula entre aluno-professor e aluno-aluno, fato considerado de extrema importância no processo de ensino aprendizagem, em geral, o comportamento dos alunos é avaliado pelo uso da tecnologia de agentes da Inteligência Artificial, mais especificamente Sistemas Multiagentes, uma Técnica de Inteligência Artificial Distribuída [COS 99] [SAN 2000]. Um agente monitora os processos de interação dos alunos com o ambiente do curso, que reflete o seu processo de aprendizagem no curso [JAQ 99]. Assim, é possível escolher um caminho próprio de aprendizagem para cada perfil de estudante. Quanto maior for a quantidade de informações sobre o comportamento de cada aluno,

os tutores virtuais poderão auxiliar os alunos mais adequadamente.

Uma das técnicas utilizadas para avaliar o comportamento dos alunos em cursos na *Web* é a análise dos registros dos *logs* de acessos ao servidor *Web* de ensino a distância. Tais registros podem fornecer o comportamento de cada aluno no tocante aos acessos aos arquivos (páginas), armazenados no servidor *Web*. Porém, é preciso realizar várias consultas nestes registros para se obterem informações que sejam importantes e que possam auxiliar a descobrir o comportamento do aluno. Um sistema de alertas pode ser usado para realizar estas consultas, o que auxiliará os tutores virtuais a avaliarem o comportamento de cada aluno.

Um sistema de alertas pode ser visto como o mecanismo intermediário entre o registro de *log* e os agentes inteligentes.

Um sistema de alertas inteligentes é projetado para detectar situações especiais e gerar avisos (*warnings*) que podem ser úteis no processo decisório. Através de uma análise no histórico contido no arquivo de *log*, é possível identificar problemas no aprendizado do aluno, ou seja, reconhecer quando o aluno está precisando de ajuda, perceber um baixo desempenho ou, então, verificar quando a estratégia de ensino não corresponde ao perfil do estudante. Depois de reconhecido o problema de aprendizado, o sistema de alertas comunica-se com os agentes que devem solucionar o problema, por exemplo, apontando uma nova estratégia de ensino correspondente ao perfil do aluno.

## 1.2 Objetivos

O objetivo deste trabalho é a criação de módulos de alertas inteligentes que irão monitorar as atividades dos alunos em Sistemas de Ensino na *Web*, mais especificamente no projeto Tapejara da Universidade Federal do Rio Grande do Sul. O projeto Tapejara – Sistemas Inteligentes de Ensino na *Web* – consiste em projetar e implementar um sistema de construção e acompanhamento de cursos a serem disponibilizados na Internet.

O sistema de alertas serve para que quando ocorra algum problema nas atividades dos alunos no curso, o docente (tutor virtual) seja avisado deste problema e realize a ação corretiva adequada. Sua principal característica é a busca de situações críticas com base nas informações que possui a sua disposição. As informações utilizadas pelo sistema de alertas estão armazenadas em um banco de dados (*log*) e correspondem as atividades realizadas pelos alunos em um curso pela *Web*.

## 1.3 Organização

Este trabalho está organizado da seguinte forma:

O capítulo 2 introduz os conceitos de alertas e eventos em banco de dados e a integração com as regras que representam conhecimento. A importância do uso de alertas em transações comerciais e a maneira que eles interagem com os sistemas já existentes em uma empresa, exemplificando com tecnologias já disponíveis no mercado, são apresentadas neste capítulo. Finalmente, mostra-se a utilidade dos sistemas de alertas na área médica, gerenciamento de redes e bibliotecas digitais.

O capítulo 3 apresenta um estudo sobre os principais ambientes de ensino-aprendizagem virtuais, onde será levado em conta, principalmente, o processo de avaliação, objetivo principal deste trabalho. Neste capítulo, são tratados ambientes a distância mais conhecidos e que estão sendo utilizados em várias instituições de ensino assim como, os ambientes de ensino a distância não tão conhecidos e, na sua maioria, implantados apenas nas suas instituições de origem. Ao final, faz-se uma comparação dos principais ambientes de ensino a distância, onde é descrita a funcionalidade do processo de avaliação em cada um dos ambientes, gerando uma comparação entre eles. Tal comparação procura destacar as técnicas e instrumentos de avaliação implementados em cada um dos ambientes, bem como a tecnologia utilizada para tal implementação.

O capítulo 4 descreve alguns modelos de avaliação existentes, ainda não implementados totalmente, que descrevem as funcionalidades, tecnologias e metodologias utilizadas para a construção de um *software* de avaliação de ensino a distância.

O capítulo 5 descreve o Sistema de Alertas e os aspectos da implementação do protótipo. São descritos o ambiente de desenvolvimento bem como o funcionamento de cada parte do Sistema de Alertas.

As conclusões deste estudo, assim como os trabalhos futuros a serem realizados com base nos estudos e implementação desenvolvidos nesta etapa, são apresentados no capítulo 6.

## 2 Definição de Sistema de Alertas

Um alerta é um programa que monitora um banco de dados e informa para alguns usuários ou para um programa quando uma determinada condição ocorre, oferecendo recursos para tomada de decisões [BUN 79] [HIN 99c].

Em bancos de dados, pode ser desejável monitorar a presença de algum estado que possa resultar de uma atualização por algum outro agente. Em um programa de manutenção de banco de sangue, por exemplo, pode ser necessário notificar o responsável quando o estoque de um tipo de sangue estiver abaixo de um nível crítico. Este é um caso típico para um programa que monitore o banco de dados para tal condição e emita um alerta quando isto ocorrer.

Existem basicamente quatro cenários em que uma aplicação pode ser avisada na ocorrência de um evento [KRI 95]:

**Notificação Síncrona:** a aplicação espera a ocorrência de um evento em particular para continuar o processamento. O evento geralmente é uma alteração de um dado na base de dados, por exemplo: uma operação *Insert, Update, Delete*.

Um exemplo deste tipo de notificação é: gerentes de uma seção podem agendar tarefas para seus empregados utilizando uma aplicação simples, que apresente uma lista das tarefas agendadas e não terminadas na sua seção. A lista deve ser atualizada automaticamente quando uma nova tarefa for agendada ou quando uma tarefa é finalizada. Um sistema de alertas pode ser usado para evitar que a aplicação fique examinando repetidamente o banco de dados até que a lista de tarefas tenha sido atualizada.

**Notificação Assíncrona:** a aplicação não espera pela ocorrência de um evento em particular, ela continua o processamento normalmente e é avisada quando um evento de interesse ocorre. Por exemplo, em uma seção de produção, o funcionário também pode visualizar a lista de tarefas em execução. Porém, ele pode usar outras partes da aplicação para verificar detalhes sobre suas tarefas. Quando uma nova tarefa é agendada para o empregado, ela pode ser incorporada dentro da lista dele, mas isto não afeta outras partes da aplicação. Isto evita que a aplicação fique inspecionando o banco de dados para detectar se um evento de interesse ocorreu.

**Notificação Interrupção:** A aplicação interrompe o processamento quando um evento de interesse ocorre e, antes de continuar, deve realizar um serviço. Neste caso, o evento é uma alteração no banco de dados.

**Notificação baseada em tempo:** o processamento da aplicação é interrompido quando um evento específico, baseado em tempo, ocorre. O evento pode ser um período de tempo que expirou ou a chegada de uma instância particular de tempo. Um exemplo deste tipo de evento que envolve uma instância de tempo é o seguinte: cada tarefa agendada possui a data e hora de término. Uma hora antes do tempo agendado, o empregado que supervisiona a tarefa é avisado por uma mensagem em seu terminal.

Alertas podem ser usados para evitar que uma aplicação fique regularmente examinando uma grande parte do banco de dados para determinar se uma instância de



tempo em particular foi materializada ou se um período de tempo expirou [CHA 82].

Um mecanismo de alerta consiste dos seguintes componentes, conforme figura 2.1.

**Definição de evento-** Todo evento que é controlado pelo banco de dados deve ser definido. A definição nomeia o evento e assinala-o como um evento de interesse. Não indica quando o evento deve ser ativado, nem quais aplicações devem ser notificadas [GEH 92]. O evento é tipicamente definido por um administrador de banco de dados (DBA) que usa as facilidades de um sistema de definição de banco de dados [CHA 94].

**Ativação de evento-** O mecanismo de ativação do evento é usado para especificar quando o evento ocorre. Um evento é considerado ativo quando todas as condições que especificam o evento são conhecidas. Isto deve ser notificado para todas as aplicações que se interessaram pelo evento.

**Registro do evento-** Na maioria dos casos, as aplicações que têm acesso ao banco de dados não são notificadas automaticamente quando eventos acontecem. As aplicações precisam indicar quais os eventos que elas têm interesse, enviando um comando de registro de evento para o banco de dados.

**Notificação de evento-** Uma aplicação que precisa receber um ou mais eventos deve indicar como este evento será notificado (síncrona, assíncrona ou interrupção). Uma aplicação especifica seu próprio modo de notificação, emitindo um comando ao banco de dados.

**Recuperação de evento-** Quando uma aplicação é avisada da ocorrência de um evento, é necessário recuperar detalhes deste evento. A notificação normalmente só informa para a aplicação que um dos eventos de interesse ocorreu. A aplicação deve recuperar o evento da fila de eventos para inspecionar seus conteúdos. Os conteúdos indicam o tipo de evento e outros detalhes especificados pelo mecanismo de notificação de eventos.

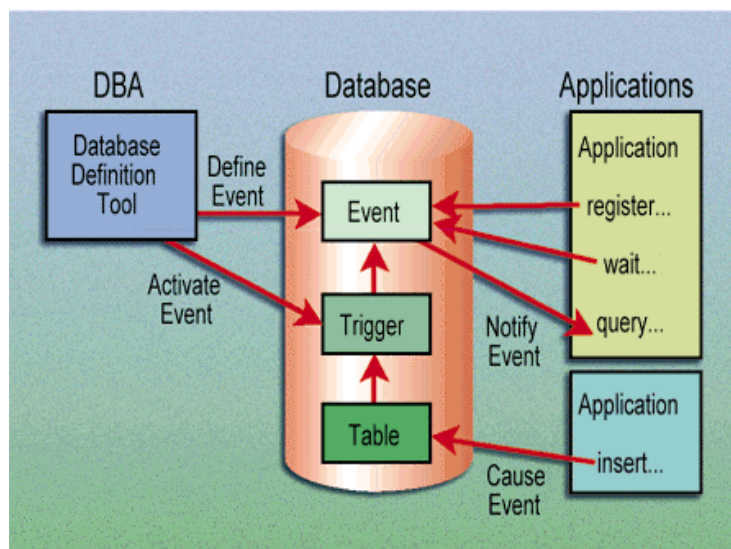


FIGURA 2.1- Mecanismo de Alerta em banco de dados [REC 97]

Em banco de dados, o controle dos eventos é realizado por *triggers* [CHA 82]. *Trigger* permite que o sistema execute automaticamente uma série de atividades quando uma determinada operação é executada pelo usuário. O *trigger* é gerenciado pelo SGBD (Sistema de Gerência de Banco de Dados) e as condições são especificadas pelo DBA (administrador do banco de dados).

Em um sistema de alertas, é o usuário que especifica condições que necessitam ser monitoradas em tempo real sobre trocas no banco de dados e as ações que devem ser realizadas na ocorrência destas condições [NAR 97].

Um sistema de geração de alertas possui duas partes distintas: uma base de dados e uma base de conhecimento. A base de dados corresponde ao Sistema de Banco de Dados da instituição, onde são armazenadas as informações tradicionalmente contidas em um sistema gerencial comum, como dados dos alunos, dos funcionários e de estoque. Estas informações são específicas, dependem de cada instituição e não são representadas de uma forma padronizada. Cada instituição armazena seus dados da forma que lhe é mais conveniente.

A base de conhecimento representa as situações de exceção que devem ser detectadas e geralmente são representadas através de regras, no formato SE-ENTÃO [NAR 97]. Uma regra é, simplesmente, um par *condição-ação*: dada a existência da condição expressa, faça a ação. As regras são expressas da seguinte forma:

SE <condição> ENTÃO <ação>

A partir de uma base sobre internação hospitalar, por exemplo, diversos tipos de regras podem ser expressas, entre eles:

SE diagnóstico = X E procedimento = Y ENTÃO tempo internação = 0

SE diagnóstico = X E faixa etária = Y ENTÃO paciente é transferido

O primeiro tipo de regra poderia ser útil, por exemplo, para se procurarem exceções que poderiam evitar fraudes. A inclusão de informações que violam esta regra na base de dados poderia representar uma situação que deveria ser observada pelo usuário. Esta regra, então, seria utilizada pelo sistema de alertas, podendo evitar a inclusão de erros na base.

Juntamente com o sistema de alertas, tem sido proposto o uso de ferramentas de mineração de dados ou *data mining* [FEL 98], que consiste no processo de analisar grandes volumes de dados sob diferentes perspectivas, a fim de descobrir informações úteis que normalmente não são visíveis.

Dentre as diversas técnicas de mineração de dados, uma que se destaca é a procura de regras de associação, que tem por objetivo encontrar relacionamentos ou padrões frequentes entre conjuntos de dados. A partir de uma base, diversas regras podem ser descobertas.

Regras de associação têm sido usadas com frequência para descrever o comportamento do consumidor no comércio varejista [BRU 2000]. Porém, esta não é a única área em que estas técnicas têm sido utilizadas. Elas podem auxiliar a compreensão

do comportamento do usuário na *Web*, através da descoberta de padrões ou regularidades que descrevem o seu perfil. Podem fazer previsão e diagnósticos de alarmes em telecomunicação, indicar complicações em certos tipos de tratamentos médicos, analisando históricos de pacientes.

De acordo com FELDENS, as regras descobertas pelo sistema de mineração, podem ser reutilizadas pelo sistema de alertas [FEL 99].

As informações contidas na base de conhecimento, por não serem específicas de uma determinada instituição, podem ser aplicadas a várias instituições e geralmente são representadas através de um padrão.

Esta não-padronização da forma em que os dados são armazenados em cada instituição traz problemas sérios para a realização da integração entre a base de conhecimento e a base de dados.

O processo de ligação entre um dado especificado na base de conhecimento e um dado da base de dados é uma tarefa complexa. É necessário que se defina uma consulta na base de dados que seja capaz de recuperar o dado necessário para a base de conhecimento. Para isso, o dado desejado na base de conhecimento deve ser especificado, os dados disponíveis da base de dados devem ser revistos, os termos apropriados e métodos de recuperação devem ser escolhidos e uma consulta sintaticamente correta deve ser criada. Isto requer conhecimento da organização da base de dados, da terminologia utilizada para armazenar a informação e da sintaxe da consulta.

Além do problema de recuperar os dados da base de dados, existe ainda o problema de realizar a interação entre um sistema ativo, como é o sistema de alertas, e o sistema de banco de dados da instituição. Os sistemas ativos devem possuir mecanismos capazes de "perceber" eventos externos e responder a eles [KRI 95]. Isto implica que os objetos que realizam um serviço de apoio à decisão devem ser capazes de tratar eventos. O problema maior, no entanto, está na forma como a ocorrência de um evento será detectada e informada no sistema externo ao serviço de suporte à decisão [WAT 86]. Considerando que a principal fonte de dados de entrada do sistema será um banco de dados, tipicamente os eventos poderão ser percebidos quando alguma modificação ocorrer no banco [BUN 79] [CHA 94] [CHA 82]. Além disso, o sistema também deve ser capaz de tratar eventos temporais e a combinação de eventos de banco de dados e eventos temporais.

A tecnologia atual de banco de dados relacional disponível comercialmente permite que se especifiquem eventos simples, baseados nas operações sobre uma relação (inserção, exclusão e alteração). Eventos temporais e eventos complexos geralmente não podem ser expressos. Além disso, o tipo de instrução que se pode executar em resposta a um evento é limitada à linguagem disponível no banco, geralmente baseada em SQL. Isto implica que uma instituição utilizando o serviço de apoio à decisão deve possuir uma arquitetura que seja capaz de tratar estes tipos de eventos.

Um serviço de alertas deve ser projetado para implementar a conexão entre a base de dados e a base de conhecimento.

O serviço de alertas implementa consultas que recuperam dados na base de dados da instituição. As modificações na base de dados, geralmente, são percebidas através de um monitor que inspeciona as modificações ocorridas na base e gera os eventos apropriados. Segundo LEÃO em [LEA 98], como é muito caro inspecionar toda a base de dados periodicamente, são utilizados *triggers* que geram relações delta. Relações delta são relações que guardam as modificações ocorridas na base de dados desde a última inspeção [NAR 97]. Assim, o monitor inspeciona as relações delta, que são muito menores, e não todo o banco de dados.

Muitas áreas, como a área médica, de gerência de redes e comercial, esta no gerenciamento de processos, já adotaram o sistema de alertas para auxiliá-los em suas funções e obtiveram resultados bastante satisfatórios.

## 2.1 Sistema de Alertas no Gerenciamento de Projetos

A todo o momento, as empresas acumulam mais informações em seus bancos de dados. Conseqüentemente, os bancos de dados passam a conter verdadeiros tesouros de informação sobre vários dos procedimentos dessas companhias: os hábitos de seus clientes, suas histórias de sucesso e fracassos estão registradas nas bases de dados. Toda essa informação pode ser usada para melhorar os procedimentos, permitindo que a empresa detecte tendências e características e reaja rapidamente a um evento que ainda pode estar por vir, como: o que pensam os clientes, como anda a concorrência, as flutuações do mercado, as variações dos estoques, o movimento do dinheiro.

No entanto, apesar do enorme valor desses dados, a maioria das organizações é incapaz de aproveitar totalmente o que está armazenado em seus arquivos. Esta informação preciosa está, na verdade, implícita, escondida sob uma “montanha de dados”, e não pode ser descoberta utilizando-se sistemas de gerenciamento de banco de dados convencionais.

É muito importante para um gerente de negócios ter controle total de seus projetos. Saber o que está acontecendo e ser informado das dificuldades que estão ocorrendo é fundamental a qualquer área, seja comercial, operacional, financeira, ou administrativa. Quanto mais rápido a empresa souber de algo que está ocorrendo, mais rápido pode agir e corrigir este problema.

Os gerentes de empresas enfrentam hoje o desafio da sobrecarga da informação, necessitando da entrega de informação de maneira rápida e concisa, em um formato que eles possam compreender e agir imediatamente. Para alcançar competitividade, as organizações estão indo mais longe dos tradicionais relatórios e análise passiva dos dados. Há necessidade de se obter tecnologias capazes de transformar grandes volumes de dados em informações significativas.

Algumas técnicas computacionais foram desenvolvidas para analisar os dados, ou ao menos ajudar a encontrar conhecimento implícito nos bancos de dados. Uma destas técnicas é o *data mining*, ou mineração de dados que são ferramentas capazes de explorar gigantescas bases de dados em busca de informações que ajudam as empresas a decifrar o comportamento do consumidor, a compra de suprimentos ou, ainda, administrar as áreas comerciais e financeira de maneira transparente. Estas ferramentas são capazes de aprender a partir dos dados, gerando e validando enormes volumes de

hipóteses e destacando conhecimento interessante para o usuário, eventualmente descobrindo conhecimento novo, útil e interessante.

As empresas voltaram-se para o uso destas tecnologias: *data warehouse*, *data marts* ou *bussiness intelligence* (inteligência de negócio), numa tentativa de dar mais valor aos dados que possuem.

Cabe aos *data warehouses*, ou armazém de dados, organizar, limpar e estruturar a “história” de uma organização, sendo úteis para revelar o conhecimento implícito sobre a organização, descobrir regras sobre seus relacionamentos com clientes, problemas de qualidade, gerência de materiais, e muitas outras aplicações [BRU 2000]. Os dados guardados nestes armazéns estão disponíveis para informação e consulta, sendo geralmente a fonte de dados para o processo de mineração.

Esses dados podem ser oriundos de diversas bases, por exemplo, filiais de uma empresa, de onde são extraídos periodicamente para serem guardados de forma centralizada. Dessa forma, o armazém de dados não é atualizado em tempo real, mas com alguma frequência que depende do balanço da eficiência operacional e da necessidade de dados atualizados. Pelo fato de utilizar um armazenamento separado, o armazém de dados pode ser acessado sem que isto se torne perceptível pelos usuários da base de dados normal, pois não ocorrerão efeitos como bloqueio de registros e redução no desempenho de processamento do servidor.

*Data Mart* é uma abordagem mais “focada” na tecnologia de *datawarehouses* em que os dados estão mais próximos aos usuários. Ao invés de se projetar um armazém de dados centralizado, capaz de atender todas as áreas de decisão da empresa, adota-se uma abordagem departamental, preservando-se a possibilidade de se integrar posteriormente diversos *data marts*, formando um *datawarehouse* completo [FEL 99].

Muitas empresas estão utilizando a idéia dos *data marts* para introduzir estas novas tecnologias de informação. Limitando o tamanho e o escopo do projeto, empresas podem dispor de sistemas de suporte à decisão em tempo e a custos razoáveis, justificando o projeto em termos de retorno de investimento muito mais facilmente, e aumentando as chances de sucesso. O *data mart* torna-se um protótipo para o *datawarehouse* global.

Todas estas soluções são alimentadas com os dados coletados dos sistemas internos da empresa, como clientes, fornecedores, e sócios. Estes sistemas contêm uma riqueza da informação, mas requerem frequentemente esforço complexo e consumo de tempo, a fim de trazer benefícios. Porque os armazéns dos dados (*datawarehouses*) representam uma cópia estática dos registros da base de dados de uma empresa, não podem ser a origem para a detecção de eventos de negócios em tempo real. Dependendo da frequência de atualização adotada pela empresa para os *datawarehouse*, a chance de se descobrir possíveis falhas, pode demorar dias ou até mesmo semanas, e isto pode significar perda de clientes e oportunidades de negócio.

Os sistemas de bancos de dados atuais tentaram resolver este problema aplicando tecnologia de baixo nível, como os *triggers* e *stored procedures* [WID 94], que alertam a gerência quando os elementos alcançam um estado crítico, por exemplo: estoques baixos. Estes Sistemas de Gerência de Banco de Dados não são capazes de percorrer a base de dados para monitorar regras de negócio únicas e para informar

então, dinamicamente ao gerente, ao cliente, ou ao fornecedor através do *e-mail*, do *pager*, do celular, do *Web site*, do fax, quando um estado crítico é alcançado.

Além disso, as companhias não armazenam, nas bases de dados, o custo de um produto, os preços e políticas do concorrente, o lucro. Na maioria das organizações, esta informação é perdida através de múltiplas bases de dados.

A tecnologia de hoje está evoluindo para encontrar soluções para as necessidades dos gerentes, permitindo que as companhias estendam suas potencialidades além do reino de relatórios básicos e de ferramentas de análise de dados. Os sistemas de alertas têm sido a solução encontrada. Eles fornecem a notificação necessária aos gerentes. Os sistemas de alertas de negócio permitem às empresas obterem um melhor resultado dos seus recursos e um controle dos dados em tempo-real [ART 99].

Apesar dos relatórios continuarem sendo importantes para o gerenciamento de negócios, até nas mais modernas empresas, os sistemas de alertas oferecem utilidades distintas, conforme OBAL em [OBA 2000]:

- Alertas são rápidos- são entregues na hora em que ocorrem o evento associado;
- Alertas são específicos do usuário;
- Podem ser entregues em qualquer lugar, por *e-mail*, telefone celular, *pager*, *Web*, fax, etc;
- Alertas são tipicamente situações críticas, baseadas em exceções às regras de negócios;
- Alertas podem sugerir uma ação apropriada ao evento;
- Alertas são iniciados automaticamente por um evento;
- Alertas podem ser enviados para os clientes, sócios e fornecedores permitindo uma integração e colaboração.

O uso de alertas pode ajudar os negócios nos seguintes itens:

- Melhora os serviços prestados ao cliente, pois reduz o número de falhas;
- Melhora o controle de custo pela redução de tempo e cobrança veloz;
- Reduz fraudes e re-trabalho;
- Aumenta a produtividade gerencial eliminado gasto de tempo na procura por dados que podem ser enviados via alerta.

A figura 2.2 mostra a integração das várias tecnologias, existentes em uma empresa com o sistema de alertas.

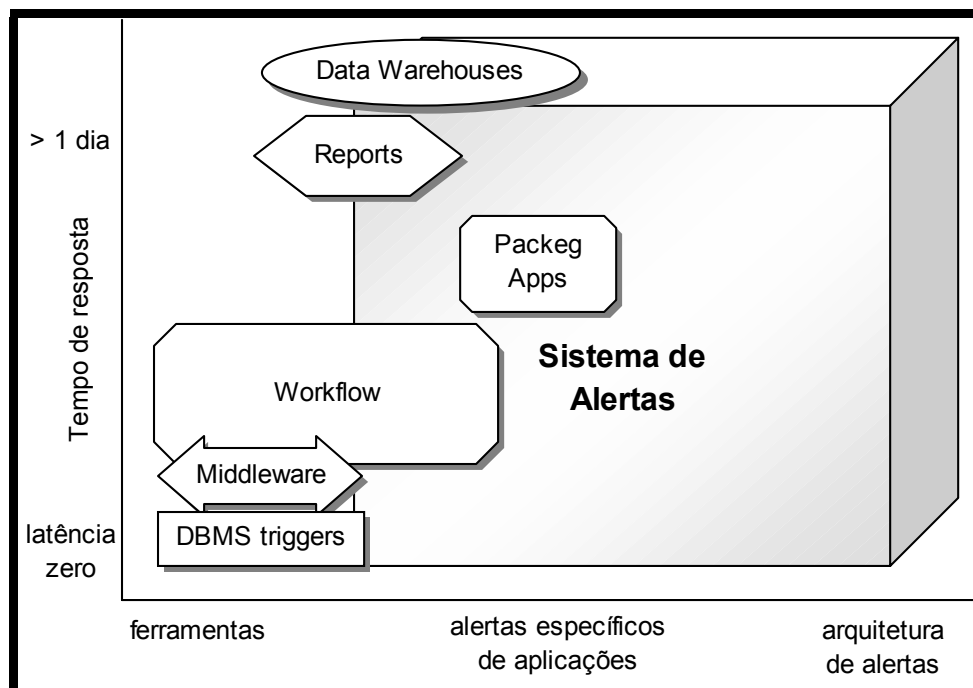


FIGURA 2.2 - Arquitetura de um Sistema de Alertas de Negócios [CAT 99]

Estes sistemas de alertas melhoram as transações comerciais das empresas. Os gerentes estão sempre atualizados, recebendo alertas dos eventos em todas as áreas como: problemas do cliente, execução do serviço, eficiência, retornos financeiros e riscos. As empresas que usam os alertas estão percebendo um retorno de investimento, redução de custos e impedindo problemas e economizando tempo gerencial.

### 2.1.1 Sistemas Comerciais para Gerenciamento de Projetos

Atualmente, já existem alguns sistemas comerciais para gerenciamento de projetos de uma empresa que utilizam a tecnologia de alertas. Exemplos desse tipo de sistema são apresentados a seguir.

#### 2.1.1.1 *Artemis Active Alert* - Gerenciamento Inteligente de Projetos

*Artemis Active Alert* é uma aplicação de monitoramento de banco de dados, baseada em agentes inteligentes para o gerenciamento de projetos, utilizado para projetos de aplicações financeiras, especialmente para administração dos recursos, negócios executivos e financeiros [ART 99].

Em um sistema, denomina-se agente cada uma de suas atividades ativas. Um agente raciocina sobre o ambiente, sobre os outros agentes e decide quais objetivos deve perseguir, quais ações tomar, etc. O termo ativo, tem como objetivo sinalizar que o conceito de agente não corresponde a noções estáticas como módulos, conjunto de regras e bases de conhecimento, sem que estas estejam intimamente relacionadas com um mecanismo de controle para a sua ativação [ALV 97].

Esse sistema de gerenciamento de projetos monitora o banco de dados da empresa para identificar acontecimentos ou tendências que podem causar impacto na organização dos projetos. Quando um evento ocorre ou uma direção contrária do

previsto é identificada, o sistema notifica o usuário responsável via *e-mail*, *fax*, celular, *pager* ou através de uma página HTML no momento da sua ocorrência.

As exceções que devem ser monitoradas são definidas pelos usuários através de limites estabelecidos ou regras de negócio relacionadas à programação do projeto, custos e recursos, como por exemplo: horário de falhas, desempenho de recursos, conflitos de recursos, problemas de custo, etc.

Diversos tipos de alertas podem ser disparados dentro de uma organização e o sistema garante a entrega do alerta esperado para cada pessoa.

O sistema de alertas oferece às organizações uma maneira mais eficiente de administrar seus projetos, aumentando a sua vantagem competitiva. Ele remove a necessidade de que pessoas analisem os dados na procura de exceções. Com isto, o negócio se torna mais ativo, respondendo rapidamente às mudanças de condições.

O sistema apresenta os seguintes componentes:

***Artemis Active Alert Administrator*** - permite a criação e sincronização das regras de negócios e a definição dos endereços de *e-mail*, número do celular e *pager*. O Administrator constrói as regras de negócio pela geração de código SQL, usando uma ferramenta própria que facilita o processo de construção de consultas complexas. Logo que a consulta tenha sido criada e testada, o administrador define o receptor e o meio de recebimento da mensagem. As regras de negócio podem ser agendadas para serem executadas em qualquer intervalo de tempo.

***Artemis Active Alert Server*** - ferramenta de consulta otimizada e entrega de mensagens que monitora os bancos de dados a espera dos eventos definidos. Quando um evento ocorre, a mensagem apropriada é enviada para a pessoa responsável.

***Artemis Active Alert Trend Analysis*** - armazena, em um banco de dados, as exceções ocorridas, permitindo aos usuários a identificação das partes do projeto que estão operando eficientemente e as que estão apresentando problema. É possível criar alertas após a análise destes dados.

#### 2.1.1.2 *Categoric Alerts'99* - Monitoramento de transações comerciais

O *Software Categoric Alerts* foi desenvolvido pela empresa Categoric Software Corporation, situada na Califórnia [CAT 99]. Permite o monitoramento de transações comerciais. Através dos sistemas de informações de uma organização, o sistema notifica os gerentes, empregados, provedores, e sócios da ocorrência de eventos críticos.

Os usuários configuram o programa para a emissão de alertas quando ocorrem eventos específicos e informam a frequência e método de notificação que pode ser: *e-mail*, *Web*, *fax*, *pager*, ou telefone celular.

Por exemplo, um comerciante financeiro pode necessitar de um alerta quando ocorrerem altas em um certo valor, enquanto que um representante de vendas pode receber um aviso sempre que uma ordem de um cliente importante for recebida. Isso permite às organizações responder rapidamente aos problemas e tomar melhores decisões, baseadas nessas informações que são apresentadas em tempo real, isto é, na



hora em que ocorre o problema o responsável é notificado.

A Carpetland S.A. Internacional, uma rede de 160 lojas ao longo da Europa, utiliza o *software Categori* para monitorar transações de negócio. Por exemplo, os gerentes são alertados quando representantes de vendas oferecem descontos excessivos ou créditos para clientes considerados "arriscados". Com o *software*, os gerentes são informados e podem, se necessário, rescindir as transações.

A arquitetura do sistema é composta dos seguintes itens, conforme figura 2.3.

**Alert Server** - servidor central de processamento dos alertas. O servidor processa as regras associadas com os alertas, incluindo acesso a banco de dados específicos, e no momento que ocorre um evento crítico, envia a mensagem para a pessoa responsável.

**Alert Administrator** - oferece recursos para construção e gerenciamento dos alertas.

**Alert Trender** – guarda as informações sobre os alertas gerados com o intuito de identificar tendências dentro de uma empresa. Ex. a maioria das pessoas que compra o produto do tipo A, também compra o produto do tipo B.

**Alert Publish and Subscribe** - permite aos usuários identificar os tipos de alertas desejados através de uma página *Web*.

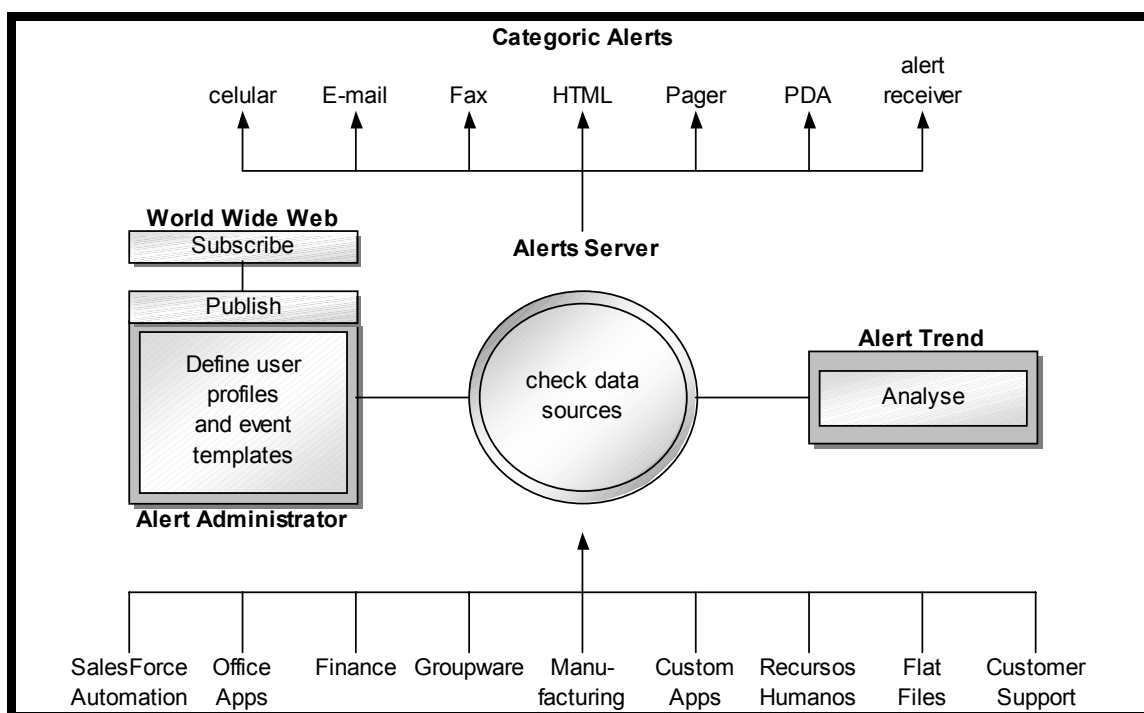


FIGURA 2.3 - Arquitetura do Sistema Categori Alerts [CAT 99]

## 2.2 Sistema de Alertas na Área Médica

Uma das aplicações de sucesso da Inteligência Artificial na Medicina é o desenvolvimento de sistemas automáticos de emissão de alertas e lembretes ao profissional da área médica, acoplados a bancos de dados de pacientes (registro médico computadorizado) [SHA 2000].

Um sistema de saúde com módulos de alertas monitora os bancos de dados, detectando situações de exceção e emitindo os alertas correspondentes [BRA 89]. Os alertas gerados devem ser úteis tanto para o corpo clínico quanto para a administração dos hospitais [AKI 98]. Os alertas que tipicamente interessam aos profissionais da área médica são críticas e sugestões de procedimentos, informações sobre resultados de exames, alertas para situações de risco, entre outros. Para a administração hospitalar são úteis alertas para incidência de determinadas doenças, para informações sobre procedimentos inadequados, sobre uso indevido de recursos, e muitos outros.

A quantidade e a qualidade da informação disponível para os profissionais da saúde em um ambiente hospitalar exercem significativa influência nos seus desempenhos. Estudos realizados comprovam que o comportamento desses profissionais é influenciado pela presença de um sistema que os auxilie emitindo alertas e sugestões de procedimentos [AKI 98].

Muitos sistemas e metodologias têm sido propostos. Exemplos de sistemas desse tipo, em operação rotineira há muitos anos, são os sistemas "*HELP*", desenvolvido na Universidade de Utah, EUA [HAU 94] [KUP 90] e o *Regenstrief Medical Record* (RMR), desenvolvido na Universidade de Indiana, EUA [MCD 83]. Esses sistemas funcionam da seguinte maneira: ao ser solicitado um registro computadorizado de um paciente a ser atendido, o programa de alerta/lembrete é acionado. Os conteúdos dos campos do registro médico são usados para acionar regras de produção, que, por sua vez, originam mensagens, tais como: necessidade de realização de exames preventivos, alertas sobre condições clínicas críticas, necessidade de se evitarem interações medicamentosas, etc.

Até o presente, todos os sistemas em existência se baseiam em bases de conhecimentos utilizando a técnica lógica/heurística, as quais são codificadas em linguagens seminaturais próprias. Essa abordagem é interessante, pois permite codificar o conhecimento médico qualitativo e quantitativo nas áreas de prognóstico, diagnóstico, conduta e planejamento, como, por exemplo: SE Potássio < 240 mEq/ml E Medicação (inclui insulina) ENTÃO Aviso "Cuidado, paciente com potássio reduzido, controlar o nível de insulina sendo administrado" [SAB 95].

Para tanto, é necessário, a existência de um Sistema de Informatização Hospitalar que visa facilitar a comunicação, integrar a informação e coordenar a ação conjunta dos múltiplos profissionais de saúde, além de possibilitar a organização e o arquivamento dessas informações. Além disso, o sistema deve também permitir fácil acesso ao banco de dados de pacientes para consulta, estudos e atualização de dados (exames, evolução clínica, medicamentos utilizados, solicitação de internações e altas, agendamento, etc.), bastando que o usuário credenciado possua uma senha de acesso aos vários níveis de informações.

Um Sistema de Alerta funciona continuamente entre o monitoramento de dados clínicos e o registro dos pacientes. Eles são desenvolvidos para testar tipos específicos de dados conforme critérios pré-definidos. Se houver um cruzamento entre um dado clínico e um critério, o sistema irá emitir um alerta, podendo informar as providências médicas que devem ser tomadas.

A tabela 2.1 apresenta um exemplo de correlação entre condição de alerta e critério para resultados de exames de laboratório [HAU 94]:

TABELA 2.1 - Correlação entre condição de alerta e critério

Condição de Alerta	Critério
Hiponatremia	Na+ < 120 meq/litro
Hipernatremia	Na+ > 155 meq/litro
Hipokalemia	K+ < 2,7 meq/litro
Hiperkalemia	K+ > 3,3 meq/litro
Hipoglicemia	Glicose < 45 mg%
Hiperglicemia	Glicose > 500 mg%

A implantação do Sistema de Informatização Hospitalar com módulos de alertas traz vários benefícios à instituição hospitalar, que necessita de um sistema de informações de dados ágil, seguro e dinâmico, face ao grande número de pacientes e, conseqüente ao volume de dados. Atualmente não é mais admissível um sistema manual de informação que, além de ser lento, possibilita uma enorme margem de erros.

Segundo AKIKUBO, é possível enumerar algumas vantagens na implementação desse tipo de sistema [AKI 98]:

**Aumento do Controle de Faturamento dos Serviços Hospitalares:** viabiliza o controle de atendimento de pacientes concomitante com o próprio ato de atendimento, assegurando a operação de faturamento imediatamente após a conclusão do processo, absolutamente isento de erros de apresentação;

**Eficiência no controle de Infecção Hospitalar:** possibilita ao administrador otimizar o uso de antimicrobianos, acompanhando a internação, prescrição e alta de pacientes com infecção suposta ou confirmada;

**Agilização no Processo de Agendamento de Cirurgias:** permite o controle físico e administrativo do Centro Cirúrgico, onde, através do uso otimizado das salas de cirurgias e de recuperação pós-operatória, o administrador poderá controlar de forma mais eficaz este setor do hospital.

**Racionalização no Controle do Centro Radiológico:** obtida através das facilidades apresentadas no controle das atividades dos serviços de diagnose e terapêutica que se apóiam na interpretação de imagens. Possibilita o agendamento de exames, emissão de laudos, controle de consumo de materiais, entre outros;

**Gerenciamento e Controle no Agendamento de Exames:** permite, além do

registro de exames e da emissão de laudos, as informações gerenciais para avaliação de desempenho da unidade, bem como dos funcionários, da produtividade, do consumo de materiais, etc.;

**Eficiência no Controle de Estoque e Farmácia:** rigoroso controle de materiais e medicamentos do estoque, permitindo ao administrador o gerenciamento de toda a movimentação dos materiais e medicamentos, bem como o controle por prazo de validade, movimentação de psicotrópicos e geração de relatórios gerenciais e contábeis;

**Agilização da Entrega de Contas Hospitalares:** possibilita a garantia de que toda internação tenha um faturamento correspondente, permite o preenchimento de dados durante a internação e, conforme a ocorrência dos procedimentos, o fechamento da conta no mesmo instante em que o paciente der alta no hospital;

**Correta Avaliação Financeira:** um rígido controle sobre as contas dos diversos convênios possíveis, permite uma posição dos custos reais de cada internação e procedimento.

O Projeto SIDI [LEA 98], desenvolvido no Instituto de Informática da Universidade Federal do Rio Grande do Sul, fornece informações de melhor qualidade aos profissionais da área da saúde, facilitando assim, o processo decisório. A construção de um sistema de alertas inteligentes foi proposta por LEÃO [LEA 98] para permitir que situações de exceção sejam detectadas, gerando informação de utilidade para os usuários do sistema. Através de um monitoramento das bases de dados, é possível detectar estas situações gerando os alertas correspondentes.

Os alertas são utilizados na geração de diversos tipos de informação. Entre elas:

**Informação relevante:** os módulos de alerta detectam a ocorrência de um evento que gera uma informação de interesse para um determinado grupo de usuários ou sistema. Exemplo: o resultado do exame de um paciente é incluído na base de dados. Um módulo de alerta detecta este evento e informa ao médico responsável sobre o resultado do exame do seu paciente.

**Sugestões de procedimento:** os módulos de alerta detectam a ocorrência de um evento e sugerem uma ação apropriada em resposta àquele evento. Exemplo: o resultado do exame de um paciente acusa que ele não está respondendo adequadamente ao tratamento. Um módulo de alerta sugere ao médico responsável que altere as doses dos medicamentos.

**Situações de risco:** os módulos de alerta detectam situações de risco na base de dados e geram uma mensagem de advertência. Exemplo: um módulo de alerta detecta interação medicamentosa na lista de medicamentos que determinado paciente está recebendo. Uma mensagem de advertência é gerada.

**Deteção de inconsistências:** os módulos de alerta detectam inconsistências nos dados da base e geram uma mensagem de alerta. Exemplo: um módulo de alerta detecta na base uma solicitação de parto para um paciente do sexo masculino.

A seguir é mostrado um exemplo de alerta implementado no Projeto SIDI [NAR 97].

Exemplo 1: Um alerta que detecta uma situação de risco.

**Maintenance:**

**title:** Verifica alergia a penicilina;;  
**version:** 1.00;;  
**institution:** Columbia-Presbyterian Medical Center;;  
**author:** George Hripesak, M.D.;;  
**specialist:**;;  
**date:** 1991-03-18;;  
**validation:** testing;;

**Library:**

**purpose:** Quando penicilina é prescrita, verifica se existe alergia;;  
**explanation:** é invocado quando uma medicação com penicilina é requisitada. Um alerta é gerado porque o paciente tem alergia a penicilina registrada.;;  
**keywords:** penicilina; alergia;;  
**citations:** ;;

**Knowledge:**

**type:** data-driven;;  
**data:**  
 /\* Um pedido por penicilina invoca este teste \*/  
 penicillin\_order := event {medication\_order where class = penicillin};  
 /\* Encontra alergias \*/  
 penicillin\_allergy := read last {allergy where agent\_class = penicillin};  
 ;;  
**evoke:** penicillin\_order;;  
**logic:**  
     if exist(penicillin\_allergy)then  
         conclude true;  
     endif;  
 ;;  
**action:**  
     write "Perigo, o paciente tem a seguinte alergia a penicilina documentada:"  
         || penicillin\_allergy;;  
**urgency:** 50;;

**end:**

## 2.3 Sistema de Alertas no Gerenciamento de Redes

Atualmente, as redes de computadores e os seus recursos associados, além das aplicações distribuídas, têm se tornado fundamental e de tal importância para uma organização, que elas basicamente devem operar constantemente. Isto significa que o nível de falhas e de degradação de desempenho, considerados aceitáveis, está cada vez mais diminuindo, sendo este nível igual a zero - dependendo da importância da rede para um instituição [MEE 99].

A área de gerência de redes foi inicialmente impulsionada pela necessidade de monitoramento e controle do universo de dispositivos que compõem as redes de comunicação [TAR 96].

A chave para um perfeito gerenciamento de redes encontra-se nas informações que podem ser obtidas da mesma. Assim, precisa-se ter a possibilidade de visão e manipulação dessas informações, além de poder compartilhá-las com outras pessoas que trabalhem nas mesmas funções e/ou em colaboração a fim de obter-se fundamentação e evolução nos trabalhos realizados.

Com esta crescente necessidade de gerenciamento, fez-se necessário o desenvolvimento de ferramentas que permitem avaliar o desempenho de uma rede de forma ágil, assim como tomar as atitudes cabíveis tão logo perspectivas de falhas surjam. Através dos sistemas de monitoramento, o administrador da rede não precisa se preocupar em coletar informações manualmente, pois é avisado dos problemas que estão ocorrendo.

De acordo com TAROUÇO em [TAR 96], estas ferramentas geralmente, apresentam os seguintes módulos:

**Módulo de Monitoramento** - registra e acompanha problemas detectados na rede.

**Módulo de Alertas** - permite ao administrador da rede especificar condições de exceção sobre as quais deseja ser avisado ou para as quais um registro de problemas deva ser criado.

**Módulo Processador de Eventos** - integra os módulos de monitoramento e alertas. Avalia as informações, distinguindo os eventos que não são dignos de registro daqueles que precisam ser sinalizados para o administrador da rede.

Dessa forma, o administrador é avisado quando ocorre algum problema na rede e realiza a ação corretiva adequada para manter o sistema seguro. A principal característica desses sistemas é buscar situações críticas com base nas informações que possuem a sua disposição.

A monitoração da rede é efetuada em objetos e máquinas definidas pelo administrador e, após a coleta destas informações, o sistema gera eventos. Os eventos são verificados para descobrir quais deles têm a necessária gravidade de se tornarem um alerta. Quando alertas são gerados, o administrador fica sabendo que um problema ocorreu com a rede e que deve ser corrigido o mais rápido possível. Dessa forma, o administrador fica livre para realizar outras tarefas enquanto o sistema efetua as

monitorações definidas por ele.

À medida que eventos são gerados pelo sistema, eles são passados para o processador de eventos, onde são submetidos a uma base de regras e então é verificado se há necessidade de se gerar um alerta, ou não.

Em geral, esse processo é realizado por sistemas especialistas. Um sistema especialista é um *software* de solução de problemas que incorpora conhecimento especializado em um domínio limitado para fazer um trabalho, que geralmente é feito por uma pessoa bem treinada [CRO 88]. Sistemas especialistas aplicam técnicas de Inteligência Artificial, tendo como núcleo uma base de conhecimento, que consiste numa coleção de fatos, definições e regras heurísticas, adquiridas diretamente do especialista humano [TAR 96].

As regras são construídas de acordo com a experiência prática dos administradores da rede. Essas regras são implementadas na forma de comandos *if.. then.. else* e estão voltadas à gerência de desempenho e de falhas. A seguir, é apresentado um exemplo de regra que verifica o estado de uma interface [TAR 96].

IF (ifOperStatus = down)

THEN IF (ifAdminStatus = up)

THEN IF (após três tentativas em intervalos de cinco minutos a interface continuar *down*)

THEN (manda um aviso ao administrador do sistema através de um *e-mail* dizendo que a interface de determinada máquina está *down* e a hora que isso foi observado)

onde :

ifOperStatus - estado operacional da interface

idAdminStatus - estado administrativo da interface

Eventos são indicadores de situações anormais detectadas a partir do processo de monitoração [SCH 2000]. Alertas são indicadores de que estas situações anormais são críticas e representam algum tipo de ameaça à manutenção do nível de qualidade dos serviços da rede [SIL 97].

O sistema de alertas coleta e analisa estatisticamente as informações obtidas, auxilia na monitoração dos níveis de serviço e, por conseqüência, provê meios para a gerência de desempenho e de falhas.

Os alertas devem ser notificados contendo informações detalhadas sobre o evento ocorrido a fim de dar subsídios a uma ação por parte do gerente de rede. Segundo PINTO em [PIN 95], essas informações devem compreender:

- tipo de alarme;
- causa do alarme;

- severidade da ocorrência (indeterminada, crítica, maior, menor);
- serviço do usuário que detectou o alarme (mecanismo de segurança);
- usuário do serviço de segurança que gerou o alarme;
- fornecedor do serviço;
- breve descrição de dados adicionais sobre o evento.

Exemplo de um alerta e suas possíveis causas [PIN 95]:

- violação da integridade da mensagem;
- informação duplicada;
- informação ausente;
- informação fora de seqüência;
- informação não esperada.

Através de monitorações contínuas, é possível encontrar a utilização corrente dos enlaces e segmentos da rede, identificar áreas de possível congestionamento, isolar altas taxas de erros e examinar os padrões de tráfego da rede. Cada um desses valores pode auxiliar o gerente de rede a assegurar se a rede está, ou não, atingindo as expectativas de seus usuários.

A medida que as monitorações são efetuadas, elas são armazenadas em um banco de dados. Posteriormente, utiliza-se o sistema para monitorar determinados objetos por um período de tempo e, quando algo estiver fora do “normal”, isto é, quando os dados consultados não estiverem de acordo com a base de dados, um evento é criado podendo ser gerado um alerta ou não. Quando o sistema de alertas consulta a base de regras e verifica que um alerta deve ser enviado ao usuário, ele também consulta a base de monitoramento para verificar se o mesmo problema já aconteceu anteriormente e qual solução foi utilizada para corrigi-lo. Desta forma, o sistema pode aprender maneiras diferentes para corrigir um mesmo problema.

## 2.4 Sistema de Alertas em Biblioteca Digital

A publicação eletrônica torna-se cada dia mais popular e a quantidade de documentos e publicações científicas disponíveis na *Web* vem aumentando consideravelmente. O modo clássico de procura de informação em WWW por recuperação e navegação torna-se cada vez mais ineficiente, o que gera uma grande necessidade para o uso de sistemas de alertas [FAE 98] [HIN 99]. Tais sistemas mantêm os usuários informados entregando automaticamente os conteúdos de acordo com as necessidades do cliente [ROS 97], que são definidas em um perfil de interesses [MCC 97].

Algumas técnicas de informação para os usuários de eventos interessantes estão sendo desenvolvidas, fazendo com que os leitores não precisem procurar regularmente pela informação que eles necessitam [BRN 97]. Muitas delas utilizam o conceito de descrição de perfil, onde o cliente define os seus interesses. Isto assegura que cada cliente somente receba notificação sobre eventos que estão dentro do seu interesse. O



sistema de alertas compara os eventos ocorridos (por exemplo, chegada de documentos novos) com os existentes nos perfis e se necessário, informa os usuários interessados [CRE 97].

Alguns serviços como *Ariadne* [ARI 99] ou TEL-HP [THE 99] oferecem a técnica de perfil. Os clientes são notificados no momento em que os documentos novos chegam e são compatíveis com o perfil estabelecido. Serviços de perfil têm também sido oferecido por banco de dados bibliográficos, alguns com linguagens de definição de perfil, como o *CompuScience* [COM 99]. Algumas editoras também oferecem serviços de alertas para informar sobre conteúdo novo para os assinantes. Isto é oferecido pelo *Springer* com *Spring Link Alert* [SPR 99] e *Elsevier* com *Contents Direct* [ELS 99] [ELS 99a]. *Elsevier* oferece um serviço de notificação baseado nos canais *Web (Science Channel)*. O usuário pode escolher um tópico de uma lista de palavras chaves e assinar os jornais examinados pela palavra-chave. Notificação sobre troca de conteúdo nos canais chega por *e-mail* ou através de um ícone no *desktop*. O *SwetScan* [SWE 96] oferece um serviço de anúncio para as bibliotecas. Isto oferece uma entrega eletrônica da alteração do catálogo. Bibliotecas inscritas podem definir seus perfis de interesse com dados bibliográficos.

Segundo HINZE em [HIN 99a], o grande número de diferentes serviços possui algumas desvantagens como:

- O usuário precisa se registrar em cada fornecedor de serviço, o que pode ocasionar a perda de alguns fornecedores de informação importante;
- A situação exige que o usuário aprenda os diferentes serviços de fornecedores;
- Muitas das ferramentas somente enviam um *e-mail* com informações irrelevantes para o usuário;
- Personalização é muito raro. Os serviços que dispõem desta facilidade, geralmente forçam o usuário a fornecer um dado pessoal para muitas pessoas ou diferentes companhias.

Todos esses problemas conduzem para a instalação do serviço de alerta. Com isso, o usuário pode assinar e ter acesso aos recursos de vários esquemas de notificação de diferentes serviços.

Na figura 2.4 é apresentada a arquitetura completa do Serviço de Alerta para Bibliotecas Digitais proposto por HINZE [HIN 99b]:

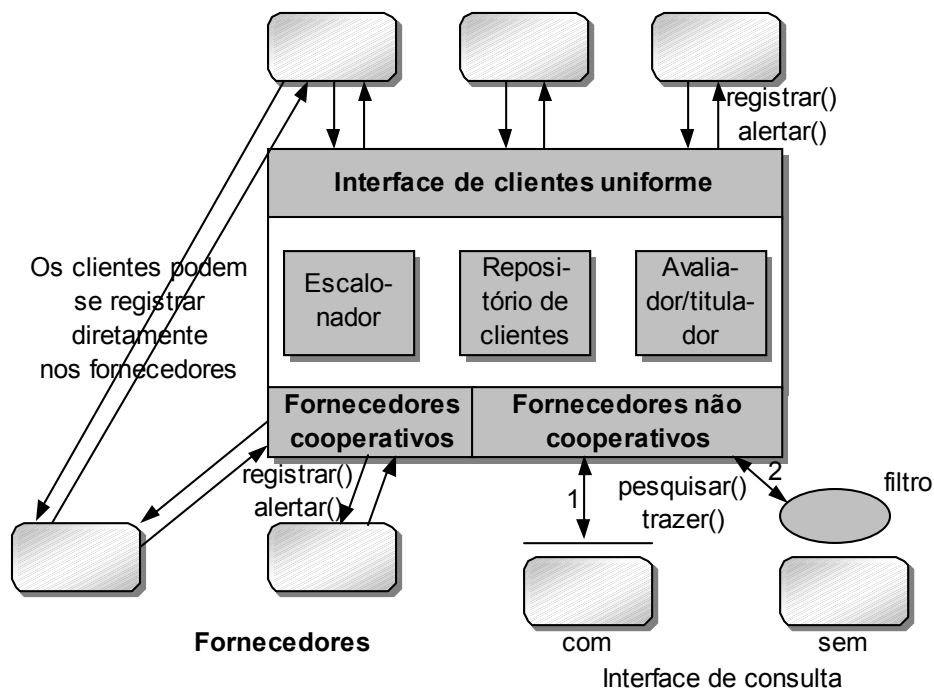


FIGURA 2.4- Arquitetura do Serviço de Alertas em Biblioteca Digital [HIN 99b]

Existem fornecedores cooperativos e não-cooperativos. Fornecedores cooperativos notificam o serviço de alertas da ocorrência de eventos em seu repositório, enquanto que fornecedores não-cooperativos não oferecem qualquer notificação, então *wrappers* devem ser implementados para consultar o repositório do fornecedor [HIN 99b]. Os *wrappers*, na figura, estão representados pelos itens (1) e (2). Isto fica a escolha da técnica apropriada para a notificação entre o fornecedor e o serviço de alerta e entre o serviço e os usuários.

### **3 Processo de Avaliação nos Ambientes de Ensino a Distância via *Web***

Atualmente, existem vários ambientes de ensino-aprendizagem disponíveis na *Web*. A maioria deles trata da criação de cursos a distância utilizando a *Web* como ferramenta de disponibilização de material para os alunos. Estas ferramentas apresentam poucas alternativas na hora de avaliar o aluno ou de fornecer informações sobre o seu aprendizado.

A fim de apoiar o trabalho desenvolvido, realizou-se uma pesquisa na busca de padrões e metodologias de avaliação utilizadas por outras instituições. A seguir serão descritos alguns dos mais importantes ambientes de ensino a distância que implementam o processo avaliativo do ensino-aprendizagem, mostrando a forma como cada um trata a avaliação do aluno à distância, e os mecanismos que utilizam nestes procedimentos.

Observa-se que o processo de avaliação, nestes ambientes, normalmente é feito através de provas e testes. Desse modo, há uma necessidade de alternativas, para que, mesmo à distância, possam obter dos alunos informações complementares, que trarão um melhor entendimento sobre a real situação do aprendizado. A utilização de um sistema de alertas seria uma opção que se adequaria para os devidos fins.

#### **3.1 WebCT**

O *WebCT* foi desenvolvido pelo Departamento de Ciência da Computação da Universidade de British Columbia no Canadá.

O *WebCT* [GOL 97] é um sistema que facilita a criação de ambientes educacionais baseados na *Web*. Podendo ser usado para criar cursos ou simplesmente disponibilizar material de cursos já existentes. Possui ferramentas que facilitam o aprendizado, comunicação e colaboração e ferramentas administrativas que auxiliam o professor no decorrer do curso.

Possui os seguintes componentes básicos:

- Quadro de avisos para alunos e professores
- Correio eletrônico
- Ferramenta de bate-papo
- Auto-avaliação do estudante
- Arquivo para busca de imagens utilizadas em aulas
- Glossário indexado com links
- Testes on-line com tempo
- Ferramenta para anotações dos alunos nas aulas
- Boletins para aluno, comparando suas notas com a turma

- Quadro de anotações compartilhado para trabalhos em grupo
- *Timed On-Line Quizzes*. Para criação dos testes, e com a correção automática, e geração de notas e desempenho dos alunos.
- Gerenciamento de alunos
- Customização de cursos

### 3.1.1 Avaliação no *WebCT*

O processo de avaliação do *WebCT* funciona da seguinte maneira:

O professor deve criar categorias que podem se referir ao nível de dificuldade de cada prova ou a partir do conteúdo a qual está relacionada. Posteriormente as provas serão classificadas nestas categorias. Após criar as categorias, o professor deve escolher o tipo de questão que incluirá na prova [DES 98]. O sistema *WebCT* oferece cinco tipos de questões: múltipla escolha, relacionar colunas, respostas curtas, de calcular e discursivas.

Nos quatro primeiros tipos, as questões são corrigidas automaticamente pelo *software* a partir da resposta correta especificada pelo professor no momento da elaboração da prova. As questões discursivas devem ser corrigidas pelo professor.

No momento da edição de cada questão, o professor deve preencher um campo com a resposta certa, para que na correção automática o *software* reconheça a resposta, e também para facilitar a correção feita pelo professor, no caso de questões discursivas, que pode optar por tornar visível para o aluno a resposta correta na entrega da prova.

Cada questão deve ser pontuada para que o professor, ou o *software*, atribua nota aos alunos de acordo com o número de acertos.

Após finalizar a elaboração da prova, o *WebCT* permite ao desenvolvedor habilitá-la e desabilitá-la conforme sua necessidade, assim como permite configurar o tempo de duração, número de tentativas para a resolução da prova e a colocação de mensagens ao final da mesma. Existem também opções de como mostrar somente a nota, ou mostrar a nota e as respostas.

O *WebCT* permite colocar uma determinada duração para a prova e o número de tentativas.

Questões de múltipla-escolha podem ser colocadas em algumas páginas do curso, e uma explicação pode ser anexada, indicando porque a resposta estava incorreta ou dando informações adicionais. Também tem-se a opção de perguntas *on-line*, que são feitas enquanto o aluno está acessando o curso, devendo a resposta ser dada na hora. Sempre é dado um *feedback* imediato ao aluno, e cada um tem acesso as notas que recebeu em todas as atividades já realizadas.

O professor pode corrigir questões discursivas, quando houverem, e visualizar todas as informações referentes a avaliação como, data e hora do início e término da prova, tempo de execução da prova, e outros.

O gerenciamento dos alunos é de responsabilidade do professor, o qual pode

fazer uma análise detalhada do histórico do aluno no curso para visualizar os seguintes dados: data do primeiro acesso, data do último acesso, número de artigos lidos e postados, nome completo do aluno, as páginas visitadas, o tempo de acesso de cada página, sendo também possível controlar a sua frequência. Esta é uma vantagem do *WebCT* consideravelmente importante, pois permite fazer um controle de frequência dos alunos e de suas participações em outras atividades, o que pode ser importante no momento da avaliação.

Durante a resolução da prova é mostrado ao aluno um cronômetro com o tempo limite para a realização da mesma.

Outra vantagem, é que o aluno pode visualizar sua prova após ser corrigida. Desta forma, o aluno tem a possibilidade de fazer uma auto avaliação, de si próprio e do professor, para que diante de suas considerações possam identificar os pontos que foram absorvidos e os que devem ser reforçados, assim como identificar se os seus objetivos estão sendo alcançados.

### 3.2 Virtual-U

Virtual-U é uma ferramenta comercial desenvolvida pela Simon Fraser University, como parte de projetos de educação a distância do governo canadense [WHA 98]. Possui um conjunto de ferramentas integradas baseadas na *Web* que tem por objetivo a criação de cursos on-line. Ele é intencionalmente projetado para permitir e facilitar discussões assíncronas, aprendizado cooperativo (*groupware*), e construção de conhecimento. O *software* é utilizado em uma grande variedade de áreas, principalmente para a construção de cursos pós-secundários e treinamento *on-line* e está sendo correntemente testado em um grupo de universidades e escolas tais como York University, Université Laval, University of Waterloo, Canadian Union of Public, Bank of Montreal, entre outros.

O conjunto de ferramentas do Virtual-U inclui ferramentas para projeto de cursos, discussões e apresentações em salas de aula virtuais, tratamento de recursos dos cursos, e gerenciamento da evolução das salas de aula.

Virtual-U inclui os seguintes componentes:

- ***Vgroup***: sistema de conferência que dá aos instrutores a capacidade para criar grupos e definir para eles tarefas e objetivos;
- **Ferramenta para estruturação e editoração do curso**: permite que instrutores criem cursos completos sem que o usuário (professor, instrutor) tenha conhecimentos sobre programação. *Templates* ajudam o instrutor na criação do curso. A ferramenta coloca o curso automaticamente na *Web*;
- ***GradeBook***: gerencia a base de dados de estudantes para cada curso disponível com Virtual-U. O *GradeBook* mostra o desempenho dos estudantes através de gráficos e textos, incluindo tabelas identificando a evolução nas atividades de cada aluno;
- **Ferramentas para administração**: auxilia administradores de sistema no que diz respeito a instalação e manutenção do Virtual-U.

O ambiente Virtual-U permite basicamente [FIS 98]:

- Criação de páginas *Web* de cursos;
- Estruturação de discussões interativas e atividades cooperativas entre alunos, professores e colaboradores externos;
- Criação de recursos compartilhados para disseminação do conhecimento;
- Manutenção e evolução de cursos;
- Avaliação através de exercícios e tarefas.

### 3.3 *TopClass*

O *TopClass* é um produto da WBT Systems, onde WBT significa Treinamento Baseado na *Web* (*Web-based training*). É um sistema de gerenciamento de ensino a distância que cuida de todos os aspectos no que diz respeito ao conteúdo, gerenciamento e entrega de material. Ele funciona como uma aplicação cliente/servidor em Intranet e Internet [WBT 98]. O produto vem sendo usado por muitas empresas e instituições de ensino e já começa a ser testado em empresas e instituições brasileiras. Corporações como a Dow Chemical, que possui milhares de funcionários que precisam ser treinados, já adotaram o uso do *TopClass*, assim como a Universidade de Nova Iorque.

Com o *TopClass* pode-se fazer:

- Desenvolvimento de conteúdo dos cursos;
- Gerenciamento dos usuários que estão fazendo os cursos (alunos), ou coordenando alguma sala de aula (instrutores);
- Administração da base de dados de conteúdo.

Gerenciar o treinamento ou ensino a distância é mais do que só criar conteúdo. É necessário controlar o acesso às informações, oferecer interação e colaboração instrutor-aluno, bem como, entre alunos rastrear o uso e avaliar competências.

O sistema controla o acesso, monitora o uso e coordena a comunicação entre os servidores da *Web* e outros aplicativos empresariais, a fim de oferecer características únicas de gerenciamento de treinamento ou de ensino [TOP 2000].

Conteúdo aberto - O *TopClass* suporta qualquer aplicativo compatível com a *Web*, possibilitando mídia, animação e vídeoconferência. Através do *TopClass Publisher e Assistants*, qualquer documento ou páginas já existentes na *Web*, podem ser automaticamente transformados em conteúdo de treinamento para o sistema.

Cursos personalizados - à medida que os alunos vão se submetendo aos testes, o conteúdo pode ser acrescentado ou removido automaticamente para preencher eventuais falhas de conhecimento. Com isso, os alunos que já aprenderam o assunto, podem avançar mais rapidamente. Os instrutores, se necessário, podem modificar o conteúdo direcionando-o para alunos específicos.

*TopClass* indica para o usuário, o *status* do material do curso definido para ele, através de mensagens do tipo novo, velho, lido ou não-lido. Assim, o aluno saberá facilmente que material deve rever, que material é novo, etc. O professor tem acesso a esse status para monitorar o progresso do aluno, no que se refere ao acesso e ao conteúdo.

Biblioteca de Módulos de Treinamento - As classes podem compartilhar o material dos cursos, seja o curso integral ou apenas o conteúdo de determinada mídia ou página. O *TopClass Server* organiza os cursos para fácil localização e navegação através da biblioteca de módulos e quando um módulo é atualizado, todos os cursos que o contêm terão acesso imediato ao novo material.

Biblioteca de Testes - A avaliação baseada em testes individualizados permite que novas tentativas possam ser feitas no caso de respostas incorretas, enquanto que a segurança pode ser mantida através de sorteio randômico que oferece a cada aluno um conjunto diferente de questões. O sofisticado roteiro de testes do *TopClass* permite que instrutores possam dar uma prova mais completa ou definir questões que avaliem a competência do aluno de forma mais eficiente.

Inscrição Aberta e Acompanhamento - A inscrição para turmas é feita automaticamente pelo *TopClass* e pode ser integrada a outros sistemas de gerenciamento de matrícula ou de aluno. No caso das empresas, de forma semelhante, as informações sobre acompanhamento podem ser exportadas para os sistemas de Recursos Humanos.

Grupos de Discussões - Os grupos de discussões podem ser ligados às classes ou a tópicos específicos, de forma livre ou com total moderação e controle de acesso, a fim de compartilhar apenas as informações e projetos considerados pertinentes pelo instrutor.

### **3.3.1 Processo de Avaliação no TopClass**

O sistema *TopClass* permite que o professor crie testes e exercícios que serão corrigidos pelo professor ou automaticamente pelo sistema. Os testes são criados a partir de um banco de questões armazenadas. Estes testes podem ser gerados randomicamente pelo sistema, criando-se provas diferentes para cada execução [WHA 98]. O instrutor pode definir que as questões sejam apresentadas uma por vez ou todas em uma página. Também pode ser configurado a permissão aos alunos de realizarem a mesma prova várias vezes.

O *TopClass* permite oito tipos diferentes de questões:

- *Text*: preenchimento de espaços em branco ou respostas abertas;
- *Pick One*: múltipla escolha com uma única correta;
- *Upload*: pedido de envio de um arquivo;
- *MCObs*: múltipla escolha envolvendo arquivos multimídia (som, imagens, animações);
- *MCA*: múltipla escolha com várias respostas corretas;

- *List matching*: ligação entre duas colunas;
- *ImageMap*: imagem clicável;
- *Boolean*: verdadeiro e falso.

O sistema *TopClass* permite que as provas sejam automaticamente corrigidas, com a pontuação definida pelo professor, ou enviadas para o respectivo professor corrigir. A correção automática inclui *feedback* definido pelo professor para cada questão, entre elas:

- **adicionar material instrucional**: fornecem ao aluno mais material para o estudo, como por exemplo uma página de reforço.
- **notificar professor**: envia uma mensagem ao professor indicando quais os alunos não atingiram o mínimo de nota aceitável nos testes.
- **remover material institucional**: permite aos alunos que atingiram notas altas pular certos conteúdos que foram programados pelo professor.

O sistema provê estatísticas relacionadas aos resultados de testes e provas *on-line*, como o acompanhamento das páginas visitadas pelo aluno.

### 3.4 *Web Course in a Box*

WCB – *Web Course in a Box* é um sistema desenvolvido pela Virgínia Commonwealth University para criação e manutenção de cursos na *Web* [ROD 2000]. Este ambiente permite a criação de páginas WWW para vários serviços tais como: material do curso, agenda e *home-page* pessoais, além de funções interativas como fóruns de discussão e exercícios auto-corrigíveis.

O WCB é composto por três partes, que são: um servidor para administradores do sistema, ferramentas de autoria para professores e instrutores, e páginas de cursos para alunos.

#### 3.4.1 Avaliação no WCB

O instrutor pode utilizar as ferramentas disponibilizadas pelo WCB para criar testes e exercícios de múltipla escolha ou resposta direta. Esta função é mais útil na realização de exercícios para a autoavaliação do aluno, e não como ferramenta de avaliação formal, pois não armazena os resultados em um banco de dados.

O sistema realiza a correção de todas as questões e pode fornecer os resultados a cada resposta feita pelo aluno ou somente após responder todas as questões. Além disso, o instrutor pode utilizar mensagens de resposta de acordo com o desempenho do aluno, bem como informações extra para questões respondidas erroneamente.

### 3.5 *Learning Space*

O Lotus LearningSpace Versão 3.0, desenvolvido pela empresa Lotus, é um sistema para aprendizado colaborativo, com recursos completos de mídia e gerenciamento de conhecimento [LOT 2000].



Possui cinco módulos de treinamento colaborativo que combinados oferecem um ambiente integrado que suporta o aprendizado baseado em equipes e facilitado pelo instrutor. Os módulos são:

***Schedule (agenda)*** – este módulo guia os participantes através das tarefas do curso. Ele define os objetivos e expectativas para os trabalhos do curso e atribui aos participantes recursos, testes, exames e auto-avaliações. Os instrutores podem criar cursos organizados por períodos ou módulos específicos, permitindo o aprendizado baseado em equipes ou em ritmo individualizado;

***MediaCenter (Central de mídia)*** – é uma base de conhecimentos compartilhados que contém e gerencia uma grande variedade de formatos de mídia. Ela pode funcionar como repositório para CDs de treinamento baseados em computador, integrar informação ao vivo de sites da Internet ou aumentar o conteúdo em uma variedade de formatos, desde textos simples até vídeos animados.

***CourseRoom (Sala do curso)*** – Um ambiente interativo e facilitado que permite às equipes trabalharem colaborando nas atribuições do curso. Permitindo que os participantes escolham níveis de privacidade, o CourseRoom suporta múltiplos níveis de comunicação dentro das equipes e com o instrutor.

***Profiles (Perfis)*** – Este banco de dados armazena a pasta individualizada do participante com as notas de avaliação e tarefas.

### 3.5.1 Avaliação no Learning Space

O *Lotus LearningSpace* apresenta um módulo de avaliação chamado de *Assessment Manager* – uma ferramenta para que os instrutores possam avaliar os alunos e dar-lhes um retorno de seu aproveitamento. Entre os recursos inclui-se:

- Uma abordagem revisada baseada em perguntas;
- Habilidade de distribuir aleatoriamente as perguntas, de modo que cada aluno receba uma avaliação exclusiva;
- Habilidade de controlar o tempo de avaliação, evitando que os alunos recebam várias avaliações;
- Classificação de perguntas por categoria para referência rápida e fácil na criação de avaliações;
- Habilidade de importar perguntas de bancos de testes;
- Habilidade de armazenar as avaliações corrigidas nas pastas dos alunos.

### 3.6 AulaNet

O AulaNet é um ambiente de software baseado na *Web*, desenvolvido no Laboratório de Engenharia de Software - LES - do Departamento de Informática da PUC-Rio, para administração, criação, manutenção e assistência de cursos a distância [AUL 2000]. Os objetivos do AulaNet são a adoção da *Web* como um ambiente educacional, a criação de uma transição viável da sala de aula convencional para a sala de aula virtual, oferecendo a oportunidade de se reusar o material educacional existente,

e a criação de comunidades de conhecimento.

Os cursos criados no ambiente AulaNet enfatizam a cooperação entre os alunos e entre aluno e professor e são apoiados em uma variedade de tecnologias disponíveis na Internet.

Como mecanismo para proporcionar a comunicação o AulaNet oferece:

- **Contato com o professor.** Permite que os alunos se comuniquem diretamente com o professor através de mensagens.
- **Grupo de interesse.** Permite a comunicação entre participantes através de *newsgroups*.
- **Grupo de discussão.** Permite a interação dos participantes através de listas de discussões. Com este mecanismo cada aluno do curso pode enviar uma mensagem diretamente para todos os outros alunos do curso.
- **Debate.** Mecanismo que permite a comunicação entre os participantes através de *bate-papo* ou de um *software* de videoconferência de baixo custo chamado *Cu-SeeMe*.

Os mecanismos de coordenação correspondem aos de acompanhamento do curso. São eles:

- **Agenda.** Mecanismo que permite a coordenação do agendamento de eventos, tais como debates no *bate-papo* e discussões na lista.
- **Notícias do curso.** Mecanismo que permite a coordenação do andamento geral do curso através de avisos.
- **Provas.** Mecanismo que permite a avaliação dos alunos através de provas.
- **Trabalhos.** Mecanismo que permite a avaliação dos alunos através de trabalhos.
- **Exercícios.** Mecanismo que permite a avaliação dos alunos através de exercícios.

Os mecanismos de cooperação, por sua vez, correspondem ao instrumental pedagógico. Estes devem ser utilizados durante a aplicação do curso e devem ser previamente selecionados pelo autor.

Existem nove mecanismos de cooperação no AulaNet: transparência, apresentação gravada, texto de aula, livro texto, bibliografia, demonstração, co-autoria de professor, co-autoria de aluno e *download*. Dentre eles, transparência, apresentação gravada, texto de aula, livro texto, bibliografia e demonstração são mecanismos para transmitir conteúdos. A co-autoria de professor e co-autoria de aluno são mecanismos para criar e desenvolver o curso de forma compartilhada com um outro professor ou com um aluno, respectivamente. *Download*, por fim, é um mecanismo para auxiliar o aluno a selecionar a aula que deseja assistir e a forma de receber os conteúdos que podem ser copiados para o computador e lidos pelo aluno mesmo estando desconectado da rede. Como suplemento às aulas, pode-se inserir conteúdos nos diversos serviços disponíveis, como aulas, livro texto, demonstração, bibliografia, agenda, notícias do curso, trabalho, exercício, prova e grupos de interesse

### 3.6.1 Avaliação no AulaNet

O AulaNet oferece três métodos de avaliação [AUL 2000]: prova, trabalho e exercício. Através destes, os alunos podem debater, criar projetos e compartilhar experiências. Através de provas o professor faz a avaliação formativa do processo de aprendizagem.

As provas são controlados por uma ferramenta de criação e correção automática, chamada *Quest*. Os objetivos do *Quest* são auxiliar o professor na criação e avaliação de provas, bem como proporcionar aos alunos a avaliação de suas respostas, seja de forma imediata ou através da correção pelo professor.

O propósito desta ferramenta é suportar o processo de avaliação educacional através da *Web* e possui as seguintes metas:

**Criar extensos relatórios:** *Quest* registra todas as informações sobre as respostas dadas pelos estudantes para criar um relatório compreensivo de resultados para o professor.

**Dar retorno para os estudantes:** em vez de reportar os resultados somente para o professor, *Quest* dá automaticamente retorno para os estudantes, então eles podem ter todas as informações sobre seu desempenho no percurso, em vez de perguntar ao professor.

**Ressaltar aspectos cognitivos da avaliação:** enquanto edita a questão, o professor escolhe quais domínios cognitivos ele quer avaliar.

O ambiente *Quest*, facilita a inclusão de questões, provendo uma interface gráfica acessível, tornando possível que qualquer professor, sem conhecimentos de computação, utilize o *software*.

Outras características da ferramenta incluem:

**Três estilos de apresentação de questões:** *Quest* é uma ferramenta de avaliação de múltipla escolha, mas provê três estilos de apresentação para a edição da questão: múltipla escolha (com cinco escolhas de resposta), verdadeiro ou falso e questão de lacuna.

**Facilidades de edição:** a ferramenta oferece facilidades na edição das questões como, por exemplo: mover a posição das questões, remover questões e inserir novas questões entre as já existentes. Isto é feito automaticamente devido a interface gráfica provida pelo *Quest*. *Quest* provê um caminho fácil de reedição das questões criadas, com a opção “Salvar Como”, para o professor não perder a questão original.

**Modificações nas questões:** enquanto edita a questão, o professor pode realizar algumas modificações. Estas modificações serão usadas quando for mostrada a estatística gerada pela correção automática da questão, enriquecendo estas estatísticas.

**Correção automática:** o processo de correção será feito automaticamente pela ferramenta. O professor somente necessita especificar a resposta correta no momento da edição.

**Resultados mostrados para os estudantes:** Como *Quest* é uma ferramenta de auto-avaliação, os estudantes não necessitam do professor para saber o resultado de seu desempenho.

**Armazenagem dos resultados para o estudante:** usualmente, os resultados de uma avaliação são armazenados pelo professor. *Quest* também armazena informações para serem mostradas em qualquer tempo para os estudantes. Estas informações incluem: grade com o total de pontos do teste, número de questões do teste, número de questões que o estudante respondeu corretamente, número de questões respondidas incorretamente, tempo transcorrido durante a resolução do teste, os tópicos anexados nas questões do teste, os tópicos das questões de estudantes respondidas incorretamente e uma opção para os estudantes revisarem seu teste, comparando a resposta certa com a que ele respondeu. A última característica está incluída na ferramenta para prover uma fonte de material de ajuda para outros estudantes, como estes resultados são formatados para mostrar os problemas de aprendizado.

**Estatísticas detalhadas para o professor:** o professor pode ter bastante informações derivadas da correção do teste. A principal idéia atrás destas estatísticas é prover ao professor uma maneira de visualizar o desenvolvimento de aprendizado dos estudantes, onde eles são colocados de frente as dificuldades e comparada ao desempenho do grupo com a sua, um por um, individualmente. Estas estatísticas incluem:

- Lista de grade: esta lista serve para ver as grades de cada estudante com uma opção de revisar as respostas dadas por todos os estudantes. Este é um caminho feito pela sala de aula convencional.
- Estatísticas baseadas em tópicos: um resultado mostrado como uma lista de tópicos dando ao professor um caminho para conhecer pontos do material instrucional dos estudantes que tem problemas.
- Estatísticas baseadas no domínio cognitivo: estas estatísticas mostram se o estudante está construindo seus conhecimentos corretamente. Expõe as habilidades cognitivas em que cada estudante mostra mais dificuldade. Após esta verificação, o professor pode redefinir o conteúdo do curso para resolver estes problemas, realmente ajudando o estudante a lembrar uma idéia, a entendê-la, colocá-la em prática, verificar as partes dos componentes, usar a idéia para construir outra idéia e avaliar seu uso.
- Estatística por questão: mostra os problemas que os estudantes tiveram em resolver cada uma das questões do teste. Isto pode ser usado para ver a média de dificuldade das questões, e como uma referência para as duas estatísticas anteriores mostrando se os estudantes tiveram problemas no entendimento do tópico.

- Resultados padrões: resultados padrões envolvem o cálculo dos resultados das médias e informações sobre quantos destes escores são altos e baixos. A média dos resultados devem ser concluídas com seu desvio padrão, e listando o maior e o menor resultado possível como o maior e o menor resultado atualmente recebido. Por exemplo: Existem 35 estudantes fazendo o teste. O resultado possível estende-se de 0 a 100, supondo que os estudantes resultem atualmente de 32 a 95, a média dos resultados será igual a 72 e o desvio padrão igual a 6.1.

### 3.7 *CyberQ*

*CyberQ* é um sistema de avaliação baseado na avaliação adaptativa e gerenciamento acadêmico de qualidade, desenvolvido para o grupo Apollo pela InterEd [TUC 95]. O centro deste sistema é um modelo conceitual que pode ser adaptado para algumas plataformas, incluindo DOS, Windows e Unix. O sistema gerencia um conjunto de atividades (das mais básicas as mais complexas), através de um *software* que está instalado em um servidor. O sistema funciona de forma transparente tanto para alunos como para professores, automatizando a administração de questões para avaliar a estrutura acadêmica, conteúdo, processo e resultados. Estas questões podem ser aleatórias ou apresentadas em alguns pontos no curso. Por exemplo, uma avaliação pode ser disparada quando o aluno completa uma lição ou a primeira vez na semana em que ele acessa o sistema.

Os componentes funcionais deste sistema são:

**Software de monitoramento de transações:** é o responsável por controlar o movimento eletrônico das informações, apresentando a quantidade de informações trocadas entre o professor e o estudante, entre o estudante e professor e entre o estudante e o seu grupo de trabalho.

Destes dados são extraídas diversas informações, como o tempo de resposta em consultas, o quanto um aluno está participando das aulas, etc.

**Software de análise sintática:** este módulo realiza a análise sintática do que os alunos escrevem. Pode ser feita através do uso dos índices, em que sílabas, palavras e comprimento de sentenças são contados. Uma estratégia muito usada é a combinação dos dados acima e sistemas inteligentes baseados em regras, combinados com técnicas avançadas de análise de sentenças para definir o estilo de cada estudante, podendo-se definir com tempo e exemplos suficientes, o progresso acadêmico.

**Análise de comentário:** este módulo registra as informações comentadas pelo aluno. Isto permite uma melhor avaliação do nível de instrução do aluno, do curso, do ambiente de aprendizado e dos serviços de suporte, já que as informações comentadas são mais fiéis ao que o aluno quer dizer, pois usam sua própria semântica.

**Análise de predicado:** aqui é feito o agrupamento dos textos com relação ao seu significado. O caminho mais simples para conduzir este tipo de avaliação é, verificar a frequência do uso de termos descritivos, aplicativos, avaliativos e analíticos.

**Software de avaliação adaptável:** outra maneira de se medir o aprendizado é realizar perguntas aos alunos durante o decorrer do curso. Algumas perguntas são feitas quando se acessa o sistema, outras são apresentadas quando o aluno alcança algum ponto importante do curso, etc. O estudante poderá, em muitos casos, pular a questão, sabendo que isto aumentará o peso da avaliação nas questões subseqüentes. Isto permite ao aluno melhor adaptar sua atividade educacional a outras atividades do dia. Entretanto algumas questões tem que ser respondidas na hora e sem consulta, enquanto que em outras, pode-se consultar o material. Esta avaliação inclui:

- Realização dos objetivos de aprendizado do curso;
- Cumprimento do alvo educacional do estudante;
- Crescimento e disposição afetiva do estudante;
- Satisfação do estudante com relação a instrução, curso e ambiente de aprendizado;
- Avaliação da faculdade com relação a instrução, curso e ambiente de aprendizado.
- *Software* de suporte: a decisão e disseminação de informações;
- Sistema de relação de normas e nível: registro de como o aluno deverá se comportar e responder;
- Sistemas de banco de dados e *datawarehouse*;
- Relatórios da evolução e utilização.

Através destes componentes, o *CyberQ* procura oferecer uma avaliação mais completa do aluno, buscando mesmo para questões discursivas, dar o *feedback* adequado ao aluno.

### 3.8 Carnegie Mellon On-Line

*Carnegie Mellon On-line* é um sistema para cursos independentes, distribuição de educação baseada na *Web*, treinamento e material de avaliação. O sistema gera conteúdo customizado (avaliação e *feedback*) para cada estudante individualmente.

O modelo desenvolvido por Carnegie Mellon University [BRU 97] [REH 97], possui uma infra-estrutura centrada no estudante, que realiza a entrega do material do curso para os alunos e faz seu gerenciamento baseado na *Web*. O sistema gera conteúdos personalizados (avaliações, *feedbacks*) para cada aluno, e registra todo o caminho que ele fizer durante o curso, obrigando a cumprir todas as limitações do curso (pré-requisitos, restrições de limite de tempo e momento para exames).

O sistema suporta um grande número de alunos e de cursos e considera a avaliação muito importante, pois é a maneira de estimular o aprendizado, monitorar as classes a distância e de adaptar uma aula ao aluno.

O sistema é estruturado em seis elementos:

- Distribuição de conteúdo;

- Gerenciamento e administração do curso;
- Administração do estudante;
- Conteúdo de autoria;
- Operações;
- Conteúdo.

O Sistema possui um Banco de Dados, onde é armazenado todo o conteúdo do curso e o registro dos passos do aluno, o que facilita o processo de avaliação. Possui um sistema de segurança e autenticação na rede, mas o aluno necessita de apenas um navegador *Web* e uma conexão à Internet. A segurança é feita com transações SSL (*Secure Sockets Layer*), e a autenticação com *Kerberos* [DAN 96].

Cada curso possui um número de características específicas para representar e processar a avaliação. Para cada módulo, existem práticas e exames. Por exemplo, o sistema pode gerar um exame personalizado baseado em um número de diferentes critérios e parâmetros, incluindo os trabalhos já realizados pelo aluno, bem como seus resultados. A correção é feita automaticamente ou posteriormente pelo professor, caso seja necessário, e o seu resultado é enviado para o aluno.

Com relação ao progresso do aluno, cada vez que ele acessa o Banco de Dados para uma determinada unidade, registra-se o momento do acesso, bem como a duração deste.

A arquitetura do *Carnegie Mellon On-line* é apresentada na figura 3.1.

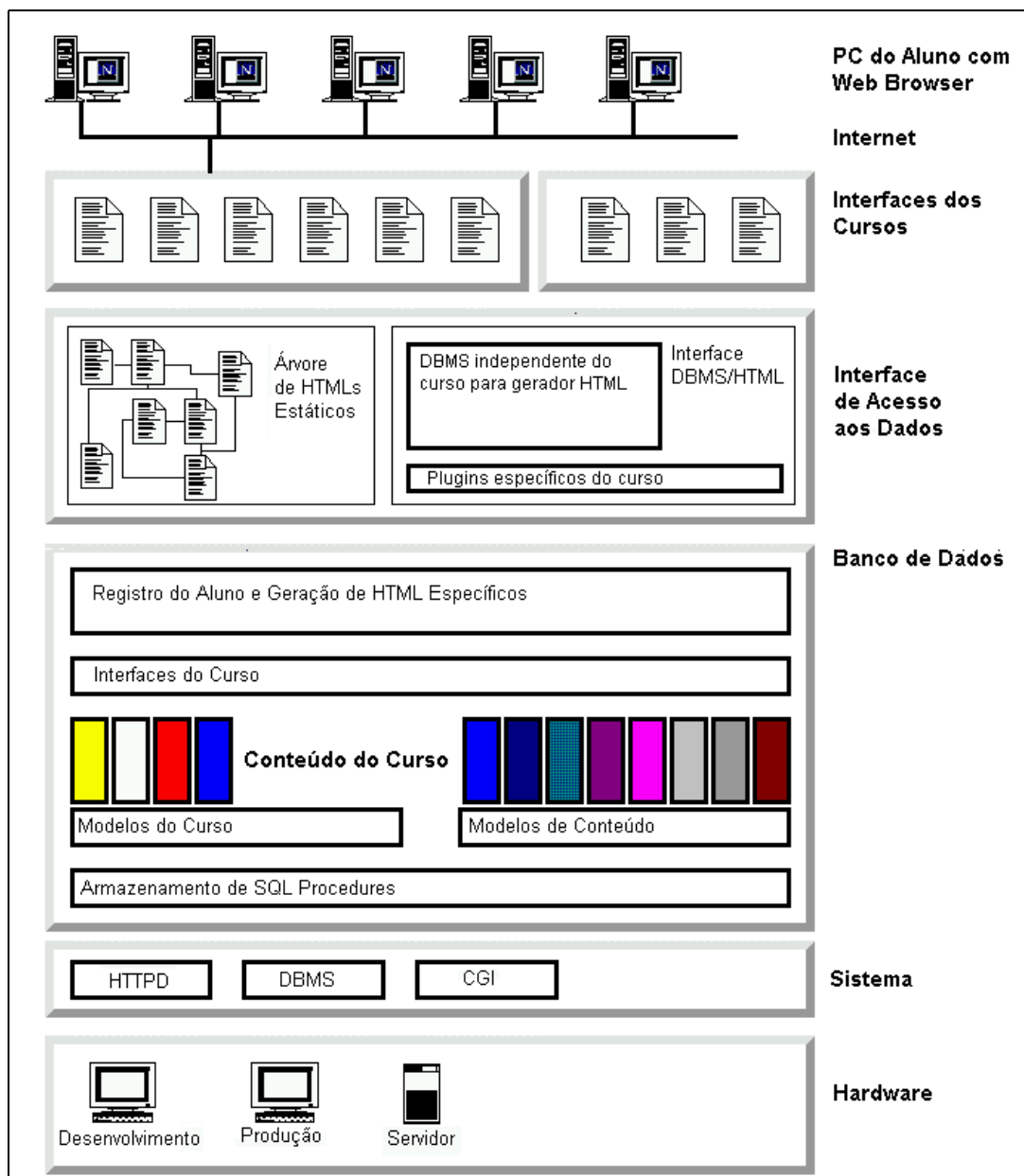


FIGURA 3.1 - Arquitetura do Carnegie Mellon On-Line [BRU 97]

### 3.9 CADAL Quiz

Em 1996, o Departamento de Ciência da Computação da Universidade de Monash, implementou um Projeto “*First Year Advanced Students*”, e um deles ficou conhecido como *CADAL Quiz*.

O projeto conhecido como *CADAL Quiz*, (*Computer Aided Dynamic Assessment & Learning Quiz*), envolve interface e implementação em WWW baseada num gerador de testes de múltipla escolha e ferramenta de avaliação [CAR 97].

O usuário necessita ter conhecimentos em linguagem HTML, pois o ambiente não apresenta uma interface para a criação das questões.



### 3.10 SMART

SMART foi desenvolvido por um grupo de cognição e tecnologia da Universidade de Vanderbilt. Durante 10 anos, os pesquisadores estudaram, com estudantes e professores, os caminhos para motivação e avaliação do aprendizado.

SMART (*Special Multimedia Arenas for Refining Thinking*) envolve uso interativo da Internet e *software* multimídia. A internet serve para três funções importantes: Age como uma ferramenta de avaliação formativa para oferecer *feedback* individualizado para os estudantes, cria uma comunidade de aprendizado mostrando dados submetidos pelos participantes das salas de aula e promove a discussão e reflexão de conceitos importantes [SMA 97].

No SMART, a Internet funciona como uma ferramenta para o professor e uma ferramenta de avaliação.

Avaliação Formativa pelos estudantes é sinônimo de atividades de auto-avaliação, onde estudantes refletem suas concepções. Avaliação Formativa pelos professores envolve monitoramento dos conhecimentos dos estudantes e habilidades para propósito de tomada de decisão instrucional.

No SMART, estudantes interagem por ciclos de solução de problemas e revisão. Estudantes acessam sua página na internet, SMART WWWeb, durante a fase de revisão. Essencialmente, SMART WWWeb possui três funções: Primeiro, provê *feedback* individualizado para estudantes. Neste caminho, o servidor *Web* serve como uma ferramenta de avaliação formativa. O *feedback* sugere aspectos do trabalho do estudante que são necessários na revisão e recursos de sala de aula que estudantes podem usar a ajuda para a revisão. O *feedback* não fala para o estudante a resposta certa. O *feedback* da *Web* é gerado dos dados que os estudantes entram individualmente. Essencialmente dados que são submetidos pelos estudantes no *browser* é coletado da base de dados do servidor.

A segunda função do SMART WWWeb é coletar, organizar e mostrar os dados coletados das salas de aula. Dados mostrados são automaticamente modificados como novos dados que são submetidos para a base de dados pelos estudantes. Esta sessão é chamada de *SMART Lab*. Os dados no *SMART Lab* consistem de respostas de estudantes para problemas e explicações para suas respostas. Os dados de cada classe são mostrados separadamente de dados de salas de aula distribuídas. Esta característica permite o professor e sua classe discutir diferentes estratégias de solução. Esta discussão provê uma rica fonte de informações para o professor de como os estudantes estão pensando sobre o problema e são planejados para estimular estudantes a refletir.

A terceira sessão do SMART WWWeb é *Kids Online*. *Kids Online* consiste da explanação pelos estudantes atores. As explicações são baseadas em textos com narração de áudio.

Estudantes são questionados para escolher e justificar sua escolha na ferramenta.

### 3.11 QuizSite

Avaliação e testagem são processos essenciais em ensino e aprendizagem. O *Bureau of Evaluative Studies and Testing* (BEST) ajuda o corpo docente da Universidade Indiana, estudantes e administradores com estas atividades [UNI 96]:

- Consulta com departamentos e corpo docente individual sobre a avaliação do aprendizado do estudante *concerning assessment of student learning*, avaliação do ensino e avaliação dos programas.
- Desenvolvimento e manutenção de ferramentas baseadas na WWW para teste, avaliação.
- Administração de testes desenvolvidos pelo professor e testes de padrões nacionais.
- Máquina de contagem de testes, levantamento e instrumentos de avaliação.

*QuizSite* é um sistema baseado na *Web* criado pela *Bureau of Evaluative Studies and Testing*, que permite instrutores distribuírem tarefas, perguntas, exames, artigos miúdos ou avaliações via WWW. *QuizSite* também provê retorno imediato para instrutores e estudantes.

### 3.12 QUIZIT

QUIZIT é um sistema de geração automática de questões baseado na WWW, desenvolvido pelo Departamento de Ciência da Computação de Virginia Tech, que é usado em vários cursos a distância na *Web* [TIN 96].

Quando o ambiente QUIZIT foi projetado, os seguintes itens foram considerados:

**Uso da WWW para distribuição de testes:** Esta necessidade pode ser explanada pelo caráter dinâmico do ambiente universitário, onde mudanças adaptativas ocorrem necessariamente durante o curso, requerendo um estável ambiente de rede e um sistema de *feedback* rápido. Outra importante característica é portabilidade.

**Independência de criação e interface:** O processo de criação de questões deve ser completamente independente do processo de tempo de execução.

**Segurança:** Provê transações seguras entre clientes WWW e servidores.

### 3.13 Hyperlearning Meter System

O *Hyperlearning Meter System*, é um sistema específico para avaliação e certificação, utilizado na *George Mason University* [MEN 98]. O sistema apresenta questões individualizadas para cada aluno, nunca repetindo a mesma questão; gera as questões a partir de modelos preparados pelos instrutores; e fornece estatísticas sobre o progresso do aluno baseado num mapa conceitual do domínio de conhecimento que está sendo avaliado. Também permite ao instrutor definir pontos de realimentação aos alunos, com informações das deficiências de cada um e os módulos a serem estudados

que poderão ajudar.

O *Hyperlearning Meter* pode ser usado por qualquer organização que mede aprendizado: de cursos colegiais a companhias de treinamento. O centro de desenvolvimento do *Hyperlearning Meter*, oferece a instalação do *Hyperlearning Meter* em sua organização e criação de questões, base de dados de questões e testes [HYP 98].

### **3.14 Comparação dos Processos de Avaliação nos Ambientes de Ensino a Distância via *Web***

Conforme observou-se no estudo dos ambientes de ensino a distância, o número de mecanismos de avaliações disponíveis, em alguns sistemas atualmente, é bastante abrangente. Fica clara a preocupação dos autores em fazer com que a avaliação do aluno não esteja restrita a provas e testes, mas também utilizando-se de outros subsídios.

É interessante perceber que a ferramenta de elaboração de testes se faz presente na maioria dos sistemas, o que comprova a informação de que esta é a forma mais utilizada para a avaliação dos alunos, apesar de nunca ser a única. Na implementação de sistemas de monitoramento, também encontrou-se tal forma de avaliação, no entanto, não verificou-se, em nenhum dos ambientes, um sistema capaz de comunicar ao professor os problemas ocorridos com algum aluno no momento de realização do curso. A maioria das técnicas utilizadas oferece relatórios finais com os dados de acesso e não um aviso para o responsável. O professor geralmente deve buscar estas informações para cada aluno e não recebe um aviso do que está acontecendo no processo de aprendizagem. Isto só vem a comprovar a utilidade do sistema de alertas em cursos a distância. A Tabela 3.1 apresenta uma comparação entre os modos de avaliação existentes nos ambientes de ensino-aprendizagem estudados.

TABELA 3.1 - Comparativo entre os modos de avaliação

<i>Software</i>	<i>Modos de Avaliação</i>
<b>WebCT</b> Universidade de British Columbia	<ul style="list-style-type: none"> <li>- possui ferramenta de avaliação denominada <i>WebCT Student Management</i></li> <li>- criação de categorias de provas;</li> <li>- cinco tipos de questões: múltipla escolha, relacionar, respostas curtas, calcular, discursiva;</li> <li>- correção automática;</li> <li>- teste periódicos</li> <li>- limite de duração da prova;</li> <li>- número de tentativas;</li> <li>- estatísticas;</li> <li>- visualização da prova pelo aluno depois de corrigida.</li> </ul>
<b>Virtual-U</b> Simon Fraser University	<ul style="list-style-type: none"> <li>- exercícios e tarefas;</li> <li>- atende processos cognitivos envolvendo o ensino e aprendizagem on-line;</li> </ul>
<b>TopClass</b> WBT Systems	<ul style="list-style-type: none"> <li>- banco de dados de questões;</li> <li>- geração randômica de testes;</li> <li>- permissão para realizar o teste mais vezes;</li> <li>- testes com correção automática;</li> <li>- feedback;</li> <li>- oito tipos de questões: Text, Pick One, Upload, MCObjs, MCA, List matching, ImageMap, Boolean;</li> <li>- relatório individual ou por turma;</li> <li>- estatísticas.</li> </ul>
<b>WebCourse in a Box</b> Virginia Commonwealth University	<ul style="list-style-type: none"> <li>- cria testes e exercícios de múltipla escolha ou resposta direta;</li> <li>- não armazena as respostas em banco de dados;</li> <li>- realiza a correção de todas as questões e pode fornecer os resultados a cada resposta feita pelo aluno ou somente após responder todas as questões;</li> <li>- o instrutor pode utilizar mensagens de resposta de acordo com o desempenho do aluno, bem como informações extra para questões respondidas erroneamente.</li> </ul>
<b>Learning Space</b> IBM- Lotus	<ul style="list-style-type: none"> <li>- possui ferramenta de avaliação denominada <i>Assessment Manager</i>;</li> <li>- revisão baseada em perguntas, distribuídas aleatoriamente;</li> <li>- avaliação com controle de tempo;</li> <li>- classificação de perguntas por categoria.</li> </ul>
<b>AulaNet</b> LES- PUC-Rio	<ul style="list-style-type: none"> <li>- possui ferramenta de avaliação, denominada <i>Quest</i>;</li> <li>- oferece <i>feedback</i> ao aluno;</li> <li>- questões de domínio cognitivo;</li> <li>- geração de relatórios;</li> <li>- três tipos de questões: V ou F, preencher lacunas, múltipla escolha;</li> <li>- possui correção automática;</li> <li>- auto-avaliação;</li> <li>- gera estatísticas para o professor.</li> </ul>

<p><b>CyberQ</b> InterEd do Grupo Apollo</p>	<ul style="list-style-type: none"> <li>- implementação de software de múltiplos métodos de estratégias de avaliação;</li> <li>- avaliação simultânea, virtualmente, de múltiplos estudantes de vários níveis;</li> <li>- recolhe dados de maneira transparente para os estudantes, corpo docente e funcionários;</li> <li>- agrega uma grande quantidade de dados praticamente para não limitar análises secundárias;</li> <li>- grande quantidade de dados a um custo muito baixo por unidade;</li> <li>- pega dados garantindo a avaliação longitudinal.</li> </ul>
<p><b>Carnegie Mellon On-Line</b> Carnegie Mellon University</p>	<ul style="list-style-type: none"> <li>- possui ferramenta de avaliação denominada <i>CSW OnLine</i>;</li> <li>- conteúdo Individualizado;</li> <li>- <i>feedback</i> para o aluno.</li> </ul>
<p><b>CADAL Quiz</b> Universidade de Monash</p>	<ul style="list-style-type: none"> <li>- gerador de testes de múltipla escolha e ferramenta de avaliação;</li> <li>- necessita de conhecimentos em HTML.</li> </ul>
<p><b>SMART</b> Universidade de Vanderbilt</p>	<ul style="list-style-type: none"> <li>- age como uma ferramenta de avaliação formativa;</li> <li>- oferece <i>feedback</i> individualizado para os estudantes.</li> </ul>
<p><b>QUIZIT</b> Virginia Tech</p>	<ul style="list-style-type: none"> <li>- uso da WWW para distribuição dos testes;</li> <li>- portabilidade;</li> <li>- independência de criação e interface;</li> <li>- envolve segurança.</li> </ul>
<p><b>QuizSite</b> University of Indiana</p>	<ul style="list-style-type: none"> <li>- permite aos instrutores distribuírem tarefas para casa como: perguntas, exames ou pequenos artigos;</li> <li>- avaliações via WWW;</li> <li>- provê retorno imediato para instrutores e estudantes.</li> </ul>
<p><b>Hyperlearning Meter System</b> George Mason University</p>	<ul style="list-style-type: none"> <li>- apresenta questões individualizadas, nunca repetindo a mesma questão;</li> <li>- gera as questões a partir de modelos preparados pelos instrutores;</li> <li>- fornece estatísticas sobre o progresso do aluno;</li> <li>- permite ao instrutor definir pontos de realimentação aos alunos.</li> </ul>

## 4 Técnicas Propostas para Avaliação de Alunos em Cursos na *Web*

Existem alguns ambientes, um pouco menos conhecidos, com funcionalidades específicas, que preocupam-se em avaliar as atividades dos alunos de uma maneira mais completa. Em geral, são projetos de Universidades que pesquisam sobre o tema educação a distância, não comercializados para outras instituições.

### 4.1 AnalogVB - Ferramenta de Análise de Atividades de Alunos de Sistemas de Ensino via Internet

Uma das técnicas sugeridas para acompanhamento das atividades dos alunos é a análise dos arquivos de *logs* dos servidores WWW [AZE 2000]. *Log* é um repositório de registros que contém informações que devem ser preservadas para futuras consultas.

Com a análise dos logs, pode-se verificar a quantidade de *hits*, *page views* e *unic user* que são feitos por dia ao servidor ou por *home-page*. Um *hit* é o número de solicitações feitas ao servidor, sejam páginas, arquivos gráficos ou de áudio. *Page views* é a quantidade de acessos à página, se a página é acessada mais de uma vez pelo mesmo usuário, a contagem é incrementada. *Unic user* é quantidade de acessos à página porém, se a página é acessada mais de uma vez pelo mesmo usuário, isto não interfere na contagem.

Em sistemas de ensino via Internet, como não há mais a interação em sala de aula entre aluno-professor e aluno-aluno, interação esta considerada de extrema importância no processo de ensino-aprendizagem, os professores estão desenvolvendo agentes inteligentes que possam monitorar os alunos para que seja escolhido um caminho próprio para cada perfil do estudante, suprindo desta forma, a lacuna criada com este tipo de ensino. Porém, os tutores virtuais poderão auxiliar os alunos mais adequadamente, quanto maior for a quantidade de informações sobre o comportamento de cada aluno. Como os registros dos *logs* de acessos ao servidor *Web* de ensino a distância são armazenados, esses registros podem fornecer o comportamento de cada aluno no que tocante aos acessos aos arquivos (páginas) armazenadas no servidor *Web*, e isso auxiliará os tutores virtuais avaliar o comportamento de cada aluno.

Como as informações armazenadas no arquivo de *log*, é possível descobrir a média de sessões de alunos por dia, as páginas mais acessadas no site. Estes dados podem auxiliar a determinar quais páginas são vistas com menor frequência ou não são acessadas, quais alunos participam mais efetivamente das tarefas, quais alunos não participam, etc.

O objetivo desta análise é de auxiliar o professor nos processos de avaliação de seus alunos, avisando de possíveis problemas que podem estar ocorrendo.

Existem basicamente dois tipos de *logs*: o de acessos e de erros. Quando o servidor *Web* está funcionando, cada solicitação de documento ou arquivo é registrada sob a forma de um item separado no arquivo de *log* do servidor. Os erros são registrados em outro arquivo. Os *logs* de erro são valiosos por mostrarem tentativas de

acesso a documentos controlados feitas por usuários não autorizados e por relatarem problemas do servidor.

Há uma grande quantidade de ferramentas que através da análise do arquivo de *log* dos servidores *Web*, fornecem informações estatísticas sobre os acessos às páginas. Porém devido às características dos hiperdocumentos disponibilizados na rede, estas informações não são suficientes para desvendar o comportamento do navegador.

O uso da ferramenta AnalogVB auxilia e apoia os agentes e os tutores virtuais inteligentes na análise do comportamento dos alunos dos cursos de ensino a distância, através da extração de dados dos registros de *log* do arquivo de *log* de acessos gerados por um servidor *Web*. A partir destes registros de *log*, a ferramenta AnalogVB, possui um módulo que gera arquivos com o nome de cada usuário/aluno que acessou o servidor *Web*. Cada arquivo gerado, contém o nome dos arquivos acessados pelo usuário/aluno e que estão armazenados no servidor *Web*, conforme registrado no arquivo de *log* de acessos do servidor *Web*.

#### **4.2 Access Miner – Ferramenta de Mineração de dados na descoberta do comportamento do usuário da *Web***

Uma das técnicas propostas para descobrir o comportamento dos alunos em cursos na *Web* é o uso da mineração de dados [BRU 2000].

A mineração de dados tem por objetivo extrair informações implícitas e potencialmente úteis de dados.

Normalmente este processo consiste em uma série de etapas, que inicia com a seleção do conjunto ou amostra dos dados a serem processados. Estes dados podem ser submetidos a uma etapa de pré-processamento onde são transformados para um formato adequado para o algoritmo de mineração, que procura por regras ou padrões ocultos nos dados. Finalmente, as informações descobertas são selecionadas e formatadas para a exibição, a fim de serem interpretadas e avaliadas, de forma que se selecione os conhecimentos úteis resultantes de todo o processo. Estas técnicas podem ser utilizadas para auxiliar a compreensão do comportamento do usuário da *Web*, através da descoberta de padrões ou regularidades que descrevem o seu perfil.

A solução proposta faz uso de um *script* (programa) inserido nas páginas que se pretende registrar o acesso ao site.

Este *script* gera um identificador exclusivo para a seção de navegação e solicita, a cada página carregada, a execução no servidor de outro *script*. Este programa é responsável por armazenar as informações em um arquivo que registra todos os acessos.

Estes dados são submetidos a uma etapa de processamento que tem por objetivo agrupar acessos consecutivos do mesmo usuário, gerando como resultado um conjunto de seções. Estas seções são submetidas ao processo de mineração para se descobrir regras de associação entre os alunos.

Muitas ferramentas de análises de sites, podem refletir apenas a própria organização do site, demonstrando não o comportamento ou preferências do usuário, mas

o fato de que esta pode ser a única alternativa para o mesmo chegar até a página desejada.

Observa-se que nenhuma destas técnicas resolve completamente o problema de descoberta do comportamento do usuário da *Web*, mas integradas a outras soluções, podem ajudar no resultado final.

### **4.3 Utilizando Agentes no Suporte à Avaliação Informal no Ambiente de Instrução baseada na *Web* - AulaNet**

Este trabalho apresenta um modelo baseado em agentes para o suporte a avaliação informal no AulaNet - um ambiente de instrução baseada na *Web*. Neste modelo, o processo de interação dos alunos com o ambiente de ensino desempenha um papel fundamental, fornecendo, aos professores, os mecanismos necessários a uma avaliação mais eficaz de seus alunos e design institucional.

Instrução baseada na *Web* (IBW) pode ser definida como o uso da WWW como um meio para a publicação de material de um curso, apresentação de tutoriais, aplicação de testes e comunicação com os estudantes. Ela também compreende o uso da *Web* para a apresentação de conferências multimídia de forma síncrona e assíncrona.

Os mecanismos de avaliação presentes em ambientes IBW estão restritos aos instrumentos de avaliação formal, tais como aqueles existentes em ambientes tradicionais de ensino, ou seja, trabalhos de ensino, ou seja, trabalhos individuais, provas, listas de exercícios, entre outros. No AulaNet esse tipo de avaliação é realizado pelo Quest, que tem por objetivo, determinar os resultados finais do processo de aprendizado, isto é, os processos e os produtos que descrevem a natureza e a extensão do aprendizado, seu grau de correspondência com os objetivos da instrução e sua relação com o ambiente de aprendizado [NOY 98].

Devido à inexistência do caráter presencial em ambientes IBW, os mecanismos de avaliação informal são extremamente complexos. Por essa razão, os professores perdem o *feedback* necessário a compreensão dos processos de aprendizagem dos alunos, que permitem não somente avaliá-los, como também apontar possíveis distorções no processo instrucional [FUK 98].

Em [FUK 98], acredita-se que através da observação do comportamento dos alunos durante o processo de interação com o ambiente pode-se captar informações sobre o processo de aquisição de conhecimento dos mesmos, que permitirá uma complementação aos processos de avaliação existentes no AulaNet. Através deste processo espera-se determinar como o aluno chega aos resultados apresentados; quais os caminhos percorridos com relação ao material didático; quais as fontes consultadas e com que frequência; se o aluno utiliza fontes suplementares fornecidas pelo professor; se o aluno tem iniciativa de pesquisar fontes suplementares por conta própria; qual a sua contribuição em atividades conjuntas; se o aluno entra em contato com professor e instrutores somente em datas próximas as datas de entrega de trabalhos, ou se mantém contato regular, qual a sua assiduidade em tarefas em grupo (*bate-papo*, videoconferências, etc.); se os trabalhos demonstram preferir determinados tipos de mídias a outros; e se o ritmo do conjunto dos alunos no consumo de material didático é homogêneo.



Para atingir esses objetivos este projeto propôs um modelo baseado em agentes assistentes de tarefa, para o suporte à avaliação informal no AulaNet.

O modelo proposto é uma agência composta de três agentes assistentes de tarefas, cada um com uma funcionalidade distinta. O Agente Assistente de Tarefa de Monitoração é responsável pelo monitoramento do processo de interação dos alunos com o Ambiente, onde é criado um *log* da interação desses alunos com o Ambiente, que reflete o processo de aprendizagem dos alunos em um curso. Esse agente possui componentes no cliente, gerando um histórico de navegação do aluno, e no servidor, para percorrer as listas de discussão, a fim de verificar a participação de alunos em atividades conjuntas. A atuação desse agente é totalmente transparente para os alunos, de forma a não comprometer o processo de ensino-aprendizagem. O professor é assistido por um Agente Assistente de Tarefa de Avaliação de Alunos, responsável pela consulta ao *log* de interação, ao modelo de alunos do AulaNet e a uma base de conhecimentos responsável pela interpretação desse *log* de interação. Esse agente também é capaz de confrontar as informações decorrentes dos processos de avaliação informal com as informações resultantes dos processos de avaliação formal do AulaNet, oferecendo um auxílio mais eficaz ao professor.

O professor também é assistido por um Agente Assistente de Tarefa de Avaliação de *Design*, responsável pela consulta ao *log* da interação, ao modelo instrucional do AulaNet e a uma base de conhecimentos responsável pela interpretação do *log* de interação. Esse agente é capaz de indicar possíveis distorções no *design* instrucional, refletidas em decorrência do comportamento verificado dos alunos e, conseqüentemente, oferecendo os mecanismos necessários à realização das correções verificadas, na busca do alcance da qualidade desejada. Os agentes assistentes de avaliação atuam como consultores - toda a decisão sobre modificações no *design* instrucional ou à avaliação dos alunos cabe ao professor. Esses agentes se baseiam em um modelo de classificação e sua implementação se dá através do uso de regras de produção, com a utilização de informações probabilísticas.

#### **4.4 Modelo de Avaliação segundo Maythe Vasconcelos**

Segundo VASCONCELOS em [VAS 99], um sistema de avaliação a distância para ser o ideal precisa possuir alguns requisitos como:

- ser acessível via *Web* ou recurso similar;
- possui interface simples, amigável e de fácil utilização tanto aos professores como aos alunos, pois o sistema deve destinar-se às pessoas de todas as áreas, não detendo-se somente a conhecedores de informática;
- manter uma base de dados que armazenem de todas as informações necessárias relativas a avaliação ;
- possuir um administrador responsável pela criação de novos cursos, gerenciamento de professores e exclusão de cursos após serem finalizados;
- fornecer ao professor a possibilidade de gerenciar os alunos, ou seja , deve permitir que o professor faça o cadastro dos alunos, bem como deve permitir que o professor mantenha o controle de frequência e informações referentes ao histórico do aluno no curso, tais como de acesso a determinadas páginas,

participações em listas de discussões e outros eventos, quando houverem;

- permitir a realização de conversas *on-line*, através de um fórum ou bate-papo, durante a prova para que os alunos tenham a possibilidade de esclarecer dúvidas quando surgirem.

#### 4.5 Modelo de Avaliação segundo Luciano Hack

O modelo de avaliação proposto por HACK em [HAC99], apresenta os seguintes mecanismos, como meio de obtenção de informações para a avaliação do aluno na Educação a Distância:

**Rastreamento:** registro de cada passo que é dado pelo aluno (página que acessou, data e hora), identificando que tipo de acesso o aluno tem feito (unidades visitadas, ferramentas utilizadas, etc.), bem como o tempo gasto com o acesso desses;

**Controle do fluxo de informações:** identificação da participação dos alunos nas ferramentas de correio eletrônico, salas de *bate-papo* e lista de discussão, registrando informações referentes a sua utilização, com relação a quantidade e qualidade da informação trocada, na quantidade, será feito o registro do número de vezes que o aluno participou da ferramenta, e na qualidade, também será armazenado algumas contribuições dadas, para posterior avaliação;

**Ferramenta Ponto de Vista:** organiza um ambiente, onde o professor pode apresentar uma questão, e através das respostas e posterior avaliação dos próprios colegas, obter de forma organizada, as melhores definições. Esta é uma maneira de classificar os alunos, não apenas por proporcionarem as respostas consideradas superiores, como pela participação neste tipo de mecanismo;

**Ferramenta Votação:** procura fornecer uma ferramenta, onde o professor pode obter um retorno rápido dos alunos sobre determinado assunto. E assim, através das respostas dos alunos, o professor poderá rever a forma como está conduzindo ou apresentando o curso. Também será através desta ferramenta que o aluno fará sua auto-avaliação.

Estas funcionalidades, devem ser implementadas de tal forma, que o professor não necessite de grandes conhecimentos, a não ser a utilização do *browser*, bem como possa obter os resultados esperados sem grandes manobras e gasto de tempo. Mas que tudo esteja ao seu alcance de forma simples e direto via Internet. A figura 4.1 apresenta a estrutura de avaliação proposta por Hack [HAC99].

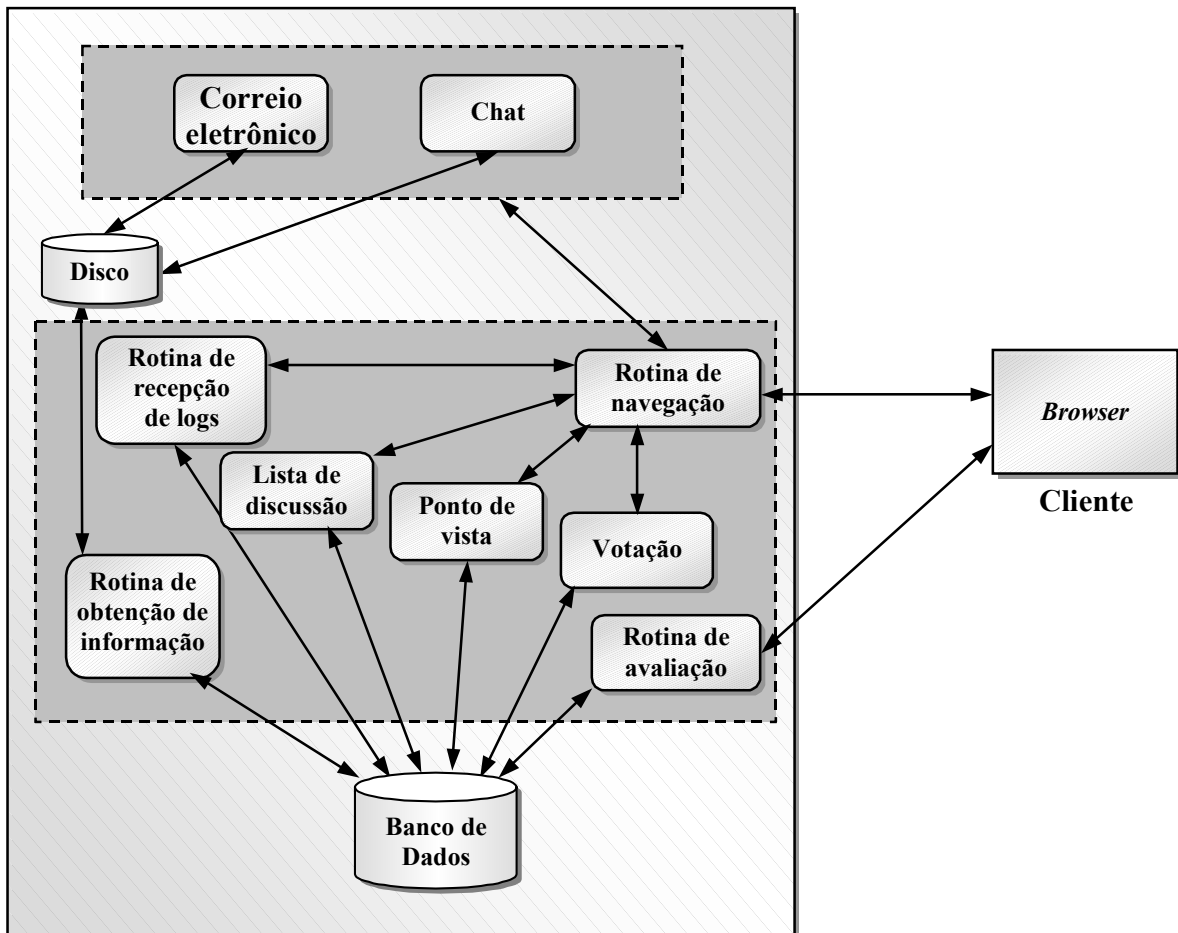


FIGURA 4.1 - Estrutura para avaliação do aluno [HAC 99]

Pode-se observar que os dados, que fornecerão registros para a avaliação do fluxo de informações, virão através de ferramentas de *bate-papo*, *mail* e lista de discussão, que terão suas informações registradas no disco, para posterior busca e seleção de informações úteis para a avaliação do aluno, registrando-as no banco de dados.

As informações da parte de rastreamento, serão fornecidas através da comunicação da rotina de navegação, que gerencia toda a parte de acesso ao curso na *Web*, com a rotina de registro de *logs*, que ficará responsável por armazenar as informações úteis para a avaliação no banco de dados.

E por último a rotina de consenso, que cuidará de implementar a ferramenta de consenso, proporcionando que os alunos primeiramente dêem a sua resposta a questão levantada, e assim que possível façam uma avaliação das respostas dadas.

#### 4.6 Proposta para um Ambiente de Avaliação Pedagógica - AmAPed

O Ambiente de Avaliação Pedagógica - AmAPed - atua como gerador de questões e provas para o professor, observador do desempenho do aluno, consultor para determinação de estratégias de ensino e terapias adequadas ao aluno em interação. Estas diferentes funções necessitam de ações em paralelo, como a observação do comportamento e desempenho do aluno frente às questões e exercícios para o aluno e

proposituras de terapia em caso de necessidades, isto tudo dentro de um ambiente que é dinâmico, porquanto o aluno está em interação e o sistema está em atuação com o aluno, aprendendo e se adequando [BER97].

O ambiente propõe uma arquitetura multiagente, baseada em agentes cognitivos e reativos e é constituída de observadores de desempenho e comportamento, de avaliação de desempenho, de analistas pedagógicos e de um mediador de construção. O mediador de construção tem a tarefa de, através de instruções específicas do professor, determinar os principais itens a serem observados e analisados quanto aos objetivos instrucionais, suprir a base de conhecimento associadas de modo a prover as formas e maneiras dos comportamentos reativos dos observadores de comportamento e desempenho.

Além disto, o mediador de construção gerencia uma base de questões, exercícios e problemas de acordo com os conteúdos e objetivos dos professores. Este banco de questões é constituído somente de originais e podem ser geradas variantes de questões dadas as especificações de cada produtor de questão ou professor. Uma linguagem ou ambiente para possibilitar a interação com o professor é necessária, quanto à descrição de questões a serem aplicadas e fórmulas para geração das questões sinônimas.

O meio de interação deve prover a possibilidade de especificação de itens importantes a serem analisados, se necessário, para cada processo tutor-aluno. A primeira versão adota uma interface orientada por menus e caixas de parâmetros e diálogos, bem dirigida, devido aos problemas naturais de comunicação livre entre um sistema de computação e um agente humano.

Os observadores de desempenho e de comportamento agem como uma sociedade de sistemas baseados em comportamentos, e atuam durante a interação com o aluno, como se fossem sensores. Seus dados e informações das observações ficam à disposição em uma área de comunicação, área de estudos e comunicação para uso de todos os agentes que necessitarem, especificamente os agentes da Sociedade de Analistas. O uso de uma área comum de estudos e informações para todos os agentes, foi adotada de forma a diminuir o fluxo de mensagens entre as diferentes sociedades de agentes.

Os Analistas são agentes cognitivos e cabe a eles a análise dos comportamentos do aluno para orientar o tutor quanto às interações instrucionais a serem adotadas.

#### **4.7 Ambiente Hipermídia Integrado para o Desenvolvimento de Cursos a Distância**

O ambiente hipermídia proposto pelo Instituto de Física de São Carlos para criação de cursos a distância constitui-se de vários módulos, que são descritos a seguir. Todo o sistema de gerenciamento dos módulos é feito através de um sistema central baseado em banco de dados, que é responsável pela coordenação dos diferentes módulos funcionais constituintes do ambiente [OLI 2000].

Módulo gerador de provas online (MGPO) - tem como função facilitar e automatizar a criação de provas para serem disponibilizados na WWW, provendo também a correção automática das questões. Todo o processo de criação das questões

das provas é feito através de páginas HTML e de uma linguagem de descrição de provas proprietária. O módulo MGPO é baseada em provas e questões. Uma prova consiste num conjunto limitado de questões, de um mesmo tipo ou de tipos diferentes, podendo cada questão possuir ou não um valor determinado. Além disso, cada prova pode, no momento de sua criação, possuir um número definido ou não de questões. Essas escolhas são feitas pelo instrutor ao criar cada prova. Cada questão de uma prova pode ou não ter sua correção exibida ao aluno. Alguns programas diferenciam entre teste, prova, *quiz*, etc. O MGPO não faz tal diferenciação. O tratamento didático depende apenas do instrutor, uma vez que o programa provê grande flexibilidade na criação das provas. A definição de uma metáfora clara propicia clareza, flexibilidade e modularidade à ferramenta.

O MGPO permite a criação dos seguintes tipos de questões:

- **Múltipla escolha:** podem possuir um número ilimitado de alternativas (o número mínimo consiste em duas alternativas). Apenas uma alternativa é a correta.
- **Preencher lacunas:** podem conter um número ilimitado de lacunas por questão, além de um número ilimitado de alternativas corretas por cada lacuna.
- **Resposta livre:** consistem de questões dissertativas.
- **Verdadeiro ou falso:** fornecem apenas duas alternativas de resposta: verdadeiro ou falso, sendo apenas uma a correta.

As questões das provas podem ter os seguintes valores:

- **Individual:** cada questão da prova terá um valor específico, escolhido pelo instrutor durante a criação de cada questão.
- **Único:** cada questão da prova terá um mesmo valor, escolhido previamente pelo instrutor.
- **Nenhum Valor:** as questões da prova não terão nenhum valor.

As provas podem conter um número fixo (definido previamente pelo instrutor) ou um número variável de questões. O programa MGPO possibilita ao instrutor optar pela correção automática ou parcial das questões das provas. Pode-se escolher quais questões de uma prova terão correção automática ou parcial. Por *default* as questões de múltipla escolha, preencher lacunas e verdadeiro ou falso possuem correção automática. As questões de resposta livre não são corrigidas automaticamente. O MGPO foi aprimorado, com a inclusão de alguns recursos não presentes em sua apresentação original. A principal mudança foi a inclusão de um banco de dados na ferramenta. Todos os professores devem ser cadastrados no banco de dados para utilizarem a ferramenta, que armazena assim todas as informações do mesmo e todas as atividades realizadas. Um recurso que está sendo desenvolvido no momento é a criação de um banco de dados para os alunos, armazenando todos os dados sobre os mesmos quando da realização de provas online.

Módulo Gerenciador de Dúvidas Online (MGDO) - permite o aluno enviar uma pergunta para o sistema. Este módulo contém “monitores” que são cadastrados por

disciplina e que serão os responsáveis pelas respostas.

O aluno necessita apenas selecionar a disciplina da qual faz parte sua pergunta, escrever sua pergunta e incluir uma ou mais palavras-chave. Ao enviar a pergunta, o sistema gerenciador irá verificar no banco de dados se existe alguma pergunta com a(s) palavra(s)-chave indicada(s). Caso encontre alguma, irá exibir ao aluno todas as perguntas que contenham as palavras-chave, além de sua resposta (caso já tenham sido respondidas), com os dados do aluno que a fez e do monitor que a respondeu. Independentemente de existir ou não uma pergunta com a palavra-chave, a pergunta do aluno será enviada a um monitor selecionado pela compatibilidade da disciplina.

Quando existe mais de um monitor em cada disciplina, o sistema seleciona aquele que está há mais tempo ocioso, distribuindo as perguntas de maneira equânime entre todos os monitores de uma mesma disciplina, de modo a não sobrecarregá-los. A pergunta e todos os dados relacionados a ela (como data de envio, horário de envio, dados do aluno) são gravados numa base de dados específica.

Quando do envio de uma pergunta por um aluno, o monitor selecionado para respondê-la irá receber um *e-mail*, contendo a pergunta do aluno e um campo para resposta. Na página de resposta, o monitor terá todos os dados sobre o aluno que enviou a pergunta, além da data e horário em que a mesma foi submetida. Quando o monitor responde a pergunta, o sistema gerenciador atualiza o banco de dados com a resposta e data e horário de resposta. Gera também o tempo total entre o envio da pergunta pelo aluno e sua resposta pelo monitor. O monitor então poderá ter uma estatística sobre seu desempenho. Tal estatística poderá ser consultada também pelo instrutor daquela disciplina. O aluno receberá um *e-mail* contendo a data de envio de sua pergunta, a data de resposta, a pergunta, a resposta dada pelo monitor e o nome e o *e-mail* do monitor responsável pela resposta. Uma vez que todas as ações efetuadas por alunos, monitores e instrutores estão gravadas no banco de dados, é fácil a criação das mais diferentes estatísticas. Pretende-se implementar numa versão futura várias estatísticas no sistema gerenciador.

Módulo Palm para Apoio ao Estudo (MPAE) – é uma ferramenta para o auxílio de aprendizagem que pode ser utilizada pelo professor como um recurso complementar às aulas. Dado um certo assunto em aula, o professor pode querer dar uma lista de exercícios para que os alunos fixem melhor o assunto. Então ele prepara um lista de questões, a qual chamamos simplesmente de teste, e transfere o teste para o Palm (organizador pessoal de mão) do aluno. Com isso, o aluno pode estudar a matéria através dos testes, em qualquer lugar e hora que tiver disponibilidade. O professor, por sua vez, pode ter um acesso a dados sobre a dedicação e as dificuldades do aluno. O aluno tem acesso aos testes propostos pelo professor e pode personalizar o seu programa de aprendizado fazendo os testes quantas vezes achar necessário, no momento em que achar conveniente e pelo tempo que tiver disponível. Os testes são baseados no conceito de cartão. Um cartão possui dois lados: o da pergunta e o da resposta e um teste possui vários cartões. O professor, através do Palm de cada aluno, tem acesso ao histórico do aluno com todas as informações relevantes, como uma lista das vezes que o aluno fez determinado teste, quanto tempo ele gastou em cada vez, porcentagem de acerto, etc. Além disso o professor pode acessar dados mais gerais como: número de vezes que o aluno fez o teste, média de aproveitamento, tempo médio gasto em cada questão, tempo total dedicado ao teste e percentual de questões que o aluno não soube responder. O professor também tem controle dos testes que ele quer deixar disponíveis ao aluno.

Sendo assim, ele pode gravar todos os testes que ele já tiver preparado e ir disponibilizando ao aluno à medida que a matéria dada em aula for avançando.

## 5 Descrição do Sistema de Alertas

Este trabalho propõe um modelo de um sistema de alertas inteligentes para ser utilizado em aplicações de ensino a distância via Internet, especificamente no projeto Tapejara da Universidade Federal do Rio Grande do Sul. A proposta do projeto Tapejara – Sistemas Inteligentes de Ensino na *Web* – é projetar e implementar um sistema de construção e acompanhamento de cursos a serem disponibilizados na rede Internet. Esta metodologia destaca o uso de técnicas pedagógicas que levem em conta aspectos afetivos e motivacionais, bem como a arquitetura multiagente para construção de sistemas tutores inteligentes.

O sistema de alertas foi desenvolvido junto aos grupos que compõem o Projeto Tapejara: UFRGS – II, CRT e UNISINOS. A CRT é responsável pelo conteúdo teórico que servirá de base ao tutor e ao sistema de alertas. Todas as formas de apresentação do material são definidas pelo grupo de pedagogos e pesquisadores da empresa. A UNISINOS é responsável pelos: modelo do aprendiz, definição das trajetórias e estilos cognitivos dos alunos, geração dos índices de aprendizagem, assim como pela definição dos perfis de alunos existentes na CRT. A UFRGS – II, que, neste projeto, é composta pelos subgrupos de *workflow*, interfaces e inteligência artificial, é encarregada de determinar a modelagem do *workflow*, a implementação do tutor nos seus diversos módulos, como a interface fixa do tutor e a comunidade de agentes.

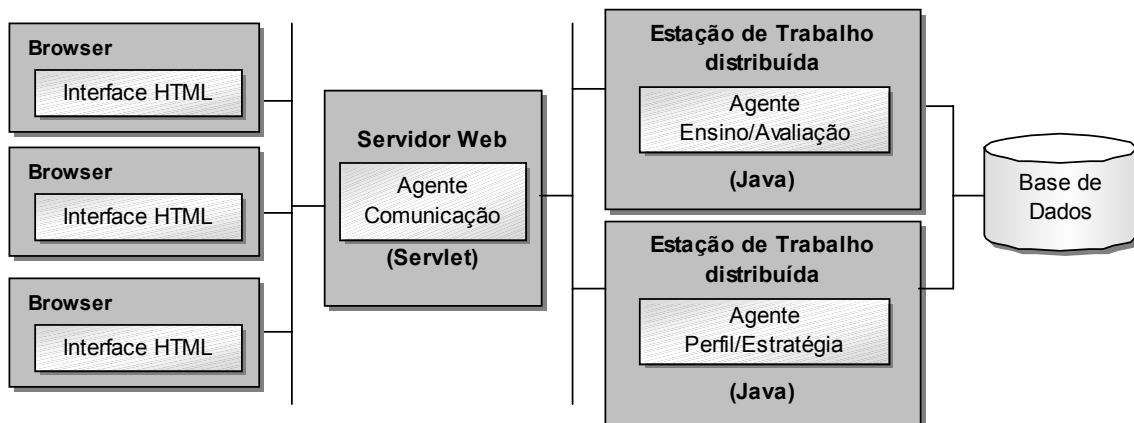


FIGURA 5.1- Arquitetura Multiagente Projeto Tapejara

A figura 5.1 mostra a arquitetura proposta no Projeto Tapejara que permite implementar um ambiente distribuído de ensino-aprendizagem inteligente baseado em técnicas de sistemas multiagentes. O ambiente contempla uma sociedade de agentes comprometidos com o aprendizado cooperativo, envolvendo os seguintes tipos de agentes:

- **Agente de Comunicação:** responsável por enviar as requisições dos usuários aos demais agentes, como, também, enviar as tarefas dos agentes aos usuários. Apenas repassará as mensagens.
- **Agente Ensino/Avaliação:** encarregado pela recuperação do conhecimento do domínio sobre cada ponto apresentando ao aluno, pela proposição de exercícios, avaliação de suas respostas, elaboração e controle para



apresentação de exemplos.

- **Agente Perfil/Estratégia:** integra os estilos cognitivos e apresenta o conteúdo de forma adaptativa. Após a verificação do estilo cognitivo, o agente identifica a melhor forma de ensinar e apresenta o conteúdo seguindo uma estratégia de ensino. A estratégia de ensino é modificada, caso o agente perceba um baixo desempenho do aprendiz.

Neste capítulo serão apresentados a arquitetura e o diagrama de classes do Sistema de Alertas. A integração deste com o sistema Tapejara também será mostrada. Ainda serão explicados aqui os principais componentes que servem de fonte de informação para envio dos alarmes (índices de aprendizagem, mecanismo de rastreamento, base de dados e base de conhecimento).

## 5.1 O Sistema de Alertas

Na maioria dos ambientes estudados e analisados, ou dos cursos a que se teve acesso, um dos problemas detectados é a falta de mecanismos que possibilitem o professor ter um acompanhamento mais completo e abrangente das atividades dos alunos, diagnosticando o seu nível de conhecimento bem como seu ritmo de aprendizagem. Isto possibilitaria uma avaliação mais precisa do estudante.

Normalmente, este processo de avaliação dos alunos na *Web* é realizado apenas através de informações referentes a testes, exercícios ou relatórios extensos que apresentam um histórico dos acessos feitos às páginas do curso [NED 99].

Através da constatação da necessidade de implantar uma metodologia de avaliação mais próxima do aluno, ou seja, que o veja mais de perto no seu dia-a-dia e que consiga captar o seu real estado de aprendizagem, o que se busca através deste trabalho, é, exatamente, proporcionar o acompanhamento da evolução do aluno mediante a sua interação no curso. Com isso, o aluno pode ser avaliado de diversas formas, desde as tradicionais provas dissertativas ou testes até avaliações que registrem os procedimentos e estratégias aparentemente utilizadas na solução de problemas.

A proposta deste trabalho é construir um sistema de alertas para apoio ao ensino, que amenize esta limitação da educação a distância, ajudando o professor a ter um acompanhamento mais completo das atividades dos alunos. Além disto, o próprio aluno pode receber alguns estímulos, ajuda, avisos, etc.

A importância do sistema de alertas para o ambiente proposto no projeto Tapejara está no estudo de uma proposta de monitoração das atividades dos alunos em cursos via Internet, principalmente, na descoberta de problemas. Este sistema é capaz de auxiliar os agentes Ensino/Avaliação e Perfil/Estratégia apresentados anteriormente, assim como os professores ou coordenadores do curso na descoberta de comportamentos fora do previsto. Com ele, o processo de avaliação do aluno será realizado a todo momento, evitando que o aluno não aproveite o curso satisfatoriamente, fato que provocaria numerosos abandonos ou fracassos. Desta forma, este sistema vem suprir a lacuna criada em sistemas de ensino a distância: a falta de um bom acompanhamento do processo de ensino-aprendizagem, avaliação do comportamento dos alunos e auxílio em suas sessões de estudo.

A principal característica de um sistema de alertas é a busca de situações críticas com base nas informações que possui a sua disposição [REC 97]. As informações utilizadas pelo sistema de alertas foram determinadas juntamente com o Grupo de Pesquisa em Processos Cognitivos da Unisinos, o qual definiu um conjunto de índices, chamados índices de aprendizagem, que indicam o processo de ensino-aprendizagem do estudante. Alguns destes índices são armazenados durante a interação do aluno no curso em um banco de dados, que contém as informações referentes aos acessos e às atividades dos alunos.

O Sistema de alertas também serve-se de uma base de conhecimento que representa as situações de exceção que devem ser detectadas. A lista de índices de aprendizagem, o banco de dados e a base de conhecimento são apresentados nas próximas seções.

Na posse destas informações, é possível descobrir comportamentos que podem exigir a emissão de um alerta. Alguns exemplos de situações que podem ser percebidas por este sistema são:

- aluno apresenta baixo desempenho nos exercícios;
- aluno não está aprendendo por uma determinada tática de ensino;
- aluno não está comparecendo às atividades do curso;
- aluno realizou uma parada brusca em alguma página.

Após reconhecido o problema no aprendizado, o sistema de alertas emite um aviso para a pessoa indicada, que pode ser o próprio aluno, professor, coordenador do curso ou agente inteligente, possibilitando assim, tomar uma decisão e solucionar o problema de aprendizagem. Alguns exemplos de mensagens que podem ser enviadas são:

- enviar material adicional ao aluno, como: ajudas simples, melhores explicações, mais exemplos e situações, exercícios, etc;
- enviar *e-mail* para o aluno desistente, incentivando-o a voltar ao curso;
- enviar *e-mail* para o professor, avisando qual aluno está participando mais ativamente das atividades de estudo.

### **5.1.1 O Modelo Especificado**

O Serviço de alertas é composto pelos seguintes módulos: Editor de Alertas, Base de Conhecimento, Monitor de Eventos, Servidor de Alertas e Banco de Dados. A figura 5.2 apresenta a arquitetura do sistema de alertas.

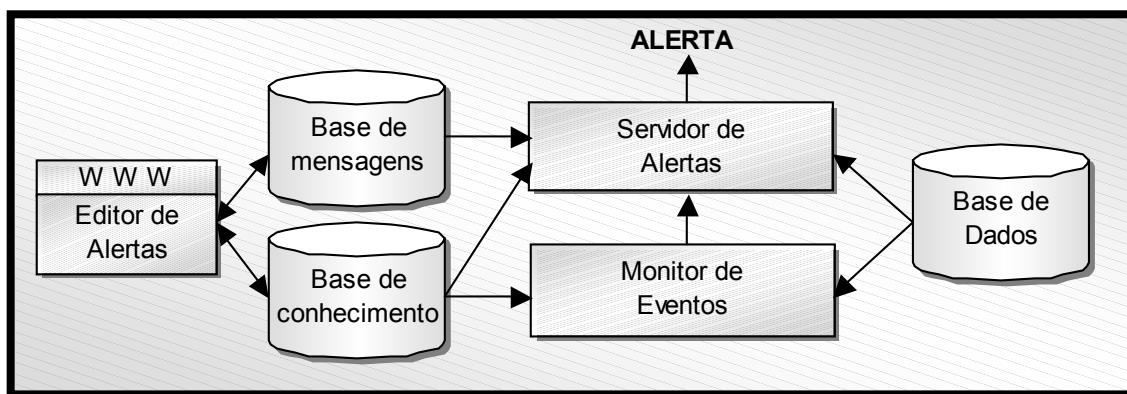


FIGURA 5.2 - Arquitetura do Sistema de Alertas

- **Editor de alertas:** módulo de construção e gerenciamento dos alertas. Através de uma página *Web*, permite ao usuário (professor, coordenador do curso, etc.) identificar os tipos de alertas desejados, especificar as condições que devem ser monitoradas e as ações que devem ser realizadas na ocorrência destas condições.
- **Base de conhecimento:** representa as situações de exceção que devem ser detectadas e geralmente são representadas através de regras. Uma regra é, simplesmente, um par condição-ação: dada a existência da condição expressa, faça a ação. As regras são expressas da seguinte forma: SE <condição> ENTÃO <ação>.
- **Monitor de eventos:** é o módulo que detecta a ocorrência de um evento de interesse e avisa o módulo Servidor de Alertas.
- **Servidor de alertas:** este módulo processa as regras associadas com os alertas, contidas na base de conhecimento, acessa a base de dados e, no momento que ocorre um evento crítico, emite o alerta correspondente, definido anteriormente através do editor de alertas.
- **Base de dados:** contém os dados sobre todas as atividades realizadas pelos alunos no curso. A estrutura que faz esta coleta de informações e armazenamento na base de dados foi implementada pelo grupo de inteligência artificial do Projeto Tapejara.
- **Base de mensagens:** contém todas as mensagens que podem ser enviadas pelo sistema de alertas. As mensagens são definidas no Editor de alertas.

### 5.1.2 Integração do Sistema de Alertas com o projeto Tapejara

O sistema de alertas desenvolvido nesta dissertação de mestrado deverá ser integrado com o sistema do Projeto Tapejara, conforme figura 5.3.

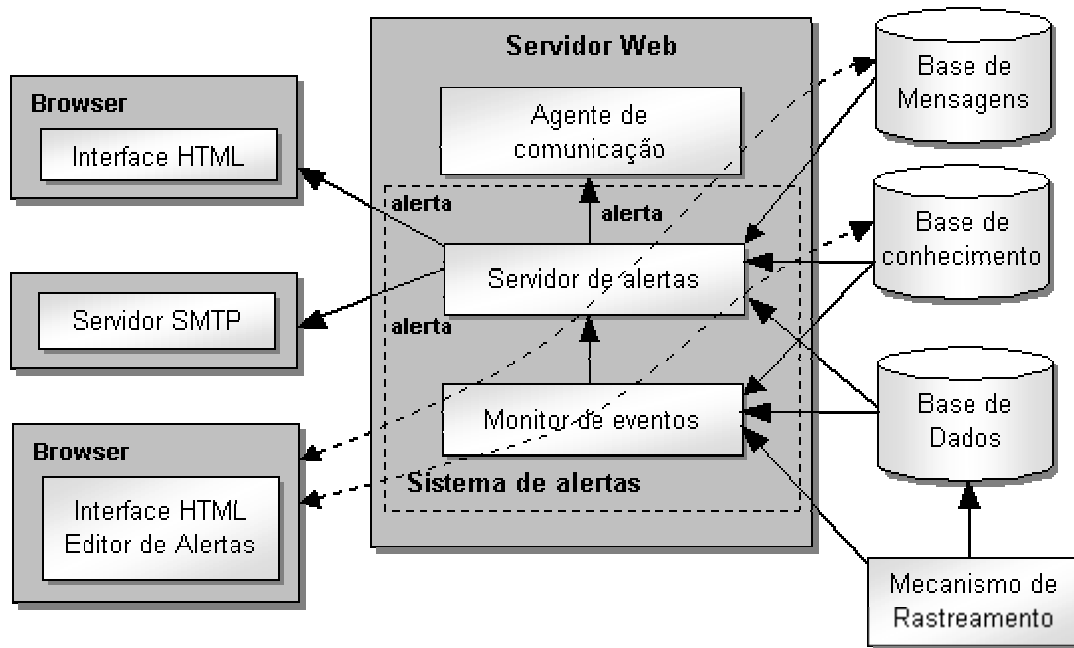


FIGURA 5.3– Integração do Sistema de Alertas com o projeto Tapejara

Após a definição, via Editor de alertas, dos alertas e mensagens a serem enviadas, o Sistema de alertas, através do módulo Monitor de Eventos, monitora alterações na base de dados para detectar ocorrências de eventos importantes assim como recebe aviso de ocorrências de eventos pelo mecanismo de rastreamento. O Monitor de eventos, através das informações obtidas da base de conhecimento, sabe quais os eventos que interessam. Quando um evento ocorre, o monitor de eventos informa o Servidor de Alertas. Este checa se existe alguma regra/condição associada para este evento na base de conhecimento. Se encontrada, ele verifica e realiza a ação indicada. Esta ação pode ser uma mensagem apresentada no *browser* do aluno, *e-mail* ou mensagem para o agente comunicação pertencente ao Tapejara, que se encarrega de repassar a mensagem para o agente correspondente (agente ensino/avaliação ou agente perfil/estratégia). A comunicação entre o Sistema de Alertas, o Agente Comunicação e o Mecanismo de Rastreamento é realizada via KQML, explicada na seção 5.4.

### 5.1.3 Dados de Entrada do Sistema

O Sistema de Alertas utiliza-se de um conjunto de dados de entrada que serve de fonte para reunir as informações necessárias para envio das mensagens de aviso. Estes dados são apresentados a seguir.

#### 5.1.3.1 Índices de Aprendizagem

O Sistema de Alertas serve-se de um conjunto de índices de aprendizagem para descobrir as situações que necessitam do envio das mensagens de alerta. Alguns índices são registrados no banco de dados pelo mecanismo de rastreamento durante a interação do aluno no curso, e outros são resultados de análises sobre os dados armazenados. Estes índices foram definidos juntamente com o Grupo de Pesquisa em Processos Cognitivos da Unisinos e são mostrados a seguir. Alguns índices necessitam de uma avaliação humana mais complexa. Logo, do conjunto de índices apresentado, alguns não

serão utilizados pelo sistema de alertas.

### *Índices do Aluno*

Contém os índices de identificação do aluno, identificação das páginas e módulos do curso e informação sobre as atividades (percurso, tempo e movimentos) realizadas pelo aluno. São considerados índices básicos e estão armazenados na base de dados. Alguns índices são armazenados na base de dados pelo mecanismo de rastreamento. Os índices marcados por ◆ são aqueles usados pelo sistema de alertas.

#### *Identificação*

- ◆ Nome / Senha do Aluno-Aprendiz
- ◆ Data/Hora de acesso do módulo
- ◆ Identificação da página = nome da página
- ◆ Tipo de página = Conceito/Exemplos/Exercícios
- ◆ Identificação do Curso e do Módulo = C1/M1 (C1- Telecomunicações / C2- SDH / C3- Negociação)
- ◆ Data e hora de saída do módulo

#### *Tempo*

- ◆ Tempo total de percurso no módulo
- ◆ Tempo médio de permanência nas páginas do módulo
- ◆ Tempo médio de permanência por tipo de página (Conceitos, Exemplos ou Exercícios).

#### *Percurso*

- ◆ Total de páginas acessadas no módulo.
- ◆ Quantidade de acessos por tipo de página.
- Total de retornos simples – volta à página anterior ou à seqüência anterior. Ex. A → B e B → A

#### *Movimentos*

- ◆ Paradas Bruscas– Trocas rápidas de páginas durante uma seqüência.
- ◆ Paradas Grandes– Pelo tempo de permanência na página.
- ◆ Aceleração e desaceleração - Significando facilidade ou dificuldade de assimilar o conteúdo, respectivamente. Pode ser verificado pelo tempo de permanência nas páginas.
- ◆ Mudanças bruscas de níveis (conteúdo, exemplos e exercícios). Ex. O aluno está lendo um conteúdo e bruscamente vai para um exercício e

começa a acertar todos os exercícios.

- Chunks - Blocos de resolução fechados numa seqüência de navegação com cortes. Ex. |Con/Exem/Exerc| Exem/Exer|
- Retroações – Quando o aluno se depara com uma dificuldade em uma página ‘X’ em função da interferência devido a um não entendimento das páginas anteriores que o impede de prosseguir e que acaba fazendo-o retornar às páginas anteriores. Ex.  $A \rightarrow B \rightarrow C \rightarrow D$  | parada |  $D \rightarrow C \rightarrow B$
- Loopings - Fazer idas e voltas numa seqüência determinada.  $A \rightarrow B \rightarrow C \rightarrow A \rightarrow B \rightarrow C$
- Repetição de *Scripts* – conjunto de procedimentos repetidos e ‘empacados’ num domínio específico de ações cognitivas.
- Esquemas Mentais – são verificados pelos conjuntos organizados de seqüências de navegação.

### *Avaliação*

- ◆ Quantidade de questões corretas
- ◆ Quantidade de questões erradas
- ◆ Total de erros por exercícios
- ◆ Total de acertos por exercícios

### *Estilos Cognitivos*

- Analógico-Analítico
- Concreto-Genérico
- Dedutivo-Avaliativo
- Relacional-Sintético

### *Índices do Sistema*

São os índices que servem de parâmetros para comparação.

- ◆ Tempo Padrão necessário em cada página (Tempo mínimo, médio máximo por página)
- ◆ Tempo Padrão necessário no módulo (Tempo mínimo, médio e máximo por módulo)
- ◆ Número de Respostas corretas por exercícios.
- ◆ Média de páginas percorridas por módulo
- ◆ Avaliação Final
- ◆ Desempenho mínimo nos exercícios

## 5.2 Mecanismo de Rastreamento

O Sistema de Alertas utiliza-se de um mecanismo de rastreamento implementado pelos pesquisadores do Projeto Tapejara. Este módulo é responsável por armazenar, na base de dados, todas as informações sobre a atividade dos usuários no curso e refere-se às sessões e à seqüência de páginas acessadas pelos alunos. Dentre as informações, destacam-se a data e hora de acesso, tempo despendido em cada página e sessão, seqüência de navegação do usuário, contendo informações sobre cada página acessada como título da página, URL, data e hora de acesso, entre outras. Também fazem parte destes dados alguns índices de aprendizagem já apresentados na seção 5.11. Estes dados podem ser utilizados na avaliação do aluno e para geração dos alertas, na medida em que os dados armazenados sofrerem mudanças acima dos limites estabelecidos durante a atividade do aluno.

A geração destas informações começa a ser produzida a partir do momento que o usuário inicia a sua sessão de estudo. A partir daí, qualquer operação efetuada é armazenada na base de dados. Esta base de dados é constituído de várias tabelas, as quais são descritas logo abaixo.

TABELA 5.1 – Tabela Curso

<b>Nome da Tabela</b>	CURSO
<b>Descrição</b>	possui dados cadastrais dos cursos.
<b>Atributos</b>	<b>Descrição do Campo</b>
codcurso	identificação do curso
codautor	identificação do autor do curso
descricao	Nome/Título do curso
codpessoa	identificação da pessoa
datacriacao	data de criacao do curso
escolaridademinima	escolaridade mínima para participar do curso
recurso	recursos necessários para execução do curso
objetivo	descrição dos objetivos do curso
notaexigida	nota mínima exigida para aprovação no curso

TABELA 5.2 – Tabela Bibliografia

<b>Nome da Tabela</b>	BIBLIOGRAFIA
<b>Descrição</b>	Bibliografia utilizada para realização de um curso.
<b>Atributos</b>	<b>Descrição do Campo</b>
codbiblio	identificação da bibliografia
codcurso	identificação do curso
descricao	Descrição da bibliografia – autor, titulo, etc

TABELA 5.3 – Tabela Modulo

<b>Nome da Tabela</b>	MODULO
<b>Descrição</b>	São os módulos que compõem um curso.
<b>Atributos</b>	<b>Descrição do Campo</b>
codmodulo	identificação do módulo
codcurso	identificador do curso
descricao	descrição do módulo
recursos	recursos necessários para execução do módulo

TABELA 5.4 – Tabela Tatica

<b>Nome da Tabela</b>	TATICA
<b>Descrição</b>	São as formas de apresentar o conteúdo do curso.
<b>Atributos</b>	<b>Descrição do Campo</b>
codtatica	identificação da tática
descricao	descrição da tática

TABELA 5.5 – Tabela URL

<b>Nome da Tabela</b>	URL
<b>Descrição</b>	URLs disponibilizadas como conteúdo de cursos
<b>Atributos</b>	<b>Descrição do Campo</b>
codurl	identificação da URL
velocleitura	velocidade de leitura prevista para determinada url
descricao	descrição da URL – caminho na rede

TABELA 5.6 – Tabela Pessoa

<b>Nome da Tabela</b>	PESSOA
<b>Descrição</b>	Cadastro de instrutores e alunos
<b>Atributos</b>	<b>Descrição do Campo</b>
codpessoa	identificação da pessoa
nmpessoa	nome completo
RG	RG
CIC	CIC
escolaridade	escolaridade
endereco	endereço
complemento	complemento do endereço
cidade	cidade
cep	cep
estado	estado
telefone	telefone
fax	fax
codtipo	identifica o tipo de pessoa (aluno ou instrutor)
senha	senha de identificação do aluno dos cursos
codestilo	identifica o estilo cognitivo do aluno
e-mail	endereço eletrônico do autor



TABELA 5.7 – Tabela AlunoCurso

<b>Nome da Tabela</b>	ALUNOCURSO
<b>Descrição</b>	informação sobre os alunos e seus cursos
<b>Atributos</b>	<b>Descrição do Campo</b>
codpessoa	identificação da pessoa
codcurso	identificação do curso

TABELA 5.8 – Tabela Autor

<b>Nome da Tabela</b>	AUTOR
<b>Descrição</b>	informação sobre os autores do curso
<b>Atributos</b>	<b>Descrição do Campo</b>
codautor	identificação do autor
nome	nome completo do autor
<i>email</i>	endereço eletrônico do autor

TABELA 5.9 – Tabela LOG

<b>Nome da Tabela</b>	LOG
<b>Descrição</b>	Cadastro do histórico de trajetórias do aluno
<b>Atributos</b>	<b>Descrição do Campo</b>
indice	contador de acessos às páginas
dataini	data inicial de acesso a URL
datafin	data final de acesso a URL
codpessoa	identificação da pessoa
codsessao	identificação da sessão de estudo
codurl	identificação da página acessada

TABELA 5.10 – Tabela Estilo

<b>Nome da Tabela</b>	ESTILO
<b>Descrição</b>	contém os estilos cognitivos dos alunos
<b>Atributos</b>	<b>Descrição do Campo</b>
codestilo	identificação do tipo de estilo
descricao	descrição do estilo (analógico-analítico, concreto-genérico, dedutivo-avaliativo, relacional-sintético)

TABELA 5.11 – Tabela Topico

<b>Nome da Tabela</b>	TOPICO
<b>Descrição</b>	contém os tópicos pertencentes ao curso
<b>Atributos</b>	<b>Descrição do Campo</b>
codcurso	identificação do curso
codmodulo	identificação do modulo do curso
codtopico	identificação do tópico
descrição	descrição do tópico

TABELA 5.12 – Tabela Avaliacao

<b>Nome da Tabela</b>	AVALIACAO
<b>Descrição</b>	contém as notas do aluno
<b>Atributos</b>	<b>Descrição do Campo</b>
codpessoa	identificação do aluno
nota	nota tirada no curso
codcurso	identificação do curso
codmodulo	identificação do módulo

TABELA 5.13 - Tabela Estratégia

<b>Nome da Tabela</b>	ESTRATÉGIA
<b>Descrição</b>	Contém as estratégias de ensino para cada estilo cognitivo
<b>Atributos</b>	<b>Descrição do Campo</b>
codestrategia	Identificação da estratégia
codcurso	Identificação do curso
codmodulo	Identificação do módulo
codtopico	Identificação do tópico
codtatica	Identificação da tática
codestilo	Identificação do estilo
codurl	Identificação da página
nropags	Total de páginas

TABELA 5.14 – Tabela Exercicios

<b>Nome da Tabela</b>	EXERCICIOS
<b>Descrição</b>	contém a descrição dos exercícios
<b>Atributos</b>	<b>Descrição do Campo</b>
codexe	Identificação do exercício
codurl	Identificação da página
minimoacertos	Mínimo de acertos estipulado
totalquestoes	Total de questões

Tabela 5.15 – Tabela Relatorios

<b>Nome da Tabela</b>	RELATORIOS
<b>Descrição</b>	contém os dados para os relatórios
<b>Atributos</b>	<b>Descrição do Campo</b>
codrel	identificação do relatório
codurl	Identificação da página
codpessoa	Identificação do aluno
codproblema	identificação do problema detectado. Ex. Parada.

Tabela 5.16 – Tabela Problemas

<b>Nome da Tabela</b>	PROBLEMAS
<b>Descrição</b>	contém os tipos de problemas para o relatório
<b>Atributos</b>	<b>Descrição do Campo</b>
codproblema	identificação do problema
descricao	descrição do problema

A figura 5.4 mostra o modelo Entidade-Relacionamento (ER) da base de dados utilizada pelo Mecanismo de rastreamento e Sistema de Alertas.

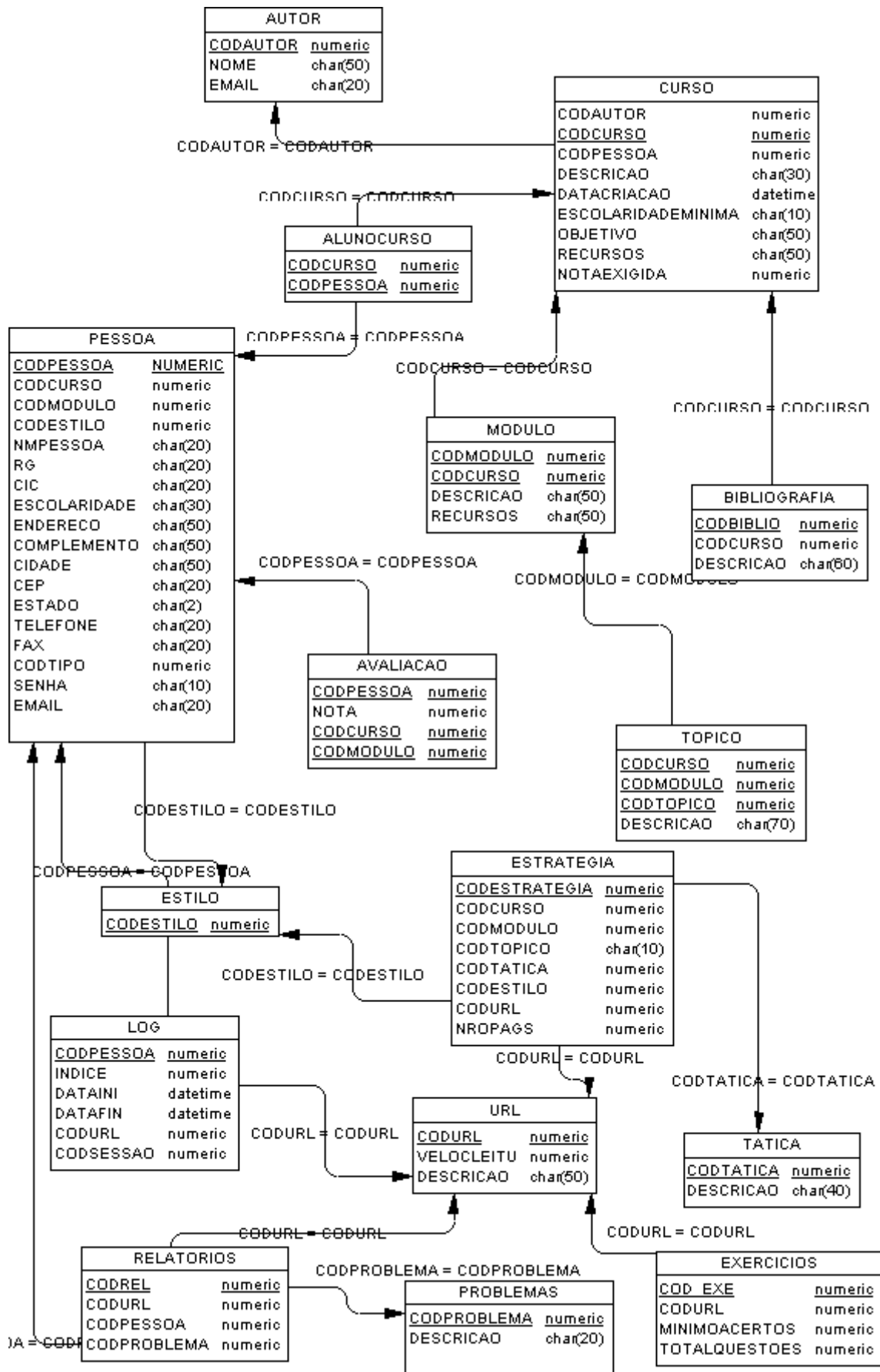


FIGURA 5.4– Modelo Entidade-Relacionamento

### 5.3 Base de Conhecimento

O Sistema de Alertas tem como núcleo uma base de conhecimento, que consiste numa coleção de fatos, definições e regras heurísticas que representam todas as situações de exceção que devem ser detectadas pelo sistema e as ações correspondentes.

Conforme LENAT e CHANG, heurística é um pedaço de conhecimento capaz de sugerir ações plausíveis a seguir ou ações implausíveis a evitar [LEN 88] [CHA 87]. Em outras palavras, heurísticas nada mais são que regras informais de julgamento que guiam as pessoas numa rápida tomada de decisão. Assim, as heurísticas levam ao objetivo em menos tempo, através de melhores escolhas.

O conhecimento utilizado no sistema de alertas é descrito na forma de regras, chamadas regras de produção. Estas regras são simplesmente um conjunto de condições no estilo SE..ENTÃO.. com a possibilidade de inclusão de conectivos lógicos relacionando os atributos no escopo do conhecimento. Boa parte do conhecimento necessário para reagir aos eventos detectados, durante a realização do curso, pode ser expressa desse modo.

#### 5.3.1 As Regras de Produção

O projetista criador da base de conhecimento deve seguir um critério para determinar as assertivas. As regras definidas devem estar de acordo com a sintaxe estabelecida pela gramática, como mostra a figura 5.4. A gramática utilizada está definida no Anexo 3.

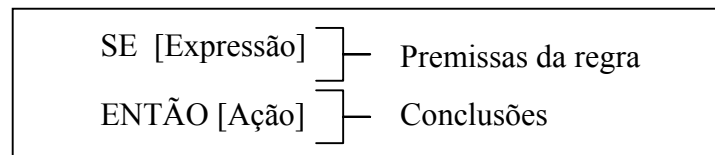
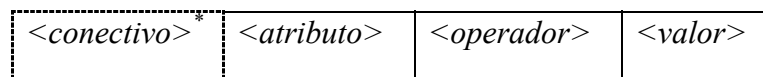


FIGURA 5.5- Regra de Produção

Denominam-se os conseqüentes de uma regra de “cabeças da regra”, e os antecedentes de “caudas”.

A estrutura de cada expressão (premissa) deve obedecer o seguinte modelo:



onde:

*conectivo* é um dos seguintes elementos utilizados na lógica clássica: NÃO, E, OU. Sua função é unir a sentença ao conjunto de premissas que formam a seção de antecedentes de uma regra.

*atributo* é uma variável capaz de assumir uma ou múltiplas instâncias no

---

\* Opcional

decorrer da consulta à base de conhecimentos. Cabe ao desenvolvedor definir o tipo de atributo. Um atributo é uma entidade totalmente abstrata, capaz de armazenar listas de valores cujo significado depende do contexto da base. Por exemplo, um atributo *TipoPag*, no sistema de alertas, é capaz de ser instanciado a qualquer um ou vários elementos da lista de valores pré-definida [*conceito, exemplo, exercício*]. Números e procedimentos também podem ser atribuídos a variáveis;

*operador* é um elo de ligação entre o atributo e o valor da premissa que define o tipo de comparação a ser realizada. São operadores relacionais: =, >, <=, <>, entre outros;

*valor* é um item de uma lista previamente criada e relacionada a um atributo. Como veremos posteriormente, a interface visual do Editor de Alertas foi idealizada de modo a minimizar o trabalho do responsável pela implementação da base de conhecimentos. Para tanto, o sistema exige que os atributos sejam definidos antes de criarmos uma regra que o utilize. No momento da criação de um atributo, também é possível que seja definida uma lista de valores que constituirá o universo ao qual suas instâncias devem necessariamente pertencer. Um valor também pode ser um número, caso o atributo o permita.

A estrutura de cada cabeça (conclusão) deve obedecer ao seguinte modelo:

<ação>

onde:

*ação* equivale a um procedimento já estabelecido anteriormente, como: *enviar e-mail*, mensagem na tela, mensagem para agente, etc.

Novas regras podem ser adicionadas à base de conhecimento assim como regras antigas podem ser removidas ou modificadas. Esta alteração na base de conhecimento pode ser feita diretamente pelo especialista humano (coordenador/responsável do curso) ou pelo especialista de conhecimento (professor do curso) que recebe a informação neste sentido do especialista e codifica-a de forma apropriada para a inclusão. Sabe-se que menos regras implicam geralmente um menor número de casos detectados.

Apesar deste trabalho diferir base de dados e base de conhecimento, é importante ressaltar que as regras estão sendo armazenadas, também, em uma banco de dados relacional. A terminologia “base de conhecimento” continuará sendo adotada para identificar claramente o tipo de informação que está sendo discutida.

As regras utilizadas pelo sistema de alertas foram definidas juntamente com os psicólogos e pedagogos, pesquisadores do projeto e são apresentadas no Anexo 1.

## 5.4 Comunicação entre o Sistema de Alertas e Agentes do Tapejara

Para que ocorra a comunicação entre o Sistema de Alertas e agentes Tapejara é necessário estabelecer uma linguagem de comunicação entre eles.

Uma das linguagens de comunicação entre agentes que vem sendo amplamente utilizada é a KQML (*Knowledge Query and Manipulation Language*), desenvolvida

pelo projeto DARPA *Knowledge Sharing Effort* [DAR 93].

A KQML é uma linguagem para troca de informações e conhecimento entre agentes e caracteriza-se, tanto por uma formatação de mensagens, como por um protocolo, para manipulá-las, suportando, em tempo de execução, o compartilhamento de conhecimento e informações entre agentes.

KQML pode ser utilizada como a linguagem que um programa aplicativo utiliza para interagir com um sistema inteligente, ou para dois ou mais sistemas inteligentes compartilharem conhecimento para solução cooperativa de problemas[HÜB 97][HAD 95].

Esta linguagem enfoca um grande conjunto de mensagens pré-definidas (*performativas*), as quais definem as operações possíveis de serem executadas pelos agentes. Assim, aqueles que estão em conformidade com KQML podem responder a estas mensagens de maneira adequada, independentemente da estrutura do seu emissor.

Em suma, a KQML é um protocolo formado por mensagens pré-formatadas com significado e funções bem específicos. Um agente, ao receber uma mensagem KQML, toma uma ação com base no seu significado e conteúdo.

Os agentes que estão de acordo com a KQML não precisam reconhecer todas as mensagens, de forma que um pequeno subconjunto pode ser suficiente num determinado sistema, ou seja, dependendo da necessidade, pode-se escolher somente algumas *performativas* a serem utilizadas na comunicação. Uma mensagem KQML (*performativas*) é expressa como uma *string* de caracteres ASCII, usando uma sintaxe definida [DAR 93].

Informações adicionais sobre KQML podem ser obtidas em [DAR 93] [BRA 97] [FIN 96].

A comunicação entre o sistema de alertas e o agente Tapejara ocorre através de uma definição de mensagem baseada em KQML. Esta abordagem segue o trabalho desenvolvido por [BIC 2000].

## 5.5 Diagrama de Classes do Protótipo Implementado

O serviço de alertas é composto de uma série de objetos que, juntos, são capazes de emitir um alerta, quando uma situação específica ocorre.

Estes objetos são:

- *Alerta*, que guarda um conjunto de objetos *Condição*, que representam cada uma das condições que fazem a lógica do alerta ser verdadeira;
- uma série de objetos *Invocar*, que guardam as condições em que um alerta será ativado;
- *Observador*, que apresenta uma visualização do resultado do alerta.

Cada um destes objetos será descrito a seguir:

### 5.5.1 Classes Alerta e Condição

O serviço de alertas usa um conjunto de regras como formalismo de representação de conhecimento. A classe *Alerta* contém informação de documentação, como título, autor, versão, etc. e é composta por um ou mais objetos *Condição*. O alerta é gerado sempre que uma condição representada em um objeto *Condição* se torna verdadeira. Cada condição é representada por um texto em uma linguagem criada para expressar premissas de regras. Regras como "a nota do aluno é maior do que 7" podem ser expressas usando esta linguagem. O objeto *Condição* tem um método *VerificaSintaxe*, que é usado para verificar a sintaxe da propriedade *texto\_regra*, e um método *VerificaExpressão*, usado para verificar se a expressão na propriedade *texto\_regra* é verdadeira. O método *Executar* do objeto *Alerta* executa cada método *VerificaExpressão* para cada objeto *Condição* e, se ao menos uma delas for verdadeira, deve-se fornecer o tempo(data e hora) em que o alerta foi gerado e algum dado (como uma frase "nota do aluno é 8", por exemplo). A figura 5.5 mostra as classes *Alerta* e *Condição*.

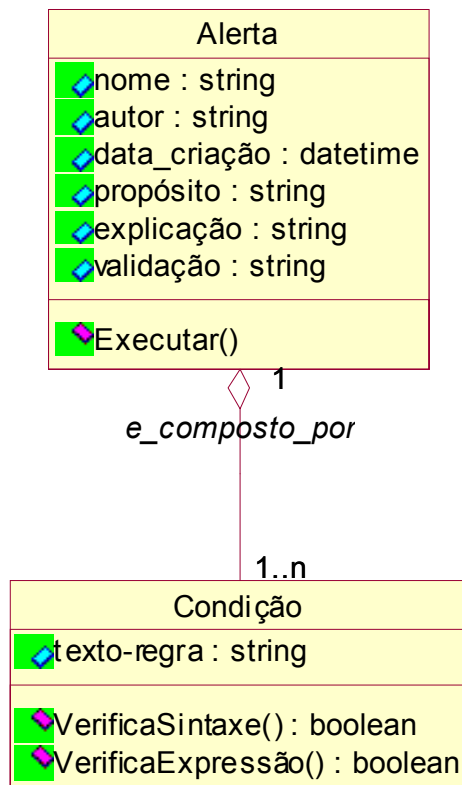


FIGURA 5.6 - As classes Alerta e Condição

### 5.5.2 Classes CondiçãoTempo, CondiçãoMensagem e CondiçãoTempoMensagem

Um sistema de alertas é tipicamente um sistema ativo. Isso significa que o sistema deve reagir quando alguma coisa ocorre. As condições de ativação especificam condições em que um alerta deveria ser ativado. Nesta representação são permitidos três tipos de ativação:

1. Quando uma condição temporal é verdadeira. Por exemplo, um alerta pode



ser invocado a cada 5 horas, iniciando em 5/3/01, às 12:00, até 10/3/01, às 15:00. Este tipo de invocação pode ser expresso pelo objeto *CondiçãoTempo*.

2. Quando uma mensagem é gerada. Por exemplo, um alerta poderia ser invocado quando um novo dado do aluno está disponível. Este tipo de invocação pode ser expressa pelo objeto *CondiçãoMensagem*.
3. Quando uma mensagem é gerada e uma restrição de tempo é verdadeira. Por exemplo, um alerta poderia ser invocado 5 horas depois que o novo dado do aluno está disponível, durante 5 semanas. Este tipo de invocação pode ser expressa pelo objeto *CondiçãoTempoMensagem*.

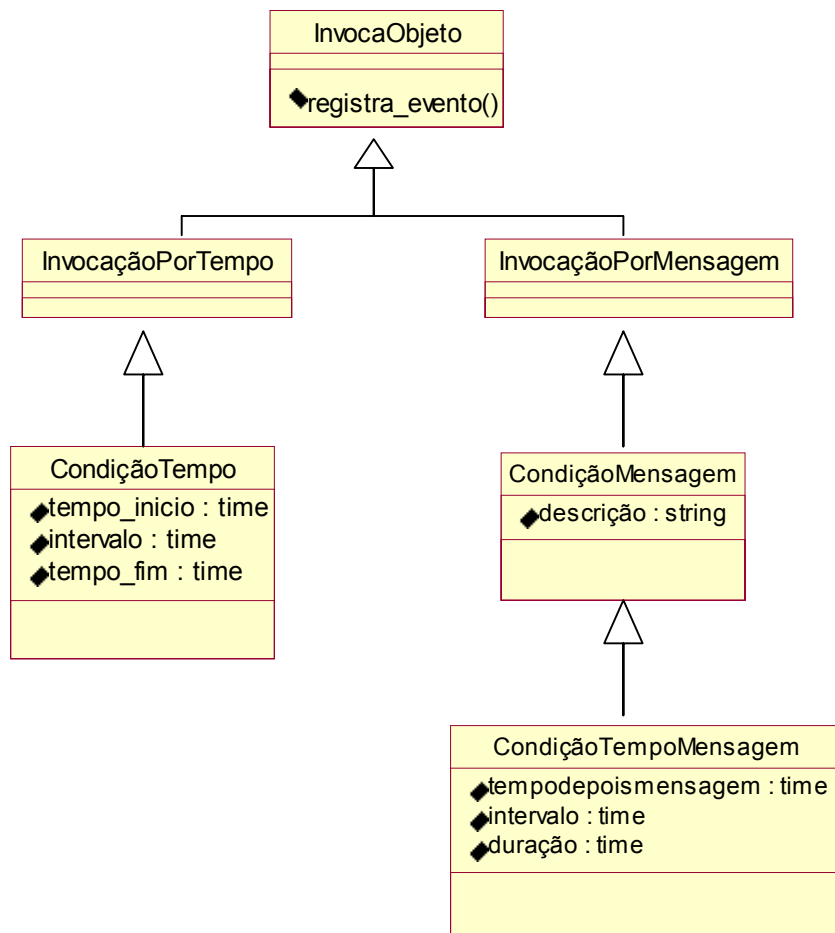


FIGURA 5.7- As classes *CondiçãoTempo*, *CondiçãoMensagem* e *CondiçãoTempoMensagem*

### 5.5.3 Classe Observador

Este objeto possui um método *EnviaMensagem* que mostra uma mensagem baseada na conclusão do alerta. O *observador* é muito dependente da implementação. Se um sistema deseja enviar um *e-mail* para o professor responsável pelo curso com a

conclusão do alerta, um novo *observador* deve ser criado para suportar esta característica. De fato, um *observador* pode ser usado em vários objetos deste modelo.

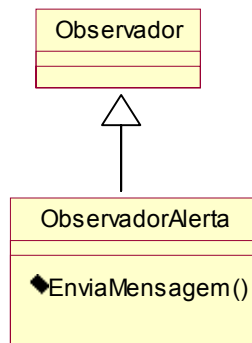


FIGURA 5.8 - A classe Observador

Através desta estrutura, o professor terá uma visão bastante ampla no que se refere a forma como o aluno vem participando do curso e se está sentindo estimulado pela maneira como o conteúdo lhe é apresentado.

## 5.6 Funcionamento do Sistema de Alertas

A seguir, é mostrado o funcionamento do Sistema de Alertas especificado anteriormente, e como o protótipo que coloca em prática esta estrutura foi implementado.

### 5.6.1 Processo de Execução do Sistema de Alertas

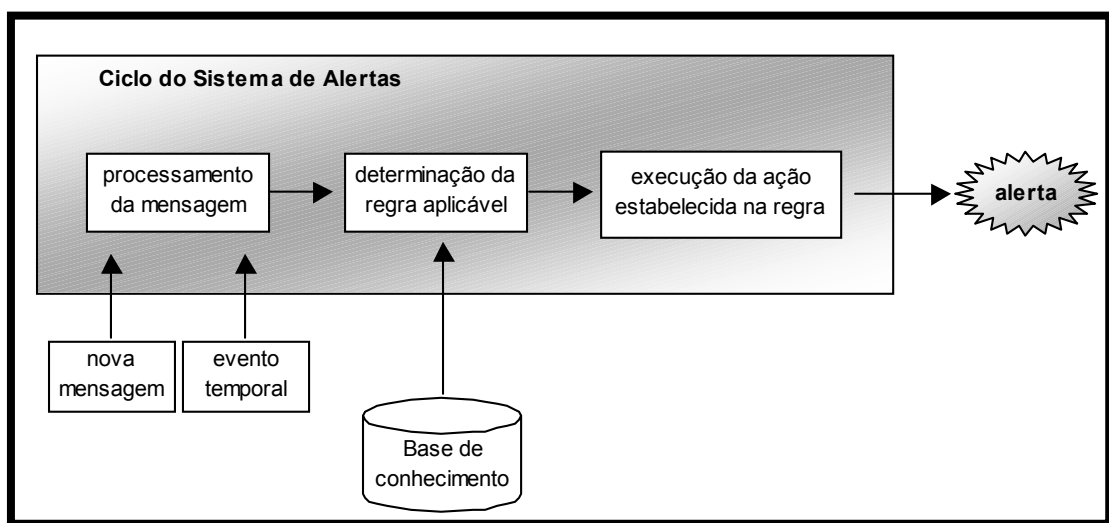


FIGURA 5.9 - Processo de Execução do Sistema de Alertas

Após recebimento da mensagem de aviso, que informa a ocorrência de um evento de interesse, ou detecção de evento temporal, o sistema executa um conjunto de operações para emissão dos alertas. As operações, mostradas na Figura 5.8, estão

descritas a seguir na seqüência em que são executadas.

- processamento da mensagem recebida: neste momento, o sistema decompõe a mensagem KQML e reconhece qual a tarefa que esta propõe.
- operação de acesso a base de conhecimento: a partir da análise da mensagem, o sistema de alertas consulta a base de conhecimento e determina qual regra é aplicável para a corrente situação;
- execução da ação estabelecida na base de conhecimento: cada regra existente na base de conhecimento possui uma ação associada. É neste momento que tais ações são executadas pelo sistema de alertas.

## **5.6.2 Eventos Percebidos**

Os eventos são indicadores de situações anormais detectadas a partir do processo de monitoração. Os eventos percebidos pelo Sistema de Alertas são ações realizadas pelos alunos e estão sendo controladas e armazenadas na base de dados pelo Mecanismo de Rastreamento explicado na seção 5.2.

Os eventos detectados pelo Sistema de Alertas são de três tipos: mensagens, eventos temporais e mensagens/tempo.

### **5.6.2.1 Mensagens**

O Sistema de Alertas fica ocioso à espera de algum aviso do Mecanismo de Rastreamento que seja interessante monitorar. Este aviso pode ser uma mensagem que informa quando o aluno loga-se no sistema ou a percepção de transições de estado no banco de dados como a inclusão de um novo registro, ou a alteração de algum dado importante.

### **5.6.2.2 Eventos Temporais**

O Sistema de alertas realiza consultas no banco de dados de acordo com as especificações de tempo registradas no formulário Alertas.html, apresentado na seção 5.7.3. Se alguma condição de tempo for satisfeita, a regra associada é disparada e o alerta enviado.

### **5.6.2.3 Eventos Temporais associado com Mensagens**

Neste caso, após inclusão ou alteração em algum registro de interesse, o sistema é avisado pelo Mecanismo de Rastreamento e, antes de disparar a regra, verifica se a condição temporal também é satisfeita, determinando se a ação estipulada para esta regra pode ser disparada.

### 5.6.3 Tipos de Mensagens

O Sistema de Alertas trabalha basicamente com o processo de troca de mensagens. As mensagens recebidas pelo sistema informam quando um determinado evento ocorrer. As mensagens enviadas são os alertas propriamente dito.

Todas as mensagens recebidas e enviadas pelo Sistema de Alertas são apresentadas no Anexo 2.

#### 5.6.3.1 Mensagens Recebidas

As mensagens de aviso recebidas pelo sistema de alertas são:

- usuário loga-se no sistema;
- inserção de data final de acesso à página;
- término do módulo;
- término de exercício;
- inserção da nota de avaliação.

#### 5.6.3.2 Mensagens Enviadas

As mensagens emitidas pelo Sistema de Alertas estão armazenadas em uma base de dados chamada base de mensagens. Elas podem ser de três tipos:

1) mensagem na tela

- dicas;
- perguntas;
- incentivo;
- congratulações;
- avisos.

2) *e-mail*

- para professor;
- autor/responsável pelo material didático;

3) mensagem para o Agente.

## 5.7 Implementação do Protótipo

A fim de validar o sistema de alertas especificado, foi desenvolvido um protótipo. Neste capítulo é apresentado o ambiente em que foi desenvolvido e implementado este protótipo incluindo os requisitos necessários para o projeto do protótipo e a descrição de cada parte que o compõe.

### 5.7.1 Requisitos para Implementação do Sistema de Alertas

Segundo HINZE em [HIN 99a], alguns critérios devem ser considerados para a criação de um Sistema de Alertas:

- Os sistemas de alertas devem ser robustos e fornecer flexibilidade para conectar várias bases de dados;
- Devem emitir avisos às pessoas indicadas no momento da ocorrência do evento de interesse. Alertas também podem ser enviados, se um determinado evento não ocorrer;
- Um sistema de alerta deve ter habilidade para gerar alertas para qualquer outro sistema;
- Os alertas devem ser gerados a partir de qualquer base de dados, aplicação ou sistema existente. Isto permite a monitoração dos eventos entre diversos banco de dados;
- Os alertas devem permitir consultas do tipo SQL para identificar os eventos nas bases de dados relacionais, emitir avisos significativos e recomendar ações para muitos sistemas e eventos;
- Os alertas devem ter capacidade de acessar recursos distribuídos na Internet;
- Os alertas devem ser criados e controlados centralmente. Não se pode controlar facilmente alertas residentes em vários sistemas, nem fornecer uma visão detalhada de como os alertas estão controlando o ambiente. Os usuários devem especificar as regras para geração dos alertas, monitorar a sua eficácia e alterá-los, baseados nas condições estabelecidas;
- Se possível, um sistema de alertas deve oferecer uma interface gráfica padrão e independente de plataforma, para ser utilizado em qualquer ambiente;
- Um sistema de alertas deve ter uma arquitetura separada, permitindo uma construção rápida e de fácil manutenção. Deve suportar uma variedade de dispositivos de entrada e saída, permitindo a extensão em todo o ambiente corporativo.

A linguagem de programação escolhida para o desenvolvimento do protótipo deveria cumprir estes requisitos. Por esse conjunto de razões, a linguagem Java foi selecionada como ferramenta de desenvolvimento. Além dessas características, Java é orientada a objetos e vem se tornando uma tendência para desenvolvimento de sistemas na Internet.

## 5.7.2 Ambiente de Implementação do Sistema de Alertas

O protótipo foi totalmente desenvolvido utilizando-se a linguagem Java, linguagem utilizada por todo o grupo de pesquisa dentro do qual este trabalho foi desenvolvido. Além de cumprir os requisitos de implementação exigidos e explicados na seção 5.7.

Para a base de dados e base de conhecimento utilizou-se o banco de dados relacional *Access*, os quais também serão integrados no ambiente educacional do Projeto Tapejara.

O Sistema de Alertas troca mensagens com os agentes seguindo um protocolo baseado na linguagem KQML, descrita na seção 5.4.

O material instrucional é formado por lições, exemplos e exercícios e foi totalmente desenvolvido pelos pesquisadores do Tapejara.

As tecnologias Java, fundamentais para o desenvolvimento do protótipo, foram as *Servlets* e JDBC (*Java DataBase Connectivity*), descritas a seguir.

### 5.7.2.1 *Servlets*

A *Web* é uma imensa aplicação distribuída onde existem diversos servidores de conteúdo e clientes que acessam este conteúdo. Cada par servidor *Web-browser* corresponde a um par cliente-servidor. O conteúdo das páginas da *Web* pode ser estático ou dinâmico. Quando o conteúdo é estático, os documentos HTML residentes no servidor são simplesmente entregues aos clientes mediante sua requisição e renderizados nos clientes pelos próprios browsers.

Quando o conteúdo é dinâmico ele pode ser criado no próprio cliente, através do *JavaScript*, *Dynamic HTML* ou *applets Java* numa situação caracterizada como *client-side* ou *client-end*. O conteúdo dinâmico pode também ser criado do lado do servidor através de programas ou *scripts CGI*, aplicações *server-side* e agora *servlets*.

Os *servlets* fornecem um mecanismo simples e consistente para estender a funcionalidade de um servidor *Web*. Uma *servlet* pode ser pensada como uma pequena aplicação que roda no servidor. A API utilizada na escrita das *servlets* não se preocupa em como ela é carregada, qual o ambiente em que ela rodará ou o protocolo usado no envio e recebimento de informações. Isto faz com que as *servlets* possam ser incorporadas em vários servidores *Web* diferentes. Enquanto as *applets* operam do lado do cliente e não tem acesso aos recursos existentes na máquina do cliente, *servlets* tem acesso aos mesmos recursos disponíveis para as aplicações *server-side*, isto é, acesso ao sistema de arquivos, bancos de dados locais ou outras conexões de rede [DAV 2001].

O *servlet* é carregado uma única vez na memória, e permanece lá até que seja necessário modificá-lo. Uma vez inicializado, o *servlet* estará apto a lidar com centenas de acessos simultaneamente, disparando para cada acesso uma nova *thread*. As *threads* de um mesmo aplicativo, utilizam um espaço de endereçamento de memória comum a todos. Isto permite que eles compartilhem dados e recursos do sistema. Ou seja, todas as *threads* de um *servlet* podem fazer uso de uma única conexão que foi estabelecida com um banco de dados no momento de inicialização do *servlet*. Esta conexão permanecerá aberta até que o *servlet* seja desativado, saia da memória, ou seja recarregado [COL 98].

Para criar um *servlet* utiliza-se a API *javax.servlet* que define a interface *Servlet*. Todo *servlet* deve ser uma implementação desta interface, ou ser uma subclasse de uma classe que a tenha implementado. Nela estão definidos todos os métodos para estabelecer a comunicação com clientes. Um exemplo de classe que implementa a interface *Servlet* é a *HttpServlet*, utilizada para desenvolver *servlets* que são instalados em servidores *Web* e que utilizam o protocolo HTTP para enviar e receber informações [FIO 2000] .

A comunicação entre um *servlet* e um cliente se dá por meio de dois objetos instanciados das classes: *ServletRequest* e o *ServletResponse*. O primeiro encapsula as funções de comunicação do cliente para o servidor, permitindo que o *servlet* receba dados como o conteúdo de um formulário HTML. O segundo, encapsula as funções de comunicação do servidor para o cliente [COL 98].

A estrutura mínima de um *servlet* é composta por três métodos que devem ser implementados. São eles: *init*, *service* e *destroy*[ COL 98].

O *init* é o primeiro método a ser executado no momento em que o *servlet* está sendo carregado, nele ocorrem a leitura de configurações estabelecidas pelo sistema, a inicialização de estruturas de dados e, se for o caso, estabelece conexão com uma base de dados que será utilizada pelo *servlet* durante a sua execução. Neste processo de inicialização o servidor pode passar parâmetros como por exemplo nome da base de dados ou diretórios de arquivos temporários. O método *init* recebe como argumento um objeto *ServletConfig* utilizado para obter os parâmetros de inicialização.

Após o término da inicialização, o *servlet* esta apto a receber e processar requisições de acesso de clientes. O processamento dos pedidos ocorrem no método *service*, que recebe informações do cliente, realiza o processamento, e envia uma resposta. Cada acesso faz com que o *servlet* crie um *thread* para executar o método *service*.

Quando um *servlet* é removido do servidor o método *destroy* é executado. Este método deve ser utilizado para fazer operações de ‘limpeza’ das estruturas de dados, gravar fisicamente o estado do *servlet* ou desalocar outros recursos do sistema que estejam sendo utilizados.

#### 5.7.2.2 JDBC

JDBC (*Java Database Connectivity*) é uma API que permite o acesso à diferentes fontes de dados através da programação Java [JAV 99][PER 99]. Ela consiste em um conjunto de classes e interfaces escritas em Java, fornecendo conectividade para uma grande variedade de bancos de dados SQL, planilhas eletrônicas ou arquivos. A tecnologia JDBC permite o envio de comandos SQL a qualquer banco relacional existente, de forma única. Em outras palavras, com a API do JDBC, é possível escrever um único programa Java que acesse uma base de dados Oracle, Sybase ou SQL Server, por exemplo.

Com o JDBC é possível executar três tarefas:

- Estabelecer uma conexão com o banco de dados;
- Enviar comandos SQL;

- Processar os resultados.

### 5.7.3 A Interface

O Sistema de Alertas funciona continuamente entre a monitorização de dados contidos na base de dados e a espera de eventos. Eles são desenvolvidos para testar dados específicos conforme critérios pré-definidos na base de conhecimento. Se houver um cruzamento entre um dado na base de dados e um critério, o sistema irá emitir um alerta, podendo informar as providências que devem ser tomadas.

Uma interface visual foi desenvolvida para testar a funcionalidade do Sistema de Alertas. A interface contém um editor para os alertas, onde o usuário registra as informações de documentação, como título, autor, data criação, etc. Os eventos, as regras e ações associadas aos alertas também são definidas aqui.

A figura 5.10 apresenta a interface do Editor de Alertas. Esta interface é um arquivo HTML denominado Alertas.html, que está disponível em um servidor do Instituto de Informática da UFRGS.

**Alerta** Nome

Autor

Data Criação (dd-mm-aaaa)

Propósito

Explicação

**Condição** Texto Regra

FIGURA 5.20 - Interface do Editor de Alertas



**Condição Tempo**

Tempo Início (dd-mm-aaaa hh:mm)

Intervalo (valor inteiro)

Tempo Fim (dd-mm-aaaa hh:mm)

**Condição Mensagem**

Descrição

**Condição Tempo Mensagem**

Tempo Depois Mensagem (hh:mm:ss PM/AM)

Intervalo (hh:mm:ss PM/AM)

Duração (hh:mm:ss PM/AM)

Document: Done

FIGURA 5.11– Interface do Editor de Alertas (continuação)

Através deste formulário, o usuário pode configurar e ajustar as regras de geração de alertas. Isto deve ser realizado por um especialista de conhecimento, o qual deve conhecer muito bem as regras e condições a serem criadas. É aqui que ele registra no sistema se necessita ser informado quando uma certa condição de exceção ocorrer. A condição de exceção e a ação de envio da mensagem aconselhável, assim como o receptor e o meio de recebimento da mensagem, também são configuradas neste formulário. O sistema também verifica se a sintaxe da regra está correta, de acordo com a gramática usada. As regras podem ser agendadas para serem executadas em qualquer intervalo de tempo.

Com estas informações o sistema de alertas inicia o processo de monitoração dos eventos.

#### 5.7.4 Operações do Sistema de Alertas

O Sistema de Alertas executa um conjunto de operações, desde o registro dos alertas desejados pelo usuário, até a operação de envio da mensagem de alerta. As operações estão descritas a seguir:

##### 5.7.4.1 Operação de Acesso à Base de Dados

A figura 5.12 apresenta um trecho do código da implementação da classe de conexão à base de dados.

```
protected String dbURL = "jdbc:odbc:alertas";
protected String userID = "";
protected String passwd = "";
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
System.out.println("DBServlet: Conectando com - " + dbURL);
dbConnection = DriverManager.getConnection(dbURL, userID, passwd);
```

FIGURA 5.32 – Código Java mostrando acesso JDBC

#### 5.7.4.2 Operação de acesso ao banco de dados para registro dos alertas

A figura 5.13 apresenta um fragmento de código Java que implementa a operação de acesso ao banco de dados para registro dos alertas desejados, especificados na interface apresentada na figura 5.10.

```
System.out.println("DBServlet : Preparando para registrar dados");
registerStatement =dbConnection.prepareStatement("insert into Alerta "
+ "(nome, autor, data_criacao, proposito, explicacao, validacao)"
// + "(texto_regra, descricao, tempo_inicio, intervalo, tempo_fim)"
// + "(tempodepoismensagem,intervalo2,duracao)"
+ " values (?, ?, ?, ?, ?, ?)");
// + "(?, ?, ?, ?, ?, ?)");
registerStatement2 =dbConnection.prepareStatement("insert into Condicao (texto_regra)" + "
values (?");
System.out.println("Tabela Condicao OK");
```

FIGURA 5.43 – Código Java mostrando operação de registro dos alertas

#### 5.7.4.3 Operação de acesso à base de dados para detectar ocorrência de evento

O Sistema de Alertas detecta os eventos de duas formas:

- consulta a base de dados para identificar acontecimentos que podem causar a emissão de um aviso;
- consulta as mensagens KQML enviadas pelo Agente comunicação.

#### 5.7.4.4 Operação de processamento da mensagem KQML recebida

O Sistema de Alertas ao perceber que existe uma mensagem KQML direcionada para ele, a decompõe e realiza a operação de acesso a base de conhecimento para descobrir qual a tarefa que esta propõe.

A implementação da mensagem KQML, segue o trabalho de [BIC 2000] e é composta por dois módulos: um formato geral que possui especificação KQML (classe KQMLMessage) e um formato específico que compõe a tarefa que o agente deve executar (classe ContentMessage).

KQML possui um conjunto de parâmetros chaves reservados que servem para estabelecer uma uniformidade dos parâmetros e suporte para que os programas entendam perfomativas desconhecidas mas com parâmetros conhecidos. Os parâmetros foram implementados na forma de uma classe Java chamada KQMLMessage, como mostra a tabela 5.1.

TABELA 5.175- Classe KQMLMessage [BIC 2000]

Classe KQMLMessage	
performative	nome da performativa
sender	transmissor
receiver	receptor
language	linguagem do conteúdo
ontology	ontologia do conteúdo
replay_to	re-enviar a mensagem para
replay_with	rótulo da resposta
force	se performativa negada
content	conteúdo – Objeto ContentMessage

De acordo com o formato de mensagem KQML, é necessário especificar o parâmetro *content*, para isso, usa-se a classe *ContentMessage*. A classe *ContentMessage*, especificada por [BIC 2000], foi alterada para servir o Sistema de Alertas. A Tabela 5.2 mostra a classe *ContentMessage* utilizada no Sistema de Alertas com todos os parâmetros necessários para compor as mensagens. Tais parâmetros são utilizados conforme o tipo de mensagem usada.

TABELA 5.16 – Classe ContentMessage

Classe ContentMessage	
type	tipo da mensagem
operation	operação realizada pelo usuário (exemplo: 1= logon)
user	codpessoa - código do aluno
url	codurl – endereço da página acessada pelo aluno
time	tempo de permanência na página
course	código do curso que está o aluno
numacertos	número de acertos nos exercícios
session	número da sessão de estudo
problem	tipo de problema detectado (exemplo: 1= erro exercício)

#### 5.7.4.5 Operação de acesso a base de conhecimento

O Sistema de Alertas realiza a operação de acesso a base de conhecimento para descobrir quais regras são aplicáveis para o evento corrente. A figura 5.14 apresenta um exemplo de código Java que realiza esta operação.

```

queryBase = dbConnection.prepareStatement("select texto_regra from Condicao c,
CondicaoMensagem cm where c.codalerta=cm.codalerta e descricao = " + eventoMsg);
ResultSet Base = db.executeQuery(queryBase);
while (Base.next()) {
    String s = Base.getString("texto_regra");
    separa_str (s); }

```

FIGURA 5.54– Código Java mostrando acesso à base de conhecimento

#### 5.7.4.6 Operação de envio de *e-mail*

A figura 5.15 apresenta um trecho de código da implementação da operação de envio de *e-mail*, como resultado da ação proposta pela regra contida na base de conhecimento.

```

public void enviemail(String email,String nome,String assunto,String conteudo){
    Socket smtp = null;
    try{
        smtp = new Socket("caracol", 25);
        OutputStream os = smtp.getOutputStream();
        ps = new PrintStream(os);
        InputStream is = smtp.getInputStream();
        dis = new DataInputStream(is);
    }catch (IOException e)
    {
        System.out.println("Erro");
    }

    try{
        String loc = InetAddress.getLocalHost().getHostName();
        send(HELO + loc);
        receive();
        send(MAIL_FROM + de);
        receive();
        send(RCPT_TO + email);
        receive();
        send(DATA);
        receive();
        send("From: " + denome);
        send("To: " + nome);
        send(SUBJECT+assunto+"\r\n");
        receive();
        send(conteudo+BODY);
        receive();
        smtp.close();
    }catch(IOException e)
    {
        System.out.println("Erro 2");
    }
    System.out.println("OK => EMAIL ENVIADO");
}
}

```

FIGURA 5.65– Código Java mostrando operação de envio de *e-mail*

#### 5.7.4.7 Operação de envio de mensagem no *browser* do aluno

O sistema envia uma mensagem para o aluno durante a sua navegação no curso, alertando-o de alguma situação.

#### 5.7.4.8 Operação de envio de mensagem KQML para agente

Caso o sistema detecte que é preciso enviar uma mensagem KQML para o agente (tutor) realizar alguma operação em especial, é necessário criar dois objetos

herdados respectivamente das classes `KQMLMessage` e `ContentMessage`.

```
KQMLMessage mensagem = new KQMLMessage( );
ContentMessage conteúdo = new ContentMessage( );
```

## 5.7.5 Exemplos de Alertas gerados pelo Sistema de Alertas

### 5.7.5.1 Alertas enviados ao browser do aluno

A figura 5.16 ilustra um exemplo de mensagem enviada para o aluno. O sistema de alertas verifica, após o *logon* do aluno, a data do seu último acesso ao curso, e se necessário, envia esta mensagem ao *browser*.

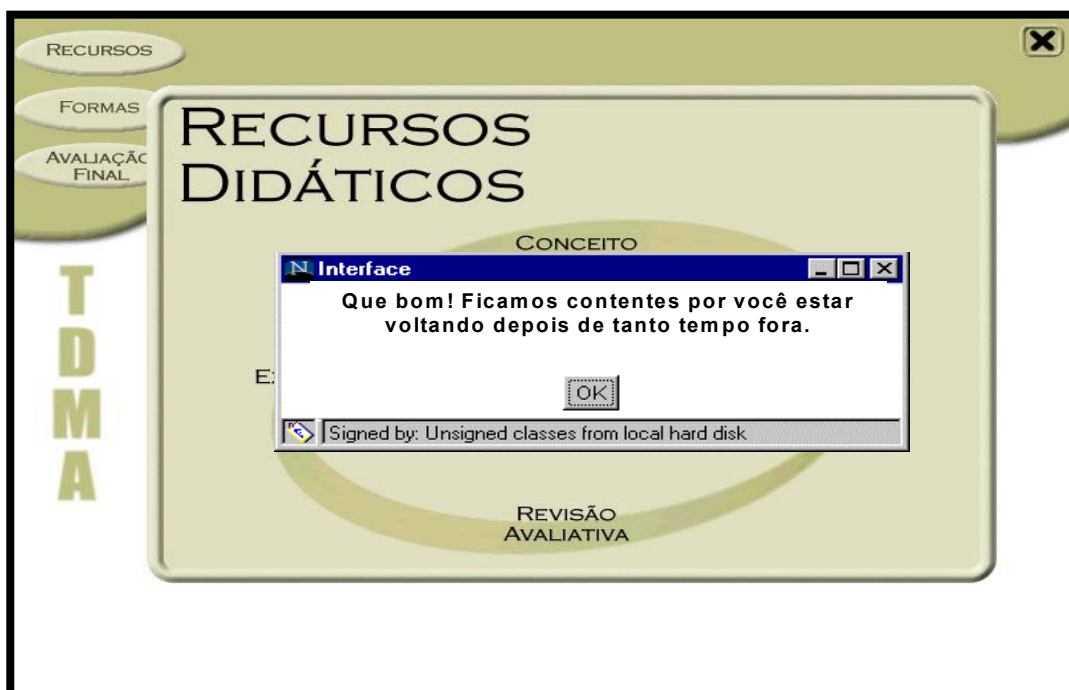


FIGURA 5.76– Exemplo de alerta “Retorno rápido ao curso”

A figura 5.17 mostra um exemplo de alerta enviada ao *browser* do aluno durante a sua interação no curso. Quando o Sistema de Alertas percebe que o aluno não está acessando a página de exercícios a seguinte mensagem é enviada.

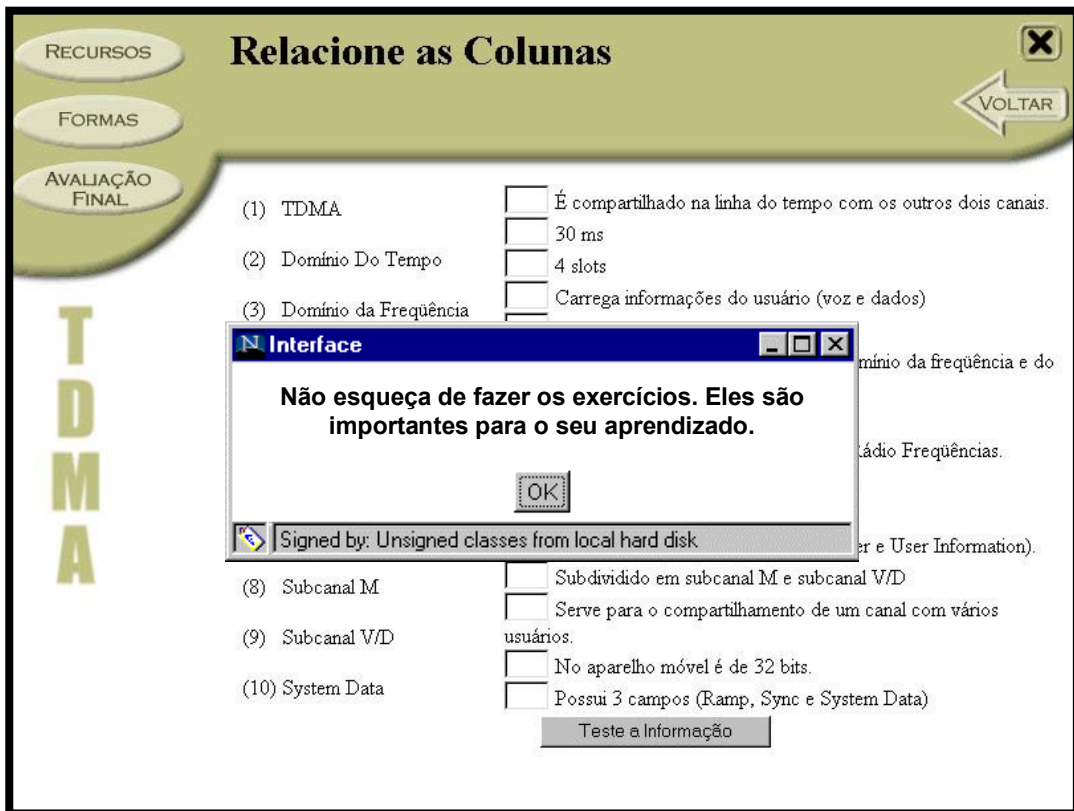


FIGURA 5.17- Exemplo de alerta “Pulo Exercício”

A figura 5.18 apresenta um exemplo de alerta chamado “Parada Brusca”. Este alerta é enviado quando o sistema detecta uma alteração no ritmo de estudo do aluno.

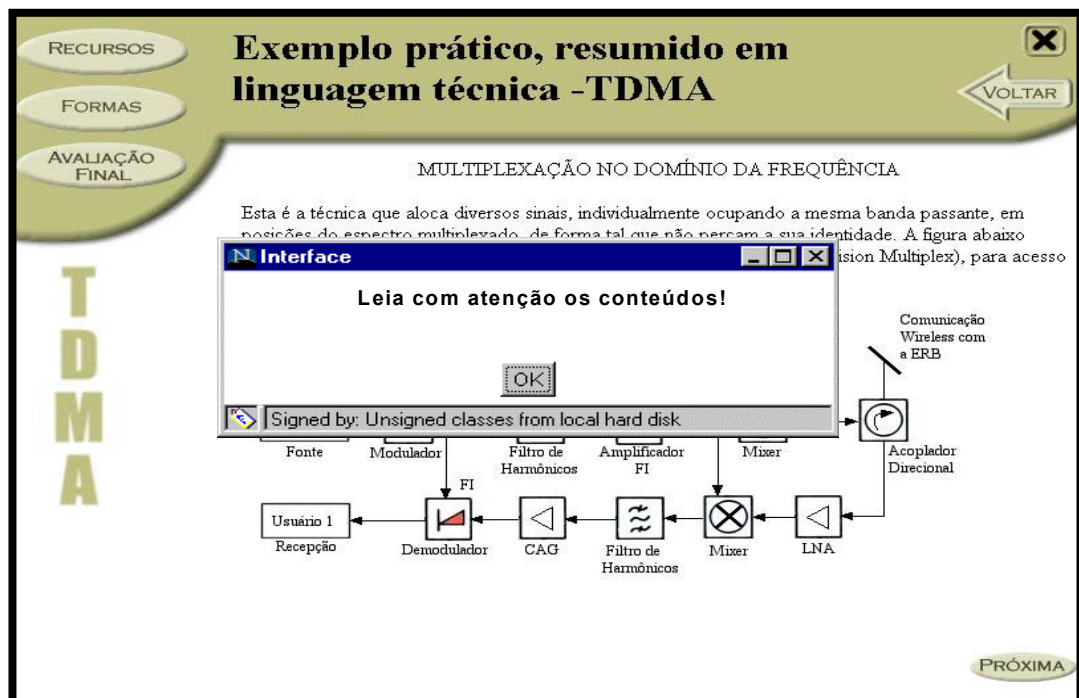


FIGURA 5.18– Exemplo de alerta “Parada Brusca”

### 5.7.5.2 Alertas enviados via *e-mail*

A figura 5.19 apresenta um exemplo de alerta enviada por *e-mail* para o professor responsável pelo curso. Este *e-mail* contém o Relatório Semanal sobre os acessos dos alunos no curso.

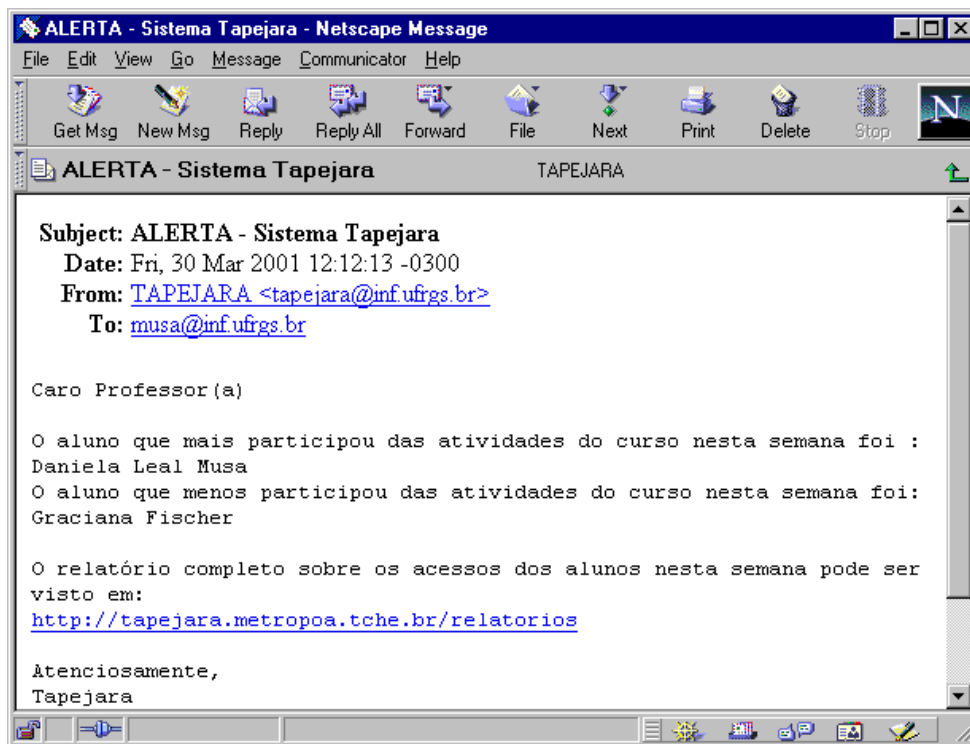


FIGURA 5.89 – Mensagem de alerta enviada ao professor, por *e-mail*

A figura 5.20 apresenta um exemplo de alerta enviada por *email* para um dos alunos do curso. Este *e-mail* é enviado quando o sistema detecta a ausência do aluno nas atividades.

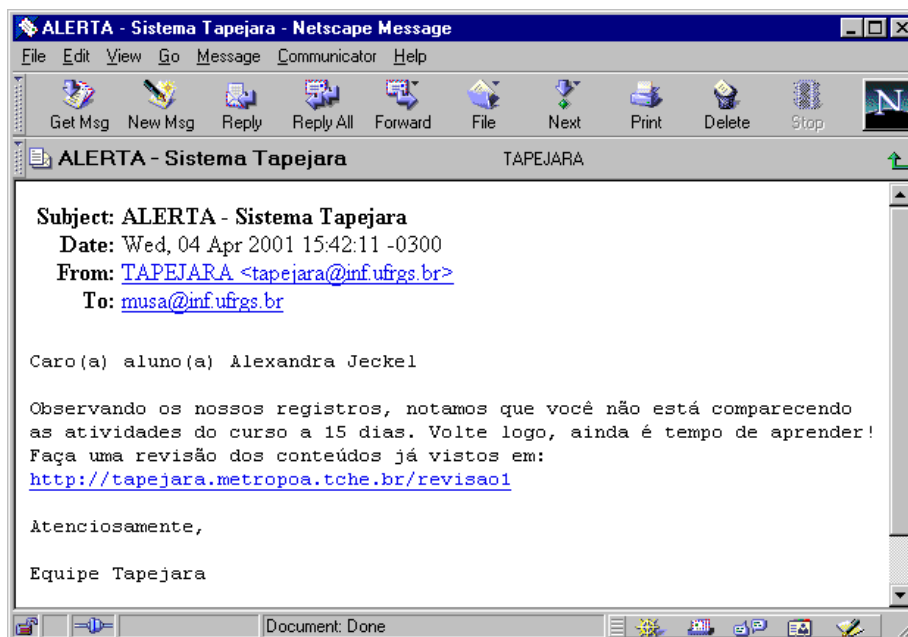


FIGURA 5.90 – Mensagem de alerta enviada ao aluno por *e-mail*

### 5.7.5.3 Alertas enviados ao agente

A figura 5.21 ilustra um exemplo de mensagem KQML enviada pelo Sistema de Alertas para o Agente Tapejara. Este alerta é enviado ao agente, quando o sistema percebe um grande número de erros nos exercícios feitos pelo aluno. O agente encarrega-se de tomar um decisão para esta situação.

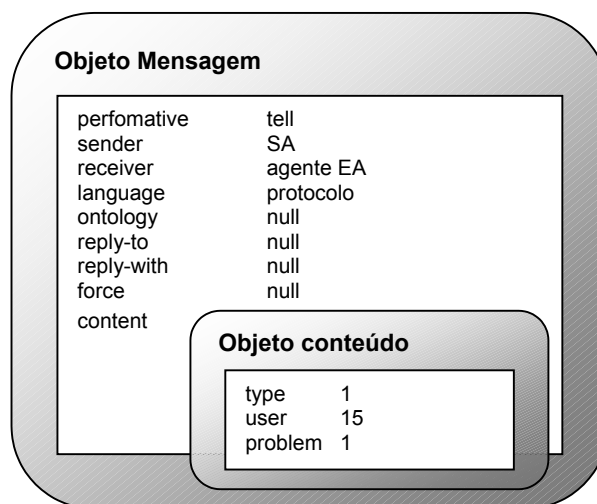


FIGURA 5.210 – Exemplo de mensagem enviada ao agente



## 6 Conclusão

Conscientes da importância de uma forma alternativa de ambientes de aprendizagem, de modo particular a Educação a Distância, que possibilita condições de acesso ao ensino a uma maior quantidade de pessoas, disponibilizando informação e conhecimento como um meio de democratização do saber, é muito relevante analisar de que forma tal sistema possa ser aperfeiçoado a fim de adequar-se às necessidades existentes, assim como superar as eventuais preocupações resultantes de uma inovação na área educacional.

Tais preocupações, basicamente, giram em torno do processo de avaliação. No sistema de Educação a Distância, como se sabe, os mecanismos de avaliação informais são bem pouco utilizados (sendo que, algumas vezes, são inexistentes) e isso faz com que o professor tenha dificuldades em ajustar seus procedimentos de ensino de acordo com as reações dos estudantes.

Dessa forma, o presente trabalho preocupou-se em encontrar uma solução para esta falha presente no processo de Educação a Distância. De acordo, então, com as possibilidades existentes, a utilização de um sistema de alertas seria uma opção que se adequaria aos devidos fins. Outras áreas, como a área médica, de gerência de redes e comercial, já adotaram o sistema de alertas para auxiliá-los em suas funções e obtiveram resultados bastante satisfatórios.

Na Educação a Distância, este sistema pode auxiliar o professor a ter um acompanhamento mais próximo das atividades dos alunos, pois além de informar sobre o rendimento dos alunos, também identifica aqueles que necessitam de ajuda, os que estão mais adiantados, assim como uma série de outras informações de muita utilidade para a evolução do ensino. Esse sistema também ajuda o aluno, estimulando-o nas suas atividades e auxiliando-o na detecção de problemas. O coordenador do material instrucional também é beneficiado, pois recebe avisos sobre as páginas que não estão sendo acessadas, etc.

Finalmente, apresentando o sistema de alertas implementado, assim como mostrando a aplicabilidade deste mecanismo no processo de Educação a Distância, concluímos que os alertas, sendo bem empregados, constituem uma forma simples e muito eficaz para aperfeiçoar o sistema de ensino a distância e permitir que seja cada vez mais utilizado. As contribuições do trabalho são:

- Modelagem de um Sistema de Alertas para o Ensino. Com base nos sistemas de alertas existentes em outras áreas (área médica, gerência de rede, projetos, bibliotecas digitais) foi desenvolvido um modelo de alertas específico para ensino a distância, o que ainda não existia.
- Integração do Sistema de Alertas em outros ambientes educacionais. O Sistema de Alertas apresentado neste trabalho foi proposto com o objetivo de ser integrado ao ambiente do Tapejara, pois foi modelado utilizando a arquitetura multiagente definida neste ambiente. No entanto, este sistema mostra-se adequado tanto para este ambiente como para a sua integração em outros ambientes educacionais, desde que estes possuam um mecanismo que armazene em um banco de dados as interações dos alunos no curso (log).

- A implementação do protótipo do Sistema de alertas, o qual pode ser integrado ao Tapejara tão logo os agentes deste ambiente estejam implementados. O Sistema de Alertas é parte fundamental para a funcionalidade do ambiente, adequando-se à proposta de um ensino adaptado ao aluno.
- Oferece aos professores acesso imediato aos resultados. Com o registro das informações em uma Base de Dados e a possível disponibilização destas informações via *e-mail* para os professores, torna-se fácil e rápida a observação do efeito de determinadas atividades sobre os alunos;
- Estímulo à participação: o aluno recebe avisos na tela e via *e-mail* incentivando o estudo, com isto ele se sentirá mais estimulado a participar das atividades que lhe são apresentadas;
- Facilidade de administração dos mecanismos: o trabalho do professor para administrar as atividades desenvolvidas pelos alunos não lhe acarretará aumento excessivo de suas atividades, pois o sistema envia por *e-mail* relatórios dos mais variados conteúdos. Com isto, o professor não precisa acessar o sistema para observar o andamento do curso, pois obtém, invariavelmente, informações de como os alunos vêm se portando no decorrer das atividades.
- Descobre problemas que podem estar ocorrendo nas atividades dos alunos no curso e avisa ao docente (tutor) a fim de que se realize a ação corretiva adequada.

Como resultado deste trabalho tem-se uma ferramenta que poderá servir de apoio aos professores, alunos e coordenadores de material instrucional em cursos que sejam ministrados a distância através da *Web*. Este sistema pode ainda ser utilizado como auxílio a professores que disponibilizam material de suas aulas presenciais na *Web*, para que estes possam obter informações sobre os acessos aos seus materiais.

## 6.1 Trabalhos Futuros

Existem ainda outras funcionalidades, que, se agregadas ao Sistema de Alertas, trariam grandes contribuições, tornando o sistema mais completo. Dentre estas funcionalidades pode-se relacionar:

- A Validação do Sistema de Alertas integrado no ambiente Tapejara. Tendo em vista que o ambiente Tapejara está em fase de implementação, não foi possível integrá-lo totalmente para testes e validação. Uma vez implementado, poderá ser aprimorado o estudo de caso que está sendo realizado na CRT com o uso de todo o ambiente em execução, frente a um curso *on-line*.
- Capturar os eventos gerados por *stored procedures* e *triggers*. São procedimentos que são executados quando alguma informação é modificada na base de dados. Com isso, o sistema poderá reagir às modificações que estão ocorrendo durante a interação do aluno no ambiente, tendo um controle feito em nível de banco de dados.
- Oferecer suporte a formatos de dados comuns e protocolos de mensagem,

como XML e COM.

- Disponibilizar diversos dispositivos de comunicação para escolha do usuário, sendo possível enviar os alertas via *pager*, telefone celular, página *Web* ou *fax*.
- Apresentar saída no formato de XML ou de EDI permitindo o envio a sistemas externos.
- Implementar uma “base de problemas”, em que seriam registrados todos os casos detectados pelo sistema. Com isto, é possível descobrir se o mesmo problema já aconteceu anteriormente e qual solução foi utilizada para corrigí-lo. Desta forma o sistema pode aprender maneiras diferentes para sanar um mesmo problema.

## Anexo 1 Alertas gerados pelo Sistema

### Alertas sobre o curso

TABELA A.1 - Tabela Acesso Mensal a Página

<b>nome:</b>	<b>Acesso Mensal a Página</b>
<b>propósito:</b>	Descobrir páginas que já foram acessadas, mas não apresentam acesso a um mês.
<b>explicação:</b>	É invocado quando a data do último acesso é maior que 30 dias.
<b>condição:</b>	SE ( hoje – ultimoacesso) > 30  ENTÃO AVISOAUTOR( 1)
<b>tipo condição:</b>	tempo
<b>ação:</b>	Enviar e-mail para o responsável pelo material didático avisando da falta de acesso a página.

TABELA A.2 - Falta Acesso Página

<b>nome:</b>	<b>Falta Acesso Página</b>
<b>propósito:</b>	Descobrir páginas que nunca foram acessadas.
<b>explicação:</b>	Checar mensalmente se existe alguma página que nunca foi acessada. Não existe registro de acesso para ela.
<b>condição:</b>	SE NÃO(paginalog)  ENTÃO AVISOAUTOR( 8)
<b>tipo condição:</b>	tempo
<b>ação:</b>	Enviar e-mail para autor do curso avisando que página não foi acessada até a data de hoje.

## Alertas sobre o Aluno

TABELA A.3 – Tabela Primeiro Acesso na Semana

<b>nome:</b>	<b>Primeiro Acesso na Semana</b>
<b>propósito:</b>	descobrir se é o primeiro acesso da semana de um determinado aluno.
<b>explicação:</b>	Verifica se é a primeira vez na semana que o aluno conecta-se ao curso. verificar se os acessos anteriores são da mesma semana.
<b>condição:</b>	SE(hoje – ultimoacesso) >= 7  ENTÃO MSG( 9)
<b>tipo condição:</b>	mensagem KQML tipo 1 – está logado
<b>ação:</b>	Enviar mensagem de Boas Vindas para o aluno.

TABELA A.4 – Tabela Paradinha

<b>nome:</b>	<b>Paradinha</b>
<b>propósito:</b>	Descobrir se o aluno ficou mais de 5 minutos do tempo de leitura estipulado para uma determinada página
<b>explicação:</b>	Verifica se tempo de acesso na página é maior que a velocidade de leitura estabelecido previamente para a página.
<b>condição:</b>	SE (tempoacesso > velocleitu) E  (tempoacesso – velocleitu) <= 5  ENTÃO PERGUNTA( 4)
<b>tipo condição:</b>	mensagem KQML tipo 2 – usuário sai da página
<b>ação:</b>	Enviar pergunta para o aluno

TABELA A.5 – Tabela Paradas Bruscas

<b>nome:</b>	<b>Paradas Bruscas</b>
<b>propósito:</b>	Trocas rápidas de páginas durante uma sequência de acesso.
<b>explicação:</b>	descobrir se o aluno mudou o seu ritmo de estudo, isto é, se estava mantendo um ritmo de acesso e mudou bruscamente ao acessar outra página.
<b>condição:</b>	SE (tempoacesso <1) E (numpagsessão) >=4  E (tempomedioacessosessão) > 4 min  ENTÃO MSG( 12) E ArmazenaParadaBrusca( )
<b>tipo condição:</b>	mensagem KQML tipo 2 – usuário sai da página
<b>ação:</b>	enviar mensagem para o aluno e cadastrá-lo na lista de paradas bruscas, para envio de relatório mensal.

Tabela A.6 – Tabela Paradas Grandes

<b>nome:</b>	<b>Paradas Grandes</b>
<b>propósito:</b>	descobrir se o tempo de permanência na página é maior que o tempo estabelecido para a sua leitura + 10 min.
<b>explicação:</b>	verificar se o tempo de permanência na página ultrapassa 10 min do previamente estabelecido.
<b>condição:</b>	SE (tempoacesso > velocleitu) E  (tempoacesso – velocleitu) <= 10  ENTÃO PERGUNTA( 2)
<b>tipo condição:</b>	mensagem KQML tipo 2 – usuário sai da página
<b>ação:</b>	pergunta ao aluno se ele está com dificuldades. Se resposta for positiva, então enviar página extra de explicação de acordo com o tópico que está sendo estudado.

TABELA A.7 – Tabela Pequeno Abandono

<b>nome:</b>	<b>Pequeno Abandono</b>
<b>propósito:</b>	Descobrir se o aluno “fugiu” do curso a mais de uma semana.
<b>explicação:</b>	verificar se último acesso do aluno no curso é a mais de 7 dias e a menos de 15 dias.
<b>condição:</b>	SE Não(terminoumodulo) E  ((hoje – ultimoacesso) >= 7 E (hoje – ultimoacesso) < 15 )  ENTÃO AVISOALUNO(15)
<b>tipo condição:</b>	tempo
<b>ação:</b>	Enviar e-mail para aluno, incentivando a voltá-lo ao curso.

TABELA A.8 – Tabela Médio Abandono

<b>nome:</b>	<b>Médio Abandono</b>
<b>propósito:</b>	Descobrir se o aluno “fugiu” do curso a mais de quinze dias.
<b>explicação:</b>	Verificar se último acesso do aluno no curso é maior que 15 dias e menor que 30 dias.
<b>condição:</b>	SE Não(terminoumodulo) E  (hoje – ultimoacesso) >= 15 E (hoje – ultimoacesso) < 30  ENTÃO AVISOALUNO(16)
<b>tipo condição:</b>	tempo
<b>ação:</b>	Enviar e-mail para aluno, incentivando a voltá-lo ao curso.

TABELA A.9 - Tabela Grande Abandono

<b>nome:</b>	<b>Grande Abandono</b>
<b>propósito:</b>	Descobrir se o aluno “fugiu” do curso a mais de trinta dias.
<b>explicação:</b>	Verificar se último acesso do aluno no curso é maior que 30 dias e menor que 60 dias.
<b>condição:</b>	SE Não(terminoumodulo) E  (hoje – ultimoacesso) >= 30 E (hoje – ultimoacesso) < 60  ENTÃO AVISOALUNO(16)
<b>tipo condição:</b>	tempo
<b>ação:</b>	Enviar e-mail para aluno, incentivando a voltá-lo ao curso.

TABELA A.10 – Tabela Retorno Rápido ao Curso

<b>nome:</b>	<b>Retorno Rápido ao Curso</b>
<b>propósito:</b>	Descobrir se o aluno voltou ao curso depois de uma semana fora.
<b>explicação:</b>	examinar se o aluno voltou ao curso após o período de 7 a 15 dias do último acesso.
<b>condição:</b>	SE (hoje – ultimoacesso) >=7 E (hoje – ultimoacesso) <15  ENTÃO PERGUNTA(17)
<b>tipo condição:</b>	KQML tipo 1 – usuário loga-se no sistema
<b>ação:</b>	Enviar pergunta na tela para o aluno questionando se deseja uma página de revisão. Se resposta = sim então enviar Página de revisão



TABELA A.11 – Tabela Retorno Médio ao Curso

<b>nome:</b>	<b>Retorno Médio ao Curso</b>
<b>propósito:</b>	descobrir se o aluno voltou ao curso depois de 15 dias fora.
<b>explicação:</b>	examinar se o aluno voltou ao curso após o período de 15 a 30 dias do último acesso.
<b>condição:</b>	SE (hoje – ultimoacesso) >=15 E (hoje – ultimoacesso) <30  ENTÃO PERGUNTA(18)
<b>tipo condição:</b>	mensagem KQML tipo 1 – usuário loga-se no curso
<b>ação:</b>	Enviar pergunta na tela para o aluno questionando se deseja uma página de revisão e exercícios. Se resposta = sim então Página de revisão e Exercícios.

TABELA A.12– Tabela Pulo Exercício

<b>nome:</b>	<b>Pulo Exercício</b>
<b>propósito:</b>	descobrir se o aluno esta acessando os exercícios.
<b>explicação:</b>	Verificar se existe acesso do aluno para a página de exercício para o aluno.
<b>condição:</b>	SE NÃO(exerciciofeito)  ENTÃO MSG(11)
<b>tipo condição:</b>	KQML tipo 3 – usuário sai da sessão
<b>ação:</b>	enviar mensagem para o aluno, avisando da importância de realizar os exercícios

TABELA A.13 – Tabela ExercícioERRO

<b>nome:</b>	<b>ExercícioERRO</b>
<b>propósito:</b>	Descobrir se o aluno não alcançou a porcentagem mínima estabelecida para todos os exercícios
<b>explicação:</b>	quando aluno terminar de realizar todos os exercícios propostos, verificar o seu número de acertos.
<b>Condição:</b>	SE numacertos <= minimoacerto  ENTÃO AVISAAGENTE(agente, mensagem)
<b>tipo condição:</b>	mensagem KQML tipo 4 – término de exercício
<b>ação:</b>	enviar mensagem para o agente Ensino/Avaliação que se encarregar de tomar uma decisão.

TABELA A.14 – Tabela Término Módulo

<b>nome:</b>	<b>Término Módulo</b>
<b>propósito:</b>	verificar se aluno já terminou módulo do curso
<b>explicação:</b>	descobrir se aluno já realizou a avaliação final do curso, terminando o módulo.
<b>condição:</b>	SE nota >= notaexigida  ENTÃO MSG(21) E  AVISOPROF(20)
<b>tipo condição:</b>	mensagem KQML tipo 5 – inserção de nota avaliação
<b>ação:</b>	enviar mensagem para o aluno parabenizando-o e para o professor avisando que o aluno terminou o módulo.

TABELA A.15 – Tabela Aluno Mais Ativo Semana

<b>Nome:</b>	<b>Aluno Mais Ativo Semana</b>
<b>propósito:</b>	descobrir aluno que mais acessou o curso na semana
<b>explicação:</b>	Aluno que contém maior número de acessos a páginas na semana e no mesmo curso
<b>condição:</b>	SE maioracesso  ENTÃO AVISOALUNO(23) E  AVISOPROF(23)
<b>tipo condição:</b>	tempo
<b>ação:</b>	enviar mensagem para o aluno parabenizando-o pelo seu empenho e avisar professor que o aluno está comparecendo ativamente das sessões de estudo.

TABELA A.16 – Tabela Aluno menos Ativo Semana

<b>nome:</b>	<b>Aluno menos Ativo Semana</b>
<b>propósito:</b>	descobrir aluno que contém menor número de acessos
<b>explicação:</b>	Aluno que contém menos número de acessos a páginas na semana ,no mesmo curso
<b>condição:</b>	SE menoracesso  ENTÃO AVISOALUNO(24) E  AVISOPROF(24)
<b>tipo condição:</b>	tempo
<b>ação:</b>	enviar mensagem para o aluno incentivando-o, e avisar professor que o aluno está comparecendo timidamente as sessões de estudo.

TABELA A.17 - Tabela Porcentagem Acesso Tipo Página – baixo

<b>Nome:</b>	<b>Tabela Porcentagem Acesso Tipo Página – baixo</b>
<b>propósito:</b>	descobrir tipo de página que teve o menor acesso na semana
<b>explicação:</b>	verificar o tipo de página menos acessada na semana (exemplo, exercício, conceito)
<b>condição:</b>	SE menorpgacesso  ENTÃO AVISOAUTOR(25)
<b>tipo condição:</b>	tempo
<b>ação:</b>	enviar mensagem para autor do curso, avisando qual tipo de página está com menos acessos.

TABELA A.18 - Tabela Porcentagem Acesso Tipo Página – alto

<b>nome:</b>	<b>Porcentagem Acesso Tipo Página - alto</b>
<b>propósito:</b>	descobrir tipo de página que teve o maior acesso na semana
<b>explicação:</b>	verificar o tipo de página mais acessada na semana (exemplo, exercício, conceito)
<b>condição:</b>	SE maiorpgacesso  ENTÃO AVISOAUTOR(26)
<b>tipo condição:</b>	tempo
<b>ação:</b>	enviar mensagem para autor do curso, avisando qual tipo de página está com mais acessos.

TABELA A.19 - Tabela Mudança Brusca de Nível

<b>Nome:</b>	<b>Mudança Brusca de Nível</b>
<b>propósito:</b>	descobrir se o aluno ficou pouco tempo na página de conteúdo e logo após acessou uma de exercícios .
<b>explicação:</b>	O aluno esta lendo um conteúdo e bruscamente vai para um exercício e começa a acertar todos.
<b>Condição:</b>	SE (ultimaacessada = E) E (antesacessada = C) E (tempoacesso < velocidade/2) E (numacertos >= minimoacertos)  ENTÃO AVISOAGENTE(agente, mensagem)
<b>tipo condição:</b>	mensagem KQML tipo 3 – fim de sessão
<b>ação:</b>	enviar mensagem para o aluno

TABELA A.20 - Tabela RelatórioSemanalCurso

<b>nome:</b>	<b>RelatórioSemanalCurso</b>
<b>propósito:</b>	enviar relatório para professor sobre os acessos ao curso.
<b>explicação:</b>	informações no relatório: páginas mais acessadas, páginas menos acessadas, quantidade de acessos por tipo de página, paradas bruscas, mudanças bruscas de nível
<b>condição:</b>	temporal – SE diasemana = sexta ENTÃO relatorios
<b>ação:</b>	enviar relatório para o professor, autor do curso contendo

TABELA A.21- Tabela RelatórioSemanalAlunos

<b>nome:</b>	<b>RelatórioSemanalAlunos</b>
<b>propósito:</b>	enviar relatório para professor sobre os acessos dos alunos nos cursos.
<b>explicação:</b>	para cada aluno informar: número de acessos, média total de acessos, data último acesso, número de acertos em cada exercício realizado, nota avaliação final, total de retornos simples
<b>condição:</b>	temporal - SE diasemana = sexta ENTÃO relatorios
<b>ação:</b>	enviar relatório para o professor contendo

## Anexo 2 Exemplo de mensagens utilizadas pelo Sistema de Alertas

<b>Mensagens KQML recebidas do mecanismo de rastreamento</b>	
<b>Tipos</b>	<b>Mensagem</b>
1	usuário loga-se no sistema
2	inserção data final de acesso a página
3	fim sessão
4	término de exercício
5	inserção de nota de avaliação

<b>Base de Mensagens</b>	
<b>Mensagens enviadas na tela ou <i>e-mail</i></b>	
1	<Cabeçalho> Analisando os nossos registros, verificamos que a página <informação> não está sendo acessada a mais de um mês.
2	Você está com dificuldades neste conteúdo?
3	A forma de exposição não está adequada?
4	Você está entendendo bem o assunto?
5	<Cabeçalho> O aluno <informação> acha que a forma de exposição da página não está adequada.
6	Desculpe lhe incomodar durante o curso.
7	Sentimos a sua falta!! Faz uma semana que você não compare ao curso.
8	<Cabeçalho> Analisando os nossos registros, verificamos que até hoje, a página <informação> não teve nenhum acesso.
9	Olá! Seja bem Vindo a aula!
10	Que bom! Ficamos contentes por você estar voltando depois de tanto tempo fora.
11	Não esqueça de fazer os exercícios. São importantes para o seu aprendizado!
12	Leia com atenção os conteúdos!

13	Ops! você deixou de visitar o conteúdo <informação> ! Não esqueça de verificá-lo posteriormente!
14	Estamos contente com o seu desempenho. Continue assim.
15	<Cabeçalho> Observando os nossos registros, notamos que você não está comparecendo as atividades do curso. Já estamos com saudades!!!
16	<Cabeçalho> Observando os nossos registros, notamos que você não está comparecendo as atividades do curso a <informação> dias. Volte logo, ainda é tempo de aprender! Faça uma revisão dos conteúdos já vistos em: <informação> .
17	Que bom! Ficamos contentes por você estar voltando depois de tanto tempo fora. Gostaria de uma revisão dos conteúdos já vistos?
18	Que bom! Ficamos contentes por você estar voltando depois de tanto tempo fora. Gostaria de uma revisão e exercícios dos conteúdos já vistos?
19	Parabéns!! Você concluiu o módulo <informação>!
20	<Cabeçalho> O aluno <informação> já concluiu o módulo <informação> do curso <informação>!
21	<Cabeçalho> Parabéns!! você foi o aluno que mais acessou o curso nesta semana! Continue assim.
22	<Cabeçalho> Melhore o seu desempenho!! Você foi o aluno que menos acessou o curso nesta semana!
23	O aluno que participou mais ativamente das atividades do curso nesta semana foi <informação>.
24	O aluno que menos participou das atividades do curso nesta semana foi <informação>.
25	<Cabeçalho> A página <informação> conteve o menor número de acessos na semana!
26	<Cabeçalho> A página <informação> conteve o maior número de acessos na semana!

<Cabeçalho> : Caro(a) <texto> <informação>, onde:

<texto> : autor | professor | coordenador

<informação>: nome do aluno | nome do professor | nome do coordenador do curso| descrição da página | descrição do módulo | descrição do curso.

## Exemplos de mensagens KQML enviadas para Agente Tapejara

<b>Aviso de Erro Exercício</b>		
performative	tell	
sender	AS	
receiver	agente EA	
language	protocolo	
ontology	null	
replay_to	null	
replay_with	null	
force	null	
content	type	1
	operation	1
	user	15
	url	14
	session	2
	time	10
	problem	1
	numacertos	5

<b>Aviso de Parada Brusca</b>		
performative	tell	
sender	AS	
receiver	agente EA	
language	protocolo	
ontology	null	
replay_to	null	
replay_with	null	
force	null	
content	type	2
	operation	2
	user	15
	url	16
	session	2
	time	2
	problem	2
	numacertos	null



<b>Aviso de “Paradinha”</b>		
performative	tell	
sender	AS	
receiver	agente EA	
language	protocolo	
ontology	null	
replay_to	null	
replay_with	null	
force	null	
content	type	1
	operation	2
	user	15
	url	16
	session	4
	time	2
	problem	3
	numacertos	null

<b>Aviso de Mudança Brusca de Nível</b>		
performative	tell	
sender	AS	
receiver	agente EA	
language	protocolo	
ontology	null	
replay_to	null	
replay_with	null	
force	null	
content	type	1
	operation	2
	user	15
	url	14
	session	2
	time	10
	problem	4
	numacertos	null

## Anexo 3 Gramática

[Regra] : :

SE [Expressão] ENTÃO [Ação]

Ação : :

\*Procedimento\* ( [Parâmetro])

[Parâmetro] : :

Nulo

\*Constante\*

[Expressão]

[Expressão] : :

[Expressão Simples]

( [Expressão Simples] = [Expressão Simples] )

( [Expressão Simples] <> [Expressão Simples] )

( [Expressão Simples] > [Expressão Simples] )

( [Expressão Simples] < [Expressão Simples] )

( [Expressão Simples] >= [Expressão Simples] )

( [Expressão Simples] <= [Expressão Simples] )

[Expressão Simples] : :

\*Constante\*

\*Procedimento\* ( [Parâmetro])

( [Expressão])

(NÃO [Expressão])

(+ [Expressão Simples])

( - [Expressão Simples])

( [Expressão Simples] + [Expressão Simples])

( [Expressão Simples] - [Expressão Simples])

( [Expressão Simples] - [Expressão Simples])

( [Expressão Simples] \* [Expressão Simples])

( [Expressão Simples] / [Expressão Simples])

( [Expressão Simples] E [Expressão Simples])

( [Expressão Simples] OU [Expressão Simples])

## Bibliografia

- [AKI 98] AKIKUBO, R. **Sistema de Alerta Computadorizado para Registros Médicos Computadorizados em Cardiologia**. São Paulo: Unicamp, 1998. Dissertação de Mestrado.
- [ALV 97] ALVARES, L.O; SICHAMAN, J.S. Introdução aos Sistemas Multiagentes. In: JORNADA DE ATUALIZAÇÃO EM INFORMÁTICA, 1997. **Anais...** Brasília: UnB, 1997.
- [ARI 99] ARIADNE. 1999. Disponível em: <<http://ariadne.inf.fu-berlin.de:8000>>. Acesso em: 01 dez. 2000.
- [ART 99] ARTEMIS ACTIVE ALERT. **A database monitoring, proactive alerts and trend analysis**. 1999. Disponível em: <<http://www.artemisp.com/Pralert.html>>. Acesso em: 20 jan. 2000.
- [AUL 2000] AULANET. **Ambiente de ensino baseado na Web**. Disponível em: <<http://www.les.inf.puc-rio.br/aulanet>>. Acesso em: 10 out. 2000.
- [AZE 2000] AZEVEDO, L.A.O. **Análise de Atividades de Alunos de Sistemas de Ensino via Internet**. Porto Alegre: PPGC da UFRGS, 2000. Dissertação de Mestrado.
- [BER 97] BERCH, M. **Avaliação Pedagógica como fator para a construção de estratégias de ensino em ambientes de ensino e aprendizagem computadorizados**. Porto Alegre : CPGCC da UFRGS, 1997. (EQ-14).
- [BIC 2000] BICA, F. **Eletrotutor III – Uma Abordagem Multiagente para o Ensino a Distância**. Porto Alegre: PPGC da UFRGS, 2000. Dissertação de Mestrado.
- [BRA 89] BRADSHAW, K.; GARDNER, R.M. ; PRYOR, T. A. Development of a Computerized Laboratory Alerting System. **Computers and Biomedical Research**, [S.l.], v. 22, p. 575-587, 1989.
- [BRA 97] BRADSHAW, J. M. **Software Agents**. [S.l.]: AAAI Press, 1997.
- [BRU 97] BRUSILOVSKY, M. et al. **Addressing Assessment and Adaptativity in Web-based Undergraduate Courses**. 1997. Disponível em: <<http://online.Web.cmu.edu/static/doc/ifip97/ifip97.html>> Acesso em: 08 nov. 1998.
- [BRU 2000] BRUSSO, Marcos José. **Access Miner – Ferramenta de Mineração de dados na descoberta do comportamento do usuário da Web**. Porto Alegre: PPGC da UFRGS, 1999. Dissertação de Mestrado.

- [BUN 79] BUNEMAN, P.; CLEMONS, E. Efficiently Monitoring Relational Databases. **ACM Transactions on Database Systems**, New York, v. 4, n. 3, p. 368-382, 1979
- [BRN 97] BRANDT, S.; KRISTENSEN, A. **Web push as an internet notification service.**  
Disponível em: <<http://keryxsoft.hpl.hp.com/doc/ins.html>>. Acesso em: 01 dez. 2000. Trabalho apresentado no W3C Workshop on Push Technology, 1997.
- [CAR 97] CARBONE, Angela; SCHENDZIELORZ, Peter. **Developing and Integrating a Web-Based Quiz into the Curriculum.** Australian: Monash University, 1997.
- [CAT 99] CATEGORIC SOFTWARE CORPORATION. **The Return on Investment from Enterprise-Wide Alerts.** Disponível em: <<http://www.categoric.com/whitepaper/AlertsROI.htm>>. Acesso em: 15 jan. 2000.
- [CHA 94] CHAKRAVARTHY, S. Snoop: an expressive event specification language for active databases. **Knowledge and Data Engineering Journal**, [S.l.], v.14, p. 1-26, 1994.
- [CHA 82] CHANG, J.; CHANG, S. Database Alerting Techniques for Office Activities Management. **IEEE Transactions on Communications, New York**, v. 30, n. 1, p.74-81, 1982.
- [CHA 87] CHANG, S. ; LEUNG, L. A Knowledge-Based Message Management System. **ACM Transactions on Office Information Systems**, New York, v.5, p. 213-236, 1987.
- [CHO 98] CHOREN, R; BLOIS, M; FUKS, H. QUEST: an Assessment Tool for Web-Based Learning. In: WORLD CONFERENCE OF THE WWW, INTERNET & INTRANET, WEBNET, 1998, Orlando, FL. **Proceedings...** Charlottesville, VA: Association for the Advancement of Computing in Education, 1998.
- [COL 98] COLLA, E. C. Servlets, java do lado do servidor. **Developers Magazine**, June 1998. Disponível em: <[http://www.insite.com.br/docs/develop\\_servlet95.htm](http://www.insite.com.br/docs/develop_servlet95.htm)>. Acesso em: 15 mar. 2001.
- [COM 99] COMPUSCIENCE. Disponível em: <<http://www.zblmath.fiz-karlsruhe.de/cs/CS-cs.html>>. Acesso em: 28 dez. 2000.
- [COS 99] COSTA, M. T. C. **Uma Arquitetura Baseada em Agentes para Suporte ao Ensino a Distância.** Florianópolis: Programa de Pós-Graduação em Engenharia de Produção UFSC, 1999. Tese de Doutorado.

- [CRE 97] CRESPO A; GARCÍA-MOLINA, H. Awareness services for digital libraries. In: RESEARCH AND ADVANCED TECHNOLOGY FOR DIGITAL LIBRARIES, ECDL, 1997, Pisa. **Proceedings...** Berlin: Springer, 1997. p.147-171 (Lecture Notes in Computer Science, v.1324).
- [CRO 88] CRONK, R. N.; CALLAHAN, P. H.; BERNSTEIN. L. Rule-Based Expert Systems for Network Management and Operations: introduction. **IEEE Network**, New York, p.7-21, Sept. 1988.
- [DAN 96] DANDREA, M. **Ferramentas para Segurança na Internet**. Porto Alegre: CPGCC da UFRGS, 1996. (TI-609).
- [DAR 93] THE DARPA KNOWLEDGE SHARING INITIATIVE EXTERNAL INTERFACES WORKING GROUP. **Specification of the KQML Agent- Communication Language**. Baltimore: University of Maryland, 1993.
- [DAV 2001] DAVID'S SERVLET ARCHIVE. **What are servlets?**  
Disponível em <<http://www.javacoffeebreak.com/servlets>>. Acesso em: 21 mar. 2001.
- [DES 98] DESENVOLVIMENTO de cursos online usando WEBCT. Disponível em: <<http://www.ccuec.unicamp.br/treinamentos/Webct/>>. Acesso em: 26 dez. 2000.
- [ELS 99] ELSEVIER Contents direct. Disponível em: <<http://www.elsevier.nl.>>. Acesso em: 20 dez. 2000.
- [ELS 99a] ELSEVIER Science Channel. Disponível em: <<http://www.sciencechannel.nl>>. Acesso em: 20 dez. 2000.
- [FAE 98] FAENSEN, D; HINZE, A; SCHWEPPE, H. Alerting in a Digital Library Environment. Do Channels Meet the Requirements? In: RESEARCH AND ADVANCED TECNOLOGY FOR DIGITAL LIBRARIES, ECDL, 1998. Heraklion, Greek. **Proceedings...** Berlin: Springer-Verlag, 1998. p. 643-644. (Lecture Notes in Computer Science, v. 1513).
- [FEL 98] FELDENS, M. A. ; NARDON, F. B.; CASTILHO, J. M.V. **Sistemas de Suporte à Decisão Baseados em Componentes**. Disponível em: <<http://www.inf.ufrgs.br/~feldens>>. Acesso em: 10 jan. 00.
- [FEL 99] FELDENS, M. A.; MORAES, R. L. **Inteligência Artificial e Inteligência do Negócio: a evolução das tecnologias para suportar decisões bem informadas**. Porto Alegre: PPGC da UFRGS, 1999. Disponível em: <<http://www.inf.ufrgs.br/~feldens>>. Acesso em: 19 jan. 2000.

- [FIN 96] FRITZSON, R. **KQML as an Agent Communication Language**. Baltimore Country: Computer Science Department, University of Maryland, 1996.  
Disponível em: <<http://www.cs.umbc.edu/kqml/papers/kqml-acl-html/root2.html>>. Acesso em: 15 fev. 2000
- [FIO 2000] FIORESE, M. **Uma Proposta de Autenticação de Usuários para Ensino a Distância**. Porto Alegre: PPGC da UFRGS, 2000. Dissertação de Mestrado.
- [FIS 98] FISHER, B.; CONWAY, K.; GROENEBOER, C. **Virtual-U Development Plan: issues and process**. 1998. Disponível em: <<http://virtual-u.cs.sfu.ca/vuWeb/>>. Acesso em: 22 dez. 2000.
- [FUK 98] FUKS, H.; MENEZES, R. A.; GARCIA, A. C. Utilizando agentes no suporte à avaliação informal no ambiente de instrução baseada na Web – AulaNet. In: SBIE, 1998, Fortaleza, CE. **Anais...** Disponível em: <<http://www.lia.ufc.br/sbie98/anais/artigos/art22.html>>. Acesso em: 24 dez. 1999.
- [GEH 92] GEHANI, N.; JAGADISH, H. V. Composite event specification in active databases. In: CONFERENCE ON VERY LARGE DATABASES, 1992. **Proceedings...** Los Altos, CA: Morgan Kaufman, 1992.
- [GOL 97] GOLDBERG, M. W.; SALARI, S. **An Update on WebCT (World-Wide-Web Course Tools): a Tool for the creation of sophisticated Web-based Learning Environments**. Disponível em: <<http://star.ucc.nau.edu/~nauWeb97>>. Acesso em: 15 jan. 2000.
- [HAD 96 ] HADDADI, A. **Communication and Cooperation in agent Systems**. Berlin: Springer-Verlag, 1996. (Lecture Notes in Artificial Intelligence, v.1056).
- [HAC 99] HACK, L. E. **Mecanismos Complementares para a Avaliação do Aluno na Educação a Distância**. Porto Alegre: PPGC da UFRGS, 1999. Dissertação de Mestrado.
- [HAU 94] HAUG, P.J. et al. Decision Support in Medicine: Examples from the HELP System. **Computers and Biomedical Research**, USA , v. 27, p. 396-418, 1994.
- [HAY 97] HAYDT, R. C. **Avaliação do processo Ensino-Aprendizagem**. São Paulo: Ática, 1997.
- [HIN 99] HINZE, A. **Alerting systems for Digital Libraries**. Disponível em: <<http://www.inf.fu-berlin.de/~hinze/projects/alerting/alerting.html>>. Acesso em: 05 dez. 2000.
- [HIN 99a] HINZE, A.; FAENSEN, D. A Unified Model of Internet Scale Alerting

- Services. In: INTERNATIONAL COMPUTER SCIENCE CONFERENCE, 1999, **Proceedings...** Hong Kong: [s.n.], 1999. Disponível em: <<http://www.inf.fu-berlin.de/~hinze/projects/publications>>. Acesso em: 01 dez. 00.
- [HIN 99b] HINZE, A. **Alerting Services in a Digital Library Environment** Doctoral Consortium at CAiSE 1999, Heidelberg published as Technical Report, ETH Zürich. Disponível em: <<http://www.inf.fu-berlin.de/~hinze/projects/publications>>. Acesso em 01 dez. 00.
- [HIN 99c] HINZE, A. **A Model of Alerting Services in Wide Area Networks.** Disponível em: <<http://www.inf.fuerlin.de/~hinze/projects/publications/WSdb99/paper.html>>. Acesso em: 01 dez. 2000.
- [HOR 99] HORVITZ, E.; JACOBS, A.; HOVEL, David. Attention-Sensitive Alerting. In: CONFERENCE ON UNCERTAINTY IN ARTIFICIAL INTELLIGENCE UAI-99, 15. 1999, San Francisco, CA. **Proceedings...** [S.l.]: Morgan Kaufmann, 1999.
- [HÜB 97] HÜBNER, J. F. **Linguagem para Especificação de Protocolos de Comunicação entre agentes.** Blumenau: FURB, 1997. Disponível em: <<http://www.lti.pcs.usp.br/~jomi/projetos/>>. Acesso em: 31 mar. 2001.
- [HYP 98] H YPERLEARNING Meter Brochure. Disponível em: <<http://cne.gmu.edu/hlmeter/hlmbro.html>> Acesso em: 31 mar. 2000.
- [JAQ 99] JAQUES, P. A. **Agentes de Software na Monitoração da Colaboração em Ambientes Telemáticos de Ensino.** Porto Alegre: Curso de Mestrado em Informática, PUCRS, 1999. Dissertação de Mestrado.
- [JAV 99] JAVASOT. Disponível em: <<http://www.javasoft.com/products/servlet>>. Acesso em: 14 mar. 2001.
- [KRI 95] KRISHNAMURTHY B.; ROSENBLUM D. S. Yeast: a general purpose event-action system. **IEEE Transactions on Software Engineering**, [S.l.], v.21, n.10, p. 845-857, Oct.1995.
- [KUP 90] KUPERMAN, G.J., GARDNER, R.M. The Impact of the Help Computer System. In: SYMPOSIUM ON COMPUTER APPLICATIONS IN MEDICAL CARE, SCAMC, 1990, California, USA. **Proceedings...** California, USA: [s.n.], 1990. p. 673-677.
- [LEA 98] LEÃO, B. F. et al. Decision Support Systems for Heathcare: a Methodology Review. In: ANNUAL CONFERENCE ON ELECTRONIC PATIENT RECORDS, 14., 1998, San Antonio, Texas. **Anais...** Texas: [s.n.], 1998.

- [LEN 88] LENAT, D.B. **Knowledge Representation**. USA: Addison-Wesley, USA, 1988.
- [LOT 2000] LOTUS DEVELOPMENT CORPORATION. **Learning Space 4.0**. Disponível em: <<http://www.lotus.com/home.nsf/tabs/learnspace>> Acesso em: 15 dez. 2000.
- [MCC 97] MCCRICKARD, D. S. **Information Awareness on the Desktop: a case study**. Atlanta: Georgia Institute of Technology, 1997. Relatório Técnico.
- [McD 99] McDONALD C. et al. The Regenstrief Medical Record System: a quarter century experience. **International Journal of Medical Informatics**, [S.l.] v. 54, n. 3, p. 225-253, June 1999.
- [MEN 98] MENDES, Marcel. **Conceitos e Significados de Avaliação de Aprendizagem**. Disponível em: <<http://www.mackenzie.br/artig1.htm>>. Acesso em 01 out 2000.
- [MEE 99] MENEGAZZO, C.; ALMEIDA, M. J. B. Apoio ao Gerenciamento de Redes. In: SEMANA ACADÊMICA do PPGC, 4., 1999, Porto Alegre. **Anais...** Porto Alegre: PPGC da UFRGS, 1999. p. 159-162.
- [NAR 97] NARDON, F. **Relatório de Atividades no Projeto SIDI**. Porto Alegre: UFRGS, 1997. Disponível em: <<http://www.inf.ufrgs.br/gpesquisa/bdi/SIDI/sidi-reltrab.html>> . Acesso em: 08 mar. 2001.
- [NED 99] NEDER, M. L. C. **Avaliação na Educação a Distância**. Disponível em: <<http://www.fae.ufmg.br/catedra/Artigo5.htm>>. Acesso em: 20 jun. 1999.
- [NOY 98] NOYA, R. C. RIBEIRO, M. B. FUKS, H. **Quest An Assessment Tool for Web-Based Learning**. Rio de Janeiro: PUC-Rio, 1998. (Monografia em Ciência da Computação n. 005/98).
- [OBA 2000] OBAL, P. Intelligent Business Alerts – Smart, Practical, and Saves Money. **IDII Software Newsletter**. [S.l.], v.1, n. 10, July 2000. Disponível em: <<http://idii.com/esn/vol1num10.htm>>. Acesso em: 14 dez. 2000.
- [OLI 2000] OLIVEIRA, E. H. T.; SCAPIN, R. H.; VICENTINI, W. B. **Proposta de um Ambiente Hipermídia Integrado para o Desenvolvimento de Cursos a Distância**. Disponível em: <<http://www.abed.org.br>>. Acesso em: 28 set. 2000.
- [PER 99] PEREIRA, A.S. **Um Agente para Seleção de Estratégias de Ensino em Ambientes Educacionais na Internet**. Porto Alegre: PPGC da



- UFRGS, 1999. Dissertação de Mestrado.
- [PIN 95] PINTO, Ana Clara. **SEGREDE – Mecanismos para Gerência de Segurança em redes**. Porto Alegre: CPGCC da UFRGS, 1995. Dissertação de Mestrado.
- [REC 97] RENNHACKKAMP M. **Database Event Alerters**. Disponível em: <<http://www.dbmsmag.com/9707d17.html>> Acesso em: 29 jan. 2000.
- [REH 97] REHAK, D. **Carnegie Mellon Online**. 1997. Disponível em: <<http://online.web.cmu.edu/static/doc/overview/overview.html>> Acesso em: 08 nov. 1998.
- [RIT 2000] RITZEL, M. I. **Um mecanismo para Controle de Uso de Material Didático a Distância**. Porto Alegre: PPGC da UFRGS, 1999. Dissertação de Mestrado.
- [ROD 98] RODRIGUES, R. S. **Modelo de Avaliação para cursos no ensino a distância**. Florianópolis: Programa de Pós Graduação em Engenharia de Produção da UFSC, 1998. Dissertação de Mestrado. Disponível em: <<http://www.eps.ufsc.br/disserta98>>. Acesso em: 02 dez. 2000.
- [ROD 2000] RODRIGUES, A. P. **O processo de avaliação de Ensino Aprendizagem em Ensino a Distância**. Porto Alegre: PPGC da UFRGS, 2000.
- [ROS 97] ROSENBLUM D. S. WOLF, A. L. A design framework for internet-scale event observation and notification. In: EUROPEAN SOFTWARE ENGINEERING CONFERENCE, 6., 1997, Berlin. **Proceedings...** Berlin: Springer Verlag, 1997. p.344-360. (LNCS, v.1301).
- [SAB 95] SABBATINI, R.M.E.; ORTIZ, J.; GHEFTER, C.M. Neural Network Based Approach to outcome prognosis for patients with diastolic dysfunction. In: COHEN, M.E.; HUDSON, D. L. (Eds.) **Comparative Approaches in Medical Reasoning**. New York: [s.n.], 1995.
- [SAN 2000] SANTOS, C. T. **Agente de Acompanhamento Pedagógico para Sistemas Tutores Inteligentes**. Santa Cruz do Sul: UNISC, 2000. Trabalho de Conclusão.
- [SIL 97] SILVERMAN B. G. Computer Reminders and Alerts. **IEEE Computer**, Los Alamitos, v.30, n.1, Jan. 1997.
- [SCH 2000] SCHWEPPE, H.; HINZE A.; FAENSEN. D. Event-based Notification on the Web. In: INTERNATIONAL WORLD WIDE WEB CONFERENCE, 9., 2000, Amsterdam. **Proceedings...** Disponível em: <<http://www.inf.fu-berlin.de/~hinze/projects/publications>>. Acesso em: 01 dez. 2000.
- [SHA 2000] SHABOT, M. M; LOBUE, M; CHEN, J. Wireless clinical alerts for

- physiologic, laboratory and medication data. In: AMIA SYMPOSIUM, 2000. **Proceedings...** [S.l : s.n], 2000.
- [SMA 97] COGNITION AND TECNOLOGY GROUP AT VANDERBILT. **A Smart Curriculum for Middle-School Science Instruction: a Web-Based Curriculum Integrating Assessment and Instruction.** Nashville: Vanderbilt University, 1997.
- [SPR 99] SPRINGER Link Alert. Disponível em: <<http://link.springer.de/alert>>. Acesso em: 27 dez. 2000.
- [SWE 96] SWETSSCAN. Disponível em: <<http://www.swets.nl>>. Acesso em: 27 dez. 2000.
- [TAR 96] TAROUCO, L. M. R; NUNES, C. M. Tratamento automatizado de alertas. In: WAIS, 2. 1996, Fortaleza. **Anais...** Fortaleza: UFC, 1996.
- [THE 99] THE LONDON PARALLEL APPLICATIONS CENTRE. **The london & south-east centre for high performance computing.** Disponível em: <<http://www.lpac.ac.uk/SEL-HPC/>>. Acesso em: 13 out. 2000.
- [TIN 96] TINOCO L.C. et al. QUIZIT: an interactive quiz system for WWW-based instruction. In: BRAZILIAN SYMPOSIUM OF COMPUTERS IN EDUCATION, 7., 1996. **Anais...** Brasília: [s.n.], 1996.
- [TOP 2000] TOP CLASS. **Tele-ensino tem novo software de gerenciamento.** Disponível em: <<http://www.novomilenio.inf.br/ano00/0003b005.htm>>. Acesso em: 10 dez. 2000.
- [TUC 95] TUCKER, R. W. Assessing the Virtual Classrooms: a progress report. **Journal Assessment and Accountability Forum**, [S.l.], v. 5, n. 2, 1995.
- [UNI 96] UNIVERSITY of Indiana. 1996. Disponível em: <<http://best.indiana.edu/faq.html>>. Acesso em: 08 nov. 2000.
- [VAS 99] VASCONCELOS, M. M. **Diagnóstico Preliminar para um modelo de avaliação para o ensino a distância.** Pelotas: UCPel, 1999. Projeto de Graduação em Análise de Sistemas.
- [VAV 98] VAVASSORI, F. B.; GAUTHIER, F. Á. O. Proposta de Ferramentas e Agentes Inteligentes para um Ambientes de Ensino/Aprendizagem na Web. In: SBIE, 1998. **Anais...** [S.l : s.n], 1998.
- [YAN 97 ] YANNIS, L; FININ, T. **A proposal for a new KQML specification.** Baltimore: University of Maryland Baltimore County (UMBC), 1997. Relatório Técnico Computer Science and Electrical Engineering Department (CSEE).
- [WBT 98] WBT SYSTEM. **Online Learning Business Solutions.** San Francisco:

[s.n.], 1998

- [WHA 98] WHAT is Virtual-U? 1998. Disponível em: <<http://virtual-u.cs.sfu.ca/vuWeb/vuenglish/what/what.html>>. Acesso em: 31 mar. 2000.
- [WID 94] WIDOM, J; HANSON, E. N. D. Active Database Systems In: KIM, W. (Ed.). **Modern Database Systems**. Massachusetts: Addison-Wesley, 1994. p. 434-456
- [WAT 86] WATERMAN, D. A. **A Guide to Expert System**. USA: Addison-Wesley, 1986.
- [WEI 99] WEISS, G. **Multiagent Systems: a Modern Approach to Distributed Artificial Intelligence**. [S.l.]: The MIT Press, 1999.
- [ZAN 99] ZANATTA, A. L. **Análise de padrões de interação de alunos remotos participantes de atividades colaborativas**. Porto Alegre: PPGC da UFRGS, 1999. (TI-846).