

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

**Javal - Modelo de Ambiente de Avaliação Remota  
Multiagente Baseada em Tutores Embarcados**

por

JOSÉ PEREIRA EMILIANO  
PPGC-UFRGS

Trabalho de conclusão submetido à avaliação,  
como requisito parcial para obtenção  
do grau de Mestre em Informática

Prof. Dr. Claudio Resin Geyer  
Orientador

Porto Alegre, março de 2002.

## CIP - CATALOGAÇÃO NA PUBLICAÇÃO

Emiliano, José Pereira

Javal - Modelo de Ambiente de Avaliação Remota Multiagente Baseada em Tutores Embarcados / por José Pereira Emiliano. - Porto Alegre: PPGC da UFRGS, 2002.

102p.:il.

Trabalho de Conclusão (Mestrado) - Universidade Federal do Rio Grande do Sul. Programa de Pós Graduação em Ciência da Computação, Porto Alegre, BR, 2002. Orientador: Geyer, Claudio Resin.

1. Avaliação remota para EAD. 2. Multiagentes. 3. Ensino à Distância. 4. Tutores embarcados. I. Geyer, Claudio Resin. II. Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora: Profa. Wrana Panizzi

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

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

Coordenadora do PPGC: Profa. Carlos Alberto Heuser

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

## Agradecimentos

Embora acabe esquecendo de agradecer a alguém pelo auxílio direto ou indireto na confecção deste trabalho, seguem os que pude me recordar. Àqueles que porventura tenha esquecido de mencionar aqui, também vai meu agradecimento e minhas desculpas.

Ao Pai:

Agradeço a Deus, Mestre supremo e Senhor soberano, que nos concedeu o Dom da procura pelo saber, sem ela a humanidade ainda estaria caçando animais com paus e pedras.

Aos Pais:

Agradeço e dedico este trabalho à minha amada mãe Luzinete Emiliano, que com seu exemplo de dedicação, garra e determinação, mostrou-me que tudo é possível na vida. Querer é poder.

Sem seu auxílio e exemplo, não sei se hoje estaria onde me encontro.

Meu pai, que Deus o tenha em bom lugar e que, de onde estiver, esteja orgulhoso de mim pelo que eu sou e pelo que eu tenho feito como pessoa e como profissional.

A minha família:

Agradeço à Fabiana, minha querida e dedicada esposa, que tem acompanhado os meus trabalhos incansáveis aos fins de semana e nas noites, diariamente, e que, no meu cansaço, tem estado sempre pronta a me acolher e confortar.

Sem ela, certamente, meu trabalho teria sido muito mais difícil.

Aos Mestres:

Agradeço à professora Taisy Weber pela paciência e apoio a mim dispensado e por ter escutado por diversas vezes minhas solicitações, o que me rendeu bastante motivação e interesse.

Agradeço ao professor Geyer, pela paciência que teve e pelas sugestões oportunas que me conduziram à elaboração deste trabalho.

Aos Chefes:

Agradeço aos meus Chefes, que confiaram na minha capacidade e investiram para que fosse possível a realização deste curso.

À Capes :

Agradeço pelo apoio prestado à meu projeto através do Programa de Apoio a Pesquisa em Educação à Distância (PAPED), sem o qual teria sido bastante difícil sua elaboração e divulgação.

Aos Colegas e Amigos:

Agradeço também aos amigos de Mestrado, novas e duradouras amizades conquistadas nas listas de discussão e nos encontros nos intervalos de aula, o que, certamente, deixará saudades.

<b>Agradecimentos.....</b>	<b>3</b>
<b>Abstract.....</b>	<b>12</b>
<b>1 Introdução.....</b>	<b>13</b>
1.1 Tema.....	13
1.2 Motivação.....	13
1.3 Objetivos .....	14
1.4 Trabalhos relacionados.....	15
1.5 Contribuição do Autor.....	15
1.6 Estrutura da Dissertação.....	16
<b>2 Sistemas de avaliação remota em ambientes de Ensino à Distância.....</b>	<b>18</b>
2.1 Question Mark Perception.....	18
2.1.1 Elaboração e armazenamento de questões.....	19
2.1.1.1 Estrutura da avaliação.....	21
2.1.2 Estrutura de sessões e avaliações.....	21
2.1.3 Entrega das Questões.....	22
2.1.4 Relatórios .....	23
2.2 Web CT.....	23
2.3 AulaNet .....	25
2.4 AvalWeb.....	26
2.5 Resumo das características das ferramentas analisadas.....	26
2.6 Resumo do capítulo dois.....	28
<b>3 Considerações sobre o processo ensino-aprendizagem aplicado ao ambiente de Ensino à Distância.....</b>	<b>29</b>
3.1 Considerações iniciais.....	29
3.1.1 Os problemas da avaliação remota e o processo ensino-aprendizagem.....	31
3.1.2 O papel do professor na avaliação remota empregando agentes.....	34
3.2 Uma proposta de solução computacional para os problemas.....	35
3.3 O desenvolvimento de agentes.....	37
3.4 Os agentes do ambiente Javal.....	38
3.4.1 O agente tutor.....	40
3.4.2 O agente monitor.....	42
3.5 Os agentes do ambiente de avaliação Javal.....	43
3.6 Modelo de Objetos Distribuídos como base para avaliação em EAD e CWCT.....	43

<b>3.7 Resumo do capítulo três.....</b>	<b>44</b>
<b>4 O modelo funcional do ambiente Javal.....</b>	<b>45</b>
<b>4.1 Composição do ambiente.....</b>	<b>46</b>
<b>4.1.1 Módulo de Controle de Avaliações (MCA).....</b>	<b>50</b>
<b>4.1.2 Módulo de Avaliação Remota (MAR).....</b>	<b>51</b>
4.1.2.1 Repositório de Objetos Locais (ROL).....	52
4.1.2.2 Unidade de Suporte de Comunicação (USC).....	52
4.1.2.3 Unidade de Controle de Objetos (UCO) .....	53
4.1.2.4 Unidade de Controle do Módulo (UCM).....	53
<b>4.1.3 Módulo de Montagem e Acompanhamento de Avaliações (MMA).....</b>	<b>54</b>
<b>4.1.4 Módulo de Armazenamento de Dados (MAD).....</b>	<b>55</b>
<b>4.2 Descrição funcional do Módulo de Avaliação Remota (MAR).....</b>	<b>55</b>
<b>4.2.1 Carga do ambiente e ajuste.....</b>	<b>57</b>
<b>4.2.2 Execução da avaliação.....</b>	<b>59</b>
4.2.2.1 Fase de seleção.....	59
4.2.2.2 Fase de solução de item.....	60
<b>4.2.3 Avaliação do tempo de execução.....</b>	<b>62</b>
4.2.3.1 Agente de tempo da avaliação global.....	62
4.2.3.2 Agente tempo da avaliação local .....	62
4.2.3.3 Interação entre agentes e o sistema.....	63
<b>4.2.4 Encerramento da avaliação.....</b>	<b>64</b>
<b>4.3 Resumo do capítulo quatro.....</b>	<b>66</b>
<b>5 A implementação do MAR.....</b>	<b>67</b>
<b>5.1 Objetos do ambiente de avaliação.....</b>	<b>68</b>
<b>5.2 Objetos de suporte e controle.....</b>	<b>69</b>
<b>5.3 A API de classes de Javal.....</b>	<b>69</b>
<b>5.4 API do ambiente de avaliação.....</b>	<b>70</b>
<b>5.4.1 O pacote Jquest.....</b>	<b>70</b>
5.4.1.1 Classe Avaliacao.....	71
5.4.1.2 Classe Questao.....	71
5.4.1.3 Classe Item.....	72
<b>5.4.2 Recursos.....</b>	<b>72</b>
<b>5.4.3 O pacote Jstudent.....</b>	<b>73</b>
5.4.3.1 Classe FrameItem.....	74
5.4.3.2 Classe frameVerFalso.....	75
5.4.3.3 Classe frameMarque.....	75
5.4.3.4 Classe frameOrdenacao.....	76
5.4.3.5 Classe frameCorrespondencia.....	76
5.4.3.6 Classe frameResponda.....	77
5.4.3.7 Classe frameCompletamento.....	77
<b>5.4.4 O pacote Jagent.....</b>	<b>78</b>
5.4.4.1 Classe AgenteComportamento.....	80
5.4.4.2 Classe AgenteTempo.....	80
<b>5.5 Interação entre elementos do ambiente.....</b>	<b>80</b>

<b>5.5.1 Mensagens entre agentes.....</b>	<b>81</b>
<b>5.5.2 Interações aluno-ambiente.....</b>	<b>82</b>
<b>5.5.3 Mensagens ambiente-aluno.....</b>	<b>83</b>
<b>5.6 API de objetos de suporte e controle.....</b>	<b>83</b>
<b>5.6.1 Classe unidadeControle.....</b>	<b>84</b>
<b>5.6.2 Classe appletAvaliacao.....</b>	<b>84</b>
<b>5.6.3 Classe frameQuestao.....</b>	<b>85</b>
<b>5.7 Aspectos de implementação do ambiente Javal.....</b>	<b>85</b>
<b>5.8 Resumo do capítulo cinco.....</b>	<b>86</b>
<b>6 Testes realizados e resultados obtidos.....</b>	<b>87</b>
<b>6.1 Vantagens do modelo proposto.....</b>	<b>89</b>
<b>6.2 Melhorias no modelo proposto.....</b>	<b>90</b>
<b>7 Conclusão.....</b>	<b>91</b>
<b>7.1 Novos conhecimentos .....</b>	<b>91</b>
<b>7.2 Trabalhos futuros.....</b>	<b>92</b>
<b>Bibliografia.....</b>	<b>93</b>

## Lista de Figuras

FIGURA 2.1	Esquema geral do ambiente Perception.....	20
FIGURA 2.2	Aspecto do assistente para elaboração de questões do Perception.....	21
FIGURA 2.3	Vista do Question Manager.....	22
FIGURA 2.4	Editor de questões do Question Manager.....	22
FIGURA 2.5	O ambiente Session Manager.....	23
FIGURA 2.6	Tela da ferramenta de administração de perfil de aluno, do WebCT.....	26
FIGURA 3.1	Um aluno e o sensoramento possível para um ambiente informatizado.....	34
FIGURA 3.2	Relacionamento entre agentes do Módulo de Avaliação Remota (MAR).....	38
FIGURA 3.3	A dinâmica de funcionamento da avaliação empregando agentes.....	39
FIGURA 3.4	Um exemplo de um sistema de dicas.....	40
FIGURA 3.5	Uma interface gráfica de agente tutor especialista em tempo.....	42
FIGURA 3.6	Uma interface gráfica de agente tutor especialista em evento.....	43
FIGURA 4.1	O ambiente proposto pelo modelo Javal.....	48
FIGURA 4.2	Interações do sistema ambiente-professor.....	49
FIGURA 4.3	Interações do sistema ambiente-aluno.....	51
FIGURA 4.4	Descrição em blocos do Módulo de Avaliação Remota (MAR), empregado pelo Javal	53
FIGURA 4.5	A estrutura de Repositório de Objetos Locais (ROL).....	54
FIGURA 4.6	Aspecto da Interface de comunicação de grupo (ICG).....	55
FIGURA 4.7	Fluxograma de funcionamento do MAR.....	59
FIGURA 4.8	Diagrama de estados de funcionamento do MAR.....	59
FIGURA 4.9	Aspecto da interface gráfica do Módulo de Avaliação Remota (MAR).....	60
FIGURA 4.10	Interface gráfica do Navegador de Questões.....	62
FIGURA 4.11	Exemplo de eventos e mensagens trocadas entre objetos durante a avaliação.....	63
FIGURA 4.12	Estados do Agente Tempo.....	65
FIGURA 4.13	Comunicação entre a interface gráfica do item e o agente tempo.....	67
FIGURA 5.1	Uma descrição conceitual da estrutura de avaliação adotada pelo modelo.....	70
FIGURA 5.2	Os pacotes da API do ambiente Javal.....	73
FIGURA 5.3	Modelo de objetos do ambiente Jquest.....	74
FIGURA 5.4	Modelo de objetos do ambiente Jstudent.....	77
FIGURA 5.5	Interface gráfica do ambiente Verdadeiro-Falso.....	78
FIGURA 5.6	Interface gráfica do ambiente Marque.....	79
FIGURA 5.7	Interface gráfica do ambiente Ordenação.....	79
FIGURA 5.8	Interface gráfica do ambiente Correspondência .....	80
FIGURA 5.9	Interface gráfica do ambiente Responda.....	80
FIGURA 5.10	Interface gráfica do ambiente Completamento.....	81
FIGURA 5.11	Diagrama de Classes da pacote Jagent.....	82
FIGURA 5.12	Interface gráfica de um agente.....	82
FIGURA 5.13	Mensagens entre objetos componentes do ambiente MAR.....	84
FIGURA 5.14	O ambiente gráfico da Applet Avaliação.....	88
FIGURA 5.15	O ambiente gráfico do Navegador de Questões.....	88

## Lista de Tabelas

TABELA 2.1	Comparação entre as ferramentas de avaliação para EAD comerciais.....	28
TABELA 4.1	Eventos da fase de carga e ajuste.....	61
TABELA 4.2	Eventos ocorridos na fase de execução da avaliação.....	63
TABELA 4.3	Códigos de controle do campo STAT do relatório.....	68
TABELA 4.4	Códigos de controle do campo OBS do relatório.....	68
TABELA 4.5	Um exemplo de relatório de questão realizada por aluno no MAR.....	69
TABELA 5.1	Principais interações entre agentes do MAR.....	85
TABELA 5.2	Principais interações entre o aluno e o ambiente.....	85
TABELA 5.3	Principais interações entre o ambiente e o aluno.....	86
TABELA 6.1	Formato da avaliação aplicada como teste.....	91
TABELA 6.2	Mapeamento de resultados obtidos.....	91
TABELA 6.3	Mapa de resultados da avaliação.....	92
TABELA 6.4	Vantagens do modelo proposto por Javal.....	93



## Lista de Siglas

AAP	Agente Ajusta Perfil
ACA	Agente Comportamento de Avaliação
ATA	Agente Tempo de Avaliação
ALN	Asynchronous Learning Network
ASP	Active Server Pages
ASR	Agente Seleção de Recurso
ATA	Agente Tempo
ATG	Agente Tempo Global
CMA	Console de Monitoração de Avaliação
COD	Conversor Objeto/banco de dados
CORBA	Common Object Request Broker Architecture
CSCW	Computer-Supported Collaborative Work
EAD	Ensino a Distância
GUI	Graphical User Interface
HTML	Hypertext Markup Language
ICG	Interfície de Comunicação de Grupo
IIS	Internet Information Server
LES	Laboratório de Engenharia de Software
MAD	Módulo de Armazenamento de Dados
MAR	Módulo de Avaliação Remota
MCA	Módulo de Controle de Avaliações
MMA	Módulo de Montagem de Avaliações
MOA	Montador de Objetos da Avaliação
OLE	On line Learning Environment
ODBC	Open Database Connectivity
PHP	Personal Home Page Tools
PUC	Pontificia Universidade Catolica
PWS	Personal Web Server
QML	Question Markup Language
SQL	Structured Query Language
RAT	Remote Attendant
RMI	Remote Method Invocation
ROL	Repositorio de Objetos Locais
SCG	Suporte de Comunicação de Grupo
SEAD	Sistema de Ensino a Distancia
SEMEAI	Sistema Multiagente de Ensino e Aprendizagem na Internet
SSC	Serviço de Suporte à Comunicação
UCG	Unidade de Controle de Grupo

UCA	Unidade de Controle do Ambiente
UCM	Unidade de Controle do Módulo
UCO	Unidade de Controle de Objetos
USC	Unidade de Suporte a Comunicação
Web CT	Web Course Training

## Resumo

Trata o presente objeto de pesquisa da proposta de desenvolvimento de um modelo de ambiente de avaliação remota para Ensino à Distância, baseado no paradigma de Orientação a Objetos e elaborado com base na tecnologia de sistemas multiagentes.

Para a validação do modelo, foi desenvolvido um protótipo denominado Javal, capaz de permitir a aplicação e monitoração da realização de avaliações e testes à distância.

As soluções comerciais disponíveis no mercado, como Question Mark Perception, Aula Net e WebCT possuem código fechado e custo elevado, além de serem baseadas em soluções proprietárias (QML - Question Mark, ASP - Microsoft, etc.), necessitando de plataforma específica para instalação de servidores e clientes.

Quanto ao aspecto funcional, estas ferramentas possuem a base de sua lógica de funcionamento em execução de scripts no servidor. Para cada item de uma avaliação a ser respondido, torna-se necessária a efetivação de uma nova conexão.

O modelo proposto traz toda a funcionalidade do sistema para o ambiente do aluno, tornando necessária apenas uma conexão para a busca da avaliação e outra para o envio dos resultados, reduzindo o tráfego na rede.

Quanto ao aspecto pedagógico, estas ferramentas limitam-se apenas a apresentar uma avaliação somativa do aluno, geração de graus e estatísticas, não se preocupando em monitorar seus aspectos comportamentais, capazes de apontar indícios de possíveis falhas no aprendizado ou na elaboração da avaliação.

A proposta do modelo é da apresentação de um ambiente orientado a objetos, capaz de distribuir elementos representativos das avaliações existentes no modelo tradicional de ensino, incorporando recursos capazes de possibilitar a monitoração de aspectos comportamentais do aluno, pelo emprego de agentes monitores ou tutores, que podem acompanhar o aluno e auxiliá-lo em situações de dificuldade.

O modelo proposto por este trabalho envolve as avaliações formativas e somativas, aplicadas de forma assíncrona e individual. Como sugestão para trabalhos futuros, o modelo propõe o desenvolvimento de classes capazes de implementar a aplicação de avaliações síncronas e em grupo.

A validação do modelo proposto foi realizado através do desenvolvimento de um protótipo que, com base no desenvolvimento de uma API Javal específica, implementa os principais tipos de questões disponíveis no sistema de ensino tradicional, além de agentes tutores de avaliação.

**Palavras-Chave:** Avaliação remota, Ensino à Distância, Sistemas Multiagentes, Tutores, Java.

**TITLE:** “JAVAL - MULTIAGENT-BASED REMOTE EVALUATION ENVIRONMENT”.

## **Abstract**

This work focuses on the development of an environment model for remote evaluations in Remote Teaching, based on the Object-Oriented paradigm and multiagent system approach.

For the validation of this model, it was developed using the Java language a tool named Javal, which is designed to help a teacher to build, distribute, apply and monitor remote evaluations.

The tools available in the Internet, like Question Mark Perception, Aula Net, WebCT, among others, are usually expensive and black-box type (or closed code), as long as they use proprietary solutions (like QML, ASP, etc.), needing specific platforms for the set up of servers and clients.

Concerning the functionality, these tools have its business logic running from the server. For each item of an evaluation to be answered, a new connection to the server is necessary, increasing the traffic in the net.

Focusing pedagogical aspects, these tools are limited to only present an exam of the student and generate statistics, neglecting the feedback from the student behavior, which could be capable of pointing out possible mistakes in the educational process.

These tools do not implement or support collective communication, useful in the development of evaluations in collaborative work.

Hence, in the present proposal the concepts of collaborative environments and of auxiliary agents in the evaluation process are present by the implementation of auxiliary didactic classes named specialist agents or tutors.

The model proposed in this work, involves both the formative and summative evaluations, in an asynchronous and individual way. As suggestion for future works, it is proposed the development of classes capable of implementing the application of synchronous or group evaluations.

The validation of the proposed model was possible due to a prototype tool, built using a specific API (Javal), which implements the main types of questions found in the traditional educational system as well as evaluation tutor agents.

Functionally, Javal has its logic based on the client, what brings up many advantages, such as reduction of the net traffic, multitask support usage and possibility of pedagogic agents inclusion.

**Keywords:** Remote evaluation, Remote Teaching, Multiagent Systems, Tutors, Java.

# 1 Introdução

A *Internet*, através de seus mais variados recursos, tem possibilitado a comunicação de seus usuários a longas distâncias, unificando diferentes plataformas e disponibilizando diversas informações e serviços. Gradativamente, tornou-se um meio usual de troca de informações de forma rápida, rompendo barreiras geográficas de espaço e tempo, e permitindo o compartilhamento de informações, cooperação e comunicação em tempo real.

As propostas do uso desta tecnologia para fins educacionais possibilita alternativas no que concerne às áreas de pesquisa. Entre essas áreas encontram-se as aplicações de hipermídia, os sistemas de autoria para cursos à distância, os sistemas de aprendizado à distância, os ambientes de comunicação e colaboração para fins educacionais, *frameworks* para aprendizagem cooperativa e os ambientes distribuídos de ensino-aprendizagem [PER98][PER99][ROD00].

Soluções disponíveis baseadas em OLE (*On Line Learning Environment*) e ALN (*Asynchronous Learning Network*) oferecem propostas para os principais problemas relacionados com o aprendizado em qualquer hora ou lugar, segundo Dringus [DRI99], de onde pode-se observar a necessidade de desenvolvimento de uma ferramenta de ensino que permita integração e colaboração entre os participantes, principalmente durante um processo de avaliação remota.

## 1.1 Tema

Este trabalho tem como finalidade apresentar um modelo para desenvolvimento de uma ferramenta para ambientes de Ensino à Distância, capaz de distribuir avaliações e permitir o acompanhamento de sua realização pelo emprego de agentes especialistas denominados tutores embarcados .

Tal ambiente visa não apenas a avaliação somativa, mas o levantamento de informações que permitam uma avaliação formativa, através da monitoração do comportamento do aluno ante cada questão apresentada.

Interações do aluno com o sistema são registradas, possibilitando identificar comportamentos que indiquem ao professor a necessidade de efetuar mudanças na estratégia de ensino adotada ou nos recursos destinados à avaliação do aluno, adequando-o ao seu tipo de perfil .

Para o desenvolvimento do protótipo, serão empregados o modelo de Orientação a Objetos e a linguagem Java como suporte, graças a suas características de segurança, portabilidade, modularidade e suporte à multitarefa.

## 1.2 Motivação

A avaliação do processo ensino-aprendizagem realizado à distância, em seu enfoque baseado em agentes [ROD00] e cuja proposta encontra-se inserida no projeto SEMEAI [GEY01], da Universidade Federal do Rio Grande do Sul, pretende integrar a avaliação formal e informal, procurando identificar os problemas de aprendizagem ocorridos durante a interação dos alunos com o ambiente de Ensino à Distância.

As ferramentas comerciais que permitem a execução de avaliações à distância, como o Perception [QMP99], AulaNet[EDU00] e Web CT[GOL98], além de estarem baseadas em soluções proprietárias, como o QML - *Question Markup Language* [QMP99]- não consideram fatores comportamentais em suas avaliações.

Nestes ambientes o professor não consegue monitorar o comportamento dos alunos durante a execução das provas, uma vez que o sistema baseia-se na geração dinâmica de páginas HTML, que oferecem limitados recursos para a monitoração de eventos de interface gráfica. Isto impede o professor de obter um *feedback* sobre a elaboração da avaliação, requerendo o emprego de recursos específicos como pesquisa de opinião ou uma análise de resultados.

Outro aspecto ligado a estes ambientes refere-se ao sigilo das questões. As páginas das questões, geradas de forma dinâmica, permitem ao aluno seu armazenamento local, graças a recursos existentes nos navegadores (*browsers*), o que pode não ser desejável para o professor, uma vez que, para cada nova turma, tornar-se-à necessária a criação de novas questões versando sobre o mesmo assunto ou tema.

As soluções existentes no mercado não implementam tratamento de falhas durante a execução das questões pelo aluno e, no caso de dúvidas ou comportamento inadequado ante um problema ou até mesmo a apresentação do enunciado de forma inadequada, não existe um elemento específico para realizar a correção instantânea da aprendizagem ou auxiliar o aluno em tempo real, servindo-lhe de apoio antes de vir a ser acionado o professor .

Quanto à funcionalidade, estas ferramentas possuem sua lógica baseada no servidor. Para que o aluno possa executar uma avaliação, torna-se necessário conectar-se ao mesmo (por linha discada ou através de uma rede local) e executar transferências de dados tantas vezes quantas forem as questões disponibilizadas. Isto pode dificultar a aplicação de avaliações que utilizem tempo definido para sua execução, pois, a velocidade de conexão, o tráfego ou congestionamentos de rede podem tornar esta transferência uma atividade excessivamente demorada.

Devido a estas limitações, estas ferramentas não possibilitam avaliações com tempo de execução determinado, limitando o controle do professor sobre o ambiente do aluno.

Com base nestas idéias, e, compreendendo a necessidade de uma ferramenta capaz de simular a presença do professor junto ao aluno, tornou-se interessante o desenvolvimento do presente trabalho, visando a consecução dos objetivos a seguir.

### 1.3 Objetivos

Observando o enfoque dos ambientes para avaliação à distância, o presente trabalho tem os seguintes objetivos :

- Apresentar um modelo baseado no paradigma da orientação a objetos, capaz de permitir o desenvolvimento de ferramentas necessárias para a implementação de ambientes de avaliação remota, incorporando modularidade, portabilidade e escalabilidade;
- Apresentar um modelo de avaliação que incorpore agentes auxiliares pedagógicos embarcados, adicionando funcionalidade ao ambiente de avaliação, em vez de deixá-lo como um painel de exibição de questões;
- Permitir maior sigilo na elaboração de avaliações uma vez que o recurso de applets não permite a gravação de objetos no navegador, dificultando a cópia e armazenamento de avaliações no ambiente do aluno;
- Apresentar a linguagem Java como plataforma viável para o

desenvolvimento de aplicações voltadas para a criação de sistemas de avaliação em ambiente de Ensino à Distância (EAD), como alternativa para plataformas atualmente empregadas por grandes empresas ;

- Aplicar os conceitos de tutores descritos em [FRA97] e apresentar sugestões de implementação de agentes descritos em [PER99]; e
- Contribuir na elaboração de uma ferramenta possível de integração ao projeto SEMEAI (UFRGS) [GEY01], em seu módulo de Avaliação.

Para a validação do modelo, foram realizados testes no sentido de:

- Desenvolver objetos avaliações capazes de incorporar as questões do ambiente de ensino tradicional;
- Permitir ao ambiente aluno identificar e carregar os objetos avaliações e suas questões; e
- Permitir aos agentes tutores especialistas o acompanhamento do aluno, efetuando a correção da avaliação e alertando ao sistema, quando ocorrerem situações de dúvida ou erro.

Foram realizados estudos em ambiente distribuído Java, de forma a permitir :

- Serialização, envio e desserialização das avaliações;
- Comunicação entre os agentes tutores ;
- Recebimento do relatório de execução de prova pelo professor; e
- Recebimento de avaliações e elaboração de resultados.

Para a consecução destes objetivos foram empregados como ferramentas de desenvolvimento:

- Bluette 1.1 e JBuilder 2.0 - para desenvolvimento das GUI;
- Kawa 4.1 e Plastic 1.1 - para desenvolvimento e implementação das classes Java e RMI; e
- Xitami 1.0 - Como servidor web, para suporte do ambiente Javal.

A fase de prototipação e teste teve como objetivo sua integração ao projeto SEAD, proposto pelo autor e em desenvolvimento pelo 1o Centro de Telemática e FEEVALE para uso no Exército, e ao projeto SEMEAI [GEY01].

## 1.4 Trabalhos relacionados

Com relação a descrição de um ambiente virtual, cooperativo e adaptativo de ensino foram consultados os artigos de [HIL98] e [TUR95], onde é enfatizada a importância do ensino colaborativo e on-line (ALN), sobretudo do aspecto da correção on-line e o incremento do processo ensino-aprendizagem.

Sobre propostas para sistemas multiagentes de ensino, foram consultados [ROD00], [REI00] e [PER99] que tratam basicamente da modelagem de agentes nestes ambientes, sobretudo agente avaliação, tomado por base para a elaboração do agente tutor.

## 1.5 Contribuição do Autor

Nos dias atuais, em qualquer ambiente de ensino, é desejo das Instituições de Ensino optar por ferramentas que sejam de livre distribuição e de código aberto, para que possam ser adaptadas conforme a necessidade dos usuários, conforme [DOU96]. Softwares como o StarOffice, e o Direto, elaborado pela PROCERGS, são nítidas demonstrações de uma tendência atual de mercado.

Além deste enfoque, as soluções atuais para avaliação em ambiente Web não são desenvolvidas de forma a suportar um ambiente colaborativo, necessário segundo [HIL98] para atividades em grupo, bem como não oferecem suporte satisfatório ao gerenciamento da distribuição das questões em ambiente Web. O ambiente proposto oferece suporte a classes capazes de prover um ambiente de avaliação colaborativo, em trabalhos posteriores.

Finalmente, os sistemas computacionais de avaliação à distância desenvolvidos até a elaboração deste trabalho não trazem consigo as idéias de agentes ou tutores embarcados que permitam monitorar comportamentos, identificar e tratar as falhas durante a fase de avaliação do processo ensino-aprendizagem. Adicionalmente, o modelo proposto oferece ao professor a obtenção de *feedback* capaz de permitir sua auto-avaliação, quanto a elaboração e aplicação de testes à distância.

Com base nestas idéias, espera o autor deste trabalho prestar sua colaboração no sentido de:

- Flexibilizar a criação, distribuição e acompanhamento de trabalhos e avaliações para os alunos, sobretudo em ambiente Web;
- Desenvolver um framework que suporte um ambiente colaborativo, visando a execução de trabalhos em grupo;
- Empregar a linguagem Java como plataforma para desenvolvimento de aplicações neste campo, tendo por base sua neutralidade de plataforma, suporte à multitarefa e portabilidade de código, garantindo abertura, modularidade e flexibilidade para expansões ou personalizações do sistema;
- Introduzir o conceito de agentes e tutores embarcados junto a avaliações, visando tratar situações de erro ou falha durante a realização de provas; e
- Oferecer uma alternativa para soluções de alto custo e proprietárias através da criação de ferramentas de código-aberto.

## 1.6 Estrutura da Dissertação

Após a introdução, este trabalho apresenta a seguinte estrutura:

- Capítulo 2 - Será feita uma breve introdução sobre sistemas de avaliação para ambientes de Ensino à Distância, realçando as principais características e deficiências encontradas nos sistemas comerciais existentes atualmente.
- Capítulo 3 - Neste capítulo será apresentada uma breve descrição das idéias que nortearam o desenvolvimento do ambiente Javal. Serão inicialmente apresentadas considerações sobre a avaliação do processo ensino-aprendizagem, especialmente em ambientes de Ensino à Distância. Ainda neste capítulo será apresentado o modelo de objetos distribuídos bem como algumas características relevantes das principais ferramentas para



desenvolvimento de aplicações distribuídas orientadas a objetos

- Capítulo 4 - Neste capítulo, será apresentada uma descrição geral do ambiente de avaliação Javal, abordando seus principais módulos.
- Capítulo 5 - Será feita uma descrição detalhada da implementação do modelo funcional do Módulo de Avaliação Remota (MAR), apresentado suas classes componentes e principais funcionalidades .
- Capítulo 6 - Expõe as conclusões alcançadas com o desenvolvimento deste trabalho e possíveis extensões do mesmo, apresentados como trabalhos futuros.

## 2 Sistemas de avaliação remota em ambientes de Ensino à Distância

Os ambientes de Ensino à Distância (EAD) foram desenvolvidos com o intuito de ultrapassar os limites da sala de aula, tornando o aprendizado dinâmico e assíncrono tanto espacial quanto temporalmente. A tecnologia de informação disponível atualmente tem permitido o emprego de redes de computadores como suporte para estes ambientes.

Inúmeras pesquisas foram desenvolvidas de forma a criar ambientes que permitissem ao professor transmitir informações ao aluno, quer de forma passiva (disponibilizando documentos para transferência via rede) ou de forma interativa (como os sistemas baseados na tecnologia de agentes e tutores inteligentes).

Sob o impulso das idéias que nortearam a criação destes ambientes, foram desenvolvidas ferramentas capazes de permitir a realização de avaliações à distância, necessárias para a obtenção da avaliação do ensino e posterior correção da aprendizagem.

Estas ferramentas são compostas por um sistema de armazenamento de questões (banco de dados de questões) que interage com um navegador (*browser*) empregado para consultar documentos de hipertexto (HTML), estes gerados de forma dinâmica por ferramentas como ASP (*Active Server Pages*), QML (*Question Markup Language*) ou PHP (*Personal Home Page Tools*) localizados no servidor.

Ao interagir com páginas geradas pelo ambiente (que reproduzem os diversos tipos de questão existentes no ambiente de avaliação tradicional), o aluno pode ser avaliado de forma somativa, pela obtenção de estatísticas de acertos e erros.

Neste capítulo será realizada uma breve abordagem das ferramentas de avaliação disponíveis em ambientes de Ensino à Distância na Web, buscando realçar e identificar suas principais características, diferenças e deficiências.

### 2.1 Question Mark Perception

A ferramenta Perception [QMP99], da empresa Question Mark, é um ambiente de desenvolvimento de questões que possui suporte a multimídia, podendo ser inseridas dentro de um contexto de uma aula (como uma avaliação durante um módulo de ensino) ou como uma bateria de testes. Dispõe de um servidor (*Perception Server*), que permite a distribuição de testes e avaliações em ambiente Web ou outro ambiente de rede.

A plataforma de *hardware* e *software* requerida envolve o sistema operacional Windows (95, 98 e NT), sendo necessário o emprego de um navegador Internet Explorer versão 4.0 ou superior (embora o fabricante afirme que o sistema trabalha satisfatoriamente em qualquer browser moderno).

O gerenciamento do servidor de questões (geração de relatórios, correção de questões, etc.) é feito através de console Web, empregando um navegador Internet Explorer 4 ou Netscape 3 ou superiores.

O servidor de questões necessita executar sobre um servidor de páginas Web compatível com Windows, como o Internet Information Server (IIS). Pode ainda ser executado sobre um *Personal Web Server* (PWS).

O mecanismo de geração de relatórios padrão (*Enterprise Reporter*) utiliza a tecnologia de *Active Server Pages* (ASP), da Microsoft, sendo possível o emprego de qualquer outro sistema gerador de relatórios (*Crystal Reports*, *Access*) que seja compatível.

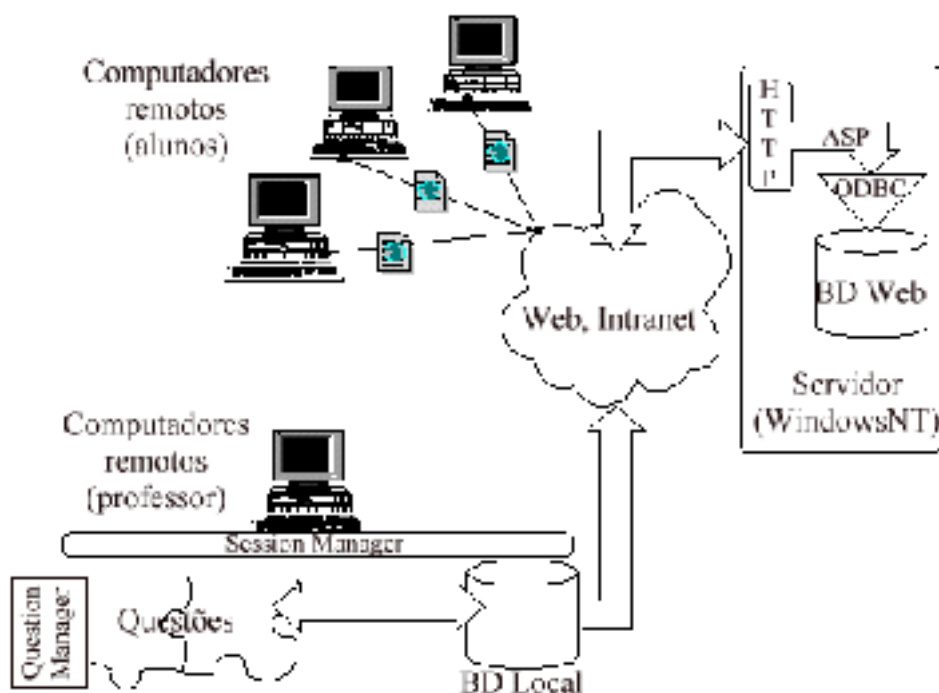


FIGURA 2.1 - Esquema geral do ambiente Perception.

O sistema Perception (figura 2.1) encontra-se dividido em duas partes: uma destinada à montagem das questões e outra destinada a controlar as sessões de aula a serem seguidas pelo aluno (ou sequência de aulas na qual as questões encontram-se inseridas), cujas ferramentas são o *Question Manager* e o *Session Manager*, respectivamente.

### 2.1.1 Elaboração e armazenamento de questões

As questões são armazenadas em um banco de dados local de questões (*Question Database*), em formato proprietário da Question Mark, compatível com o banco de dados MS Access 97.

Dentro do banco de dados de questões, estas são organizadas em uma hierarquia de tópicos e geradas por uma linguagem proprietária da QuestionMark, denominada QML (*Question Mark Language*). Esta plataforma foi desenvolvida para garantir certa portabilidade aos códigos das questões.

As questões podem ser criadas com auxílio de assistentes (figura 2.2) que são capazes de orientar o professor em qualquer passo de sua criação. A personalização do ambiente da avaliação é possível, uma vez que o professor pode modificar parâmetros (cor de fundo, texturas, cor dos caracteres, tipo e tamanho de fonte, etc.) que serão inseridos nas páginas exibidas aos alunos durante as avaliações.

Gráficos e recursos multimídia são manipulados fora do ambiente *Question Database*, sendo tratados de forma independente das questões.

Quanto aos tipos de questões suportados pelo Perception, podemos citar:

- Explicação - As orientações feitas pelo professor e que precedam às questões recebem tratamento semelhante às questões a serem resolvidas pelo aluno;

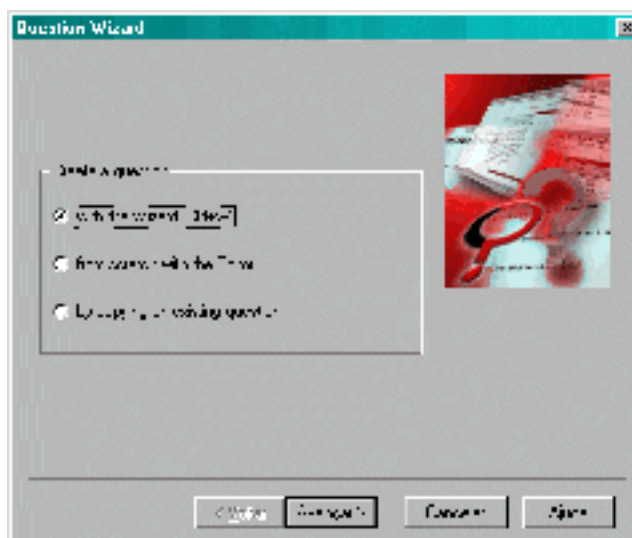


FIGURA 2.2 - Aspecto do assistente para elaboração de questões do Perception.

- Preencher lacunas - Questão de preenchimento onde o aluno deve adicionar palavra ou palavras para responder ou completar o sentido de uma frase;
- Clicar em áreas quentes (*Hotspot*) - Recurso multimídia que permite ao aluno marcar áreas da imagem de uma questão como parte da resposta ;
- Matriz de respostas - Permite ao aluno responder questões em uma lista, clicando em um componente tipo “radio button”;
- Múltipla escolha - Permite ao aluno selecionar de um número de alternativas a única resposta a um texto ou fórmula ;
- Múltipla resposta - Neste tipo de questão, semelhante a “múltipla escolha” , sendo permitido ao aluno escolher mais de uma alternativa;
- Questão numérica - Permite ao aluno inserir valores numéricos (inteiros ou não) como resposta a uma questão;
- Seleção - Exibe ao aluno um menu tipo “*drop-down*” onde as alternativas são exibidas sob forma de questionamento tipo “verdadeiro ou falso”; e
- Coincidência - Tipo de questão que permite ao aluno escrever texto como resposta.

O ambiente *Question Manager* (figura 2.3) permite ao professor elaborar as questões, inserindo formatos e imagens, e também atribuir escores e realizar simulações locais das questões antes de sua publicação no banco de dados Web.

Adicionalmente, o ambiente Perception permite que o professor possa inserir comentários de estímulo ou correções a serem exibidos para o aluno em situações em que venha a acertar ou errar questões.



FIGURA 2.3 - Vista do Question Manager.

### 2.1.1.1 Estrutura da avaliação

O *Question Manager* (figura 2.4) organiza as questões em uma estrutura hierárquica de tópicos, dentro da qual os tipos de questões ou outros subtópicos podem ser inseridos.

O tipo de questão pode ser personalizado através da atribuição de pesos às respostas, como parte da contagem de escores para a avaliação. Comentários e correção imediata podem ser exibidos para o aluno, como parte do processo de correção da aprendizagem.



FIGURA 2.4 - Editor de questões do Question Manager.

### 2.1.2 Estrutura de sessões e avaliações

O *Session Manager* (figura 2.5) organiza as sessões de aula e permite ao professor intercalar questões e orientações ao longo do processo de ensino. Uma base de dados de sessões (*Session Database*) permite armazenar as sessões programadas.

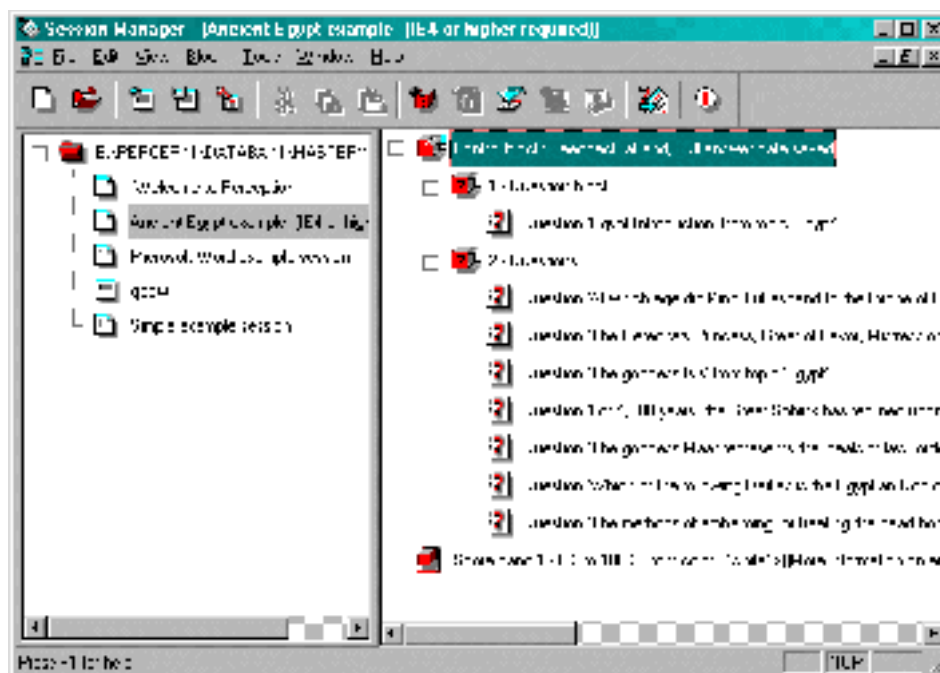


FIGURA 2.5 - O ambiente Session Manager.

Cada sessão é identificada por um título e pode agregar uma ou mais questões.

Os elementos que compõem uma sessão são:

- Bloco de controle;
- Bloco de questões e
- Controle de escores.

O bloco de controle é responsável pelos ajustes de layout do ambiente da questão, tempo de execução de cada sessão, controle do tipo de feedback emitido ao aluno (para quais tipos de resposta deve o sistema emitir feedback) e ajuste de segurança das questões (ajuste de senhas).

O bloco de questões permite importar questões de tópicos previamente elaborados pelo *Question Manager*, embaralhar questões ou empregar diferentes templates do padrão para aquele bloco.

O controle de escores permite ajustar o valor máximo de pontuação (escores) para cada sessão.

### 2.1.3 Entrega das Questões

No Perception, existem duas formas de executar o servidor de questões, o modo aberto (todas as sessões estão disponíveis para todos) ou modo fechado (apenas um aluno agendado tem acesso às sessões). No modo fechado, o aluno entra no servidor informando um nome e senha distribuídos previamente pelo professor e recebe as questões que lhe foram agendadas.

Administradores ou professores podem definir participantes e os testes que eles podem executar empregando seus navegadores em um ambiente web ou Intranet. Isto ajusta os dados em um *Security Database* a quem o servidor de questões consulta antes de entregar as questões.

O sistema permite definir participantes, suas senhas e atribuir participantes a

grupos. O grupo pode ter qualquer número de participantes e um participante pode pertencer a qualquer grupo.

No início da sessão, o primeiro bloco é enviado ao aluno, sob forma de uma página HTML ou questão, dependendo do que fora programado pelo professor. Quando a questão é recebida o *software* servidor processa a resposta e retorna (ou não, dependendo do ajuste feito pelo professor) um *feedback* ao aluno e armazena o resultado. Este processo continua até o término da sessão, quando pode ser exibida uma tela de final ou até inicializar uma nova sessão.

### 2.1.4 Relatórios

O ambiente Perception permite a geração de relatórios através de uma ferramenta denominada *Enterprise Reporter*, capaz de oferecer uma gama de consultas ao banco de dados através do navegador. O sistema permite ainda o uso de ferramentas comerciais como *Crystal Reports* ou até mesmo o *Access* para tal finalidade.

## 2.2 Web CT

O WebCT [GOL98] é uma ferramenta que propicia a criação de sofisticados ambientes educacionais baseados na Web, podendo ser usada para criar desde cursos *on-line* até a divulgação de material suplementar para algum curso.

Este sistema foi desenvolvido em PERL sobre a plataforma UNIX, possuindo cerca de 40.000 linhas de código, sendo utilizado por mais de 500 instituições, incluindo a University of British Columbia, com aproximadamente 140 cursos. Este sistema pode ser dividido sob três aspectos:

- Ferramenta de apresentação: permite ao projetista do curso definir o *lay-out*, cor, textos, contadores, para as páginas do curso;
- Conjunto de ferramentas do estudante; e
- Conjunto de ferramentas do administrador.

Para o estudante são disponibilizadas as ferramentas de comunicação disponíveis na Internet como as listas de discussão, correio eletrônico e salas de chat, sendo que das seis salas de chat existentes, quatro têm suas conversas registradas em um Banco de Dados, para que se possa monitorar a conversação e a participação dos alunos.

As questões de múltipla-escolha podem ser colocadas em algumas páginas do curso, e uma explicação é 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 imediato, ou seja, sempre é dado um *feedback* ao aluno.

Cada aluno tem acesso às notas que recebeu em todas as atividades já realizadas. Existe, também, uma área para a apresentação do projeto desenvolvido em grupo, que pode ser visualizada por todos os integrantes do curso.

As ferramentas de administração são utilizadas para auxiliar na entrega, manutenção e desenvolvimento do material do curso. As informações sobre o progresso dos alunos são obtidas nas seguintes páginas:

1. Página resumo: apresenta os estudantes (nome completo e identificador de usuário no sistema), e informações sobre a data do primeiro acesso, o último acesso e o número total de acessos;
2. Perfil do estudante: aqui além das informações da página de resumo e do título da última página visitada pelo aluno, tem-se também informações adicionais e mais detalhadas do aluno. Tem-se acesso a três tipos de páginas:
  - Distribuição do acesso: tipo de acesso que o aluno tem feito (figura 2.6), apresentando um valor e o percentual do tempo gasto com acesso a: conteúdo do curso, glossário, objetivos da aprendizagem, referências externas, anotações em sua página, questões, listas de discussão e leitura das mensagens, mensagens e respostas enviadas;

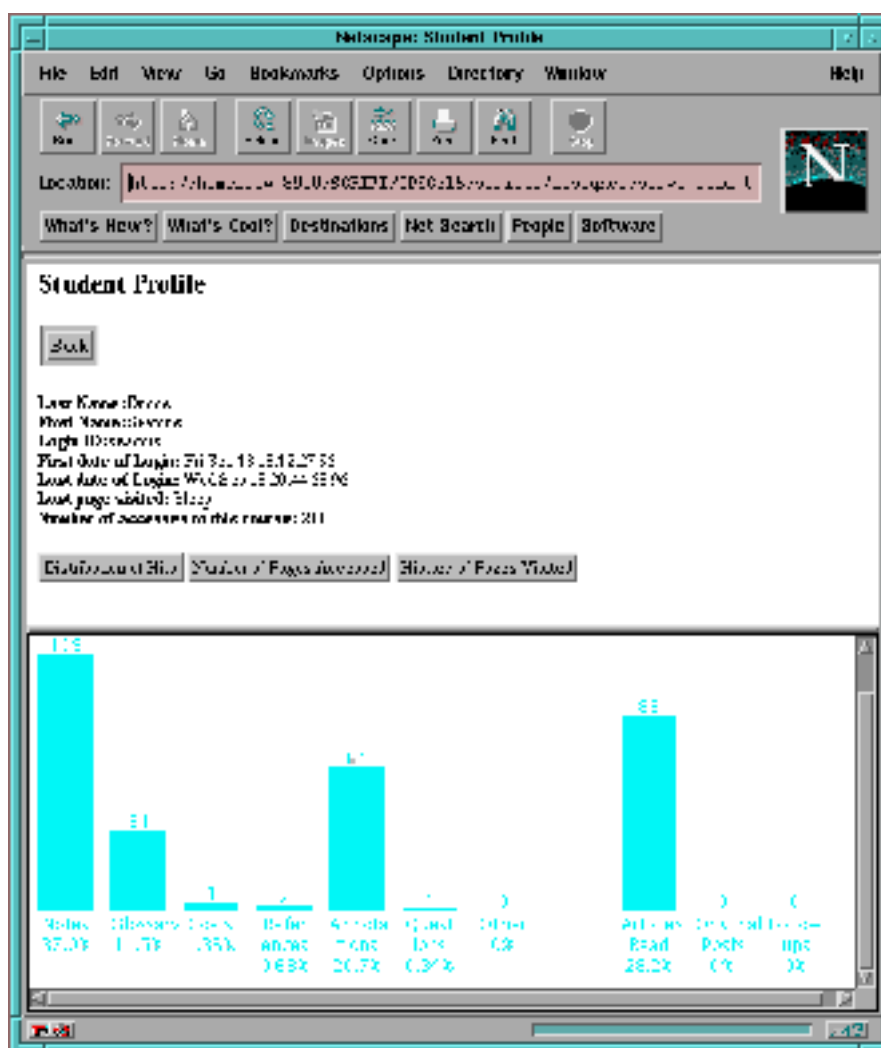


FIGURA 2.6 - Tela da ferramenta de administração de perfil de aluno, do WebCT.

- Número de páginas acessadas: número de páginas que o aluno acessou em relação ao número total de páginas que deverá acessar;
  - Histórico das páginas visitadas: páginas que foram acessadas pelo aluno, bem como o tempo gasto em cada acesso. Desta forma pode-se perceber as páginas em que o aluno gasta mais tempo;
3. Página de uso: relaciona o uso de cada componente do curso, apresentando o



número de acessos, tempo total e médio que os alunos gastaram acessando o componente, bem como mensagens enviadas.

Através de ações simples, pode-se modificar a indexação das tabelas, sendo que os resultados destas consultas podem ser copiados para uma área de trabalho e serem reaproveitados em outro componente, como por exemplo no serviço de e-mail.

Existe também um Banco de Dados de questões, onde as questões são armazenadas e agrupadas por tópicos. Podem ser questões do tipo: verdadeiro/falso, múltipla-escolha, combinação, preencher espaço em branco ou resposta curta.

Todas as questões têm *feedback* imediato indicando a resposta correta e alguma informação extra, menos o tipo de questão denominado “resposta curta” que possui um modelo de correção aproximado, onde inclui-se o que é correto/errado. É mantido um histórico da performance dos alunos em cada questão, para saber as questões onde os alunos apresentam maior dificuldade.

Todo acesso ao WebCT é controlado pelo nome do usuário e senha. Pode-se também definir importância para cada atividade, permitindo assim que o conceito final seja gerado automaticamente. O projetista define os campos do Banco de Dados que interessam e o peso de cada um.

## 2.3 AulaNet

O AulaNet [EDU00] é um ambiente de aprendizagem cooperativo baseado na Web, desenvolvido no Laboratório de Engenharia de Software (LES) do Departamento de Informática da Pontifícia Universidade Católica do Rio de Janeiro (PUC-RJ), para a criação e assistência de cursos à distância sobre os mais variados assuntos.

Os cursos do AulaNet baseiam-se nas seguintes premissas:

- O autor do curso não precisa ser um especialista em Internet;
- Os cursos criados devem buscar grande capacidade de interatividade, a fim de atrair maior participação do aluno no processo de aprendizado;
- Os recursos oferecidos para a criação dos cursos devem corresponder aos de uma sala de aula convencional, acrescidos de outros normalmente disponíveis no ambiente Web; e
- Os professores devem selecionar os mecanismos que utilizarão no curso.

A comunicação entre professor e aluno, e entre alunos-alunos, é realizada através dos seguintes mecanismos:

- Contato com o professor: permite a comunicação assíncrona com o professor;
- Grupo de discussão: é a lista de discussão do curso, permite que as mensagens sejam armazenadas para futuras consultas;
- Grupo de interesse: permite a discussão encadeada sobre um assunto específico, como em ferramentas de *Newsgroups*; e
- Debate: permite a comunicação síncrona, puramente textual.

O AulaNet oferece três métodos de avaliação: prova, trabalho e exercício. Através de exercícios e trabalhos, os alunos podem debater, criar projetos e compartilhar experiências, isto é, participar ativamente do processo de aprendizado.

Através de provas, o professor pode fazer a avaliação formativa do processo de aprendizagem, enfatizando a importância dos aspectos cognitivos da aprendizagem.

As provas são controladas por uma ferramenta de criação e correção automática chamada Quest.

Os objetivos do Quest são auxiliar o professor na criação de provas, dar feedback aos alunos e gerar relatórios para o professor, pois estes relatórios são importantes para que o professor seja capaz de avaliar o quanto os alunos aprenderam, visando o atingimento dos objetivos do processo de aprendizagem.

## **2.4 AvalWeb**

Desenvolvido pela Universidade Federal do Rio Grande do Sul, o AvalWeb [CAR01] é um sistema de gerência de questões e aplicação de avaliações .

O termo "gerência" é utilizado com o objetivo de atender requisições dos professores para elaboração de avaliações, escolher o nível de dificuldade das questões que comporão a avaliação e usar estratégias interativas para aplicação de provas, como por exemplo, a utilização de níveis de dificuldade progressivos das questões submetidas aos alunos, dependendo de suas respostas anteriores.

Como as demais ferramentas apresentadas, utiliza a ferramenta PHP para a geração dinâmica de páginas Web, em conjunto com banco de dados MySQL, acessado via ODBC.

Neste ambiente existem parâmetros que definem a forma geral da avaliação que será disponibilizada aos alunos e devem ser informados pelo professor antes que seja realizada a geração de avaliações. Ao cadastrar questões no sistema, o professor deve informar juntamente com outros dados, seu nível de dificuldade estimado. Cada questão pode ter um nível de dificuldade que pode ser do tipo (i) Muito fácil, (ii) Fácil, (iii) Média, (iv) Difícil ou (v) Muito difícil. Para geração de avaliações, a definição dos níveis de dificuldade das questões é um dos parâmetros que possuem preenchimento obrigatório.

## **2.5 Resumo das características das ferramentas analisadas.**

Visando identificar semelhanças, diferenças e deficiências, foi feita uma comparação entre as principais características das ferramentas para avaliação em EAD analisadas.

Pode-se observar que a maioria das soluções comerciais são baseadas em ambiente Windows NT (embora o Web CT suporte outros sistemas operacionais), empregam bancos de dados Microsoft Access e suas questões são geradas de forma semelhante, ou seja, mediante consulta a um banco de dados e geração dinâmica de questões, segundo templates definidos pelo sistema e personalizados pelo professor.

O aluno interage com o ambiente, recebendo questões e enviando respostas, à medida que executa a avaliação. Os sistemas oferecem ao professor suporte para a obtenção de graus, estatísticas e acompanhamento de resultados.

Com base no estudo realizado de algumas das principais ferramentas empregadas para avaliação remota em ambientes de EAD, pode-se observar suas principais características, conforme a tabela 2.1 :

TABELA 2.1- Comparação entre as ferramentas de avaliação para EAD comerciais.

<i>Aspectos</i>	<i>Question Mark Perception</i>	<i>Web CT</i>	<i>AulaNet</i>	<i>AvalWeb</i>
<i>Linguagem empregada</i>	QML (proprietária).	PERL e Scripts CGI.	ASP	PHP
<i>Banco de dados</i>	MS Access 97	Proprietário	MSAccess (ODBC)	MySQL
<i>Criação de questões</i>	Através de ferramenta Question Manager, instalada no ambiente do professor.	Via browser ou através de ferramenta Quizz Tool, instalada no ambiente do professor.	Através de ferramenta Quest.	Através de ambiente implementado via browser
<i>Entrega de questões ao aluno</i>	Páginas HTML, descarregadas do servidor. Modo aberto (livre) e modo fechado (requer senha).	Páginas HTML, geradas dinamicamente, descarregadas do servidor.		
<i>Lógica do ambiente</i>	Baseada no servidor.			
<i>Suporte a multitarefa</i>	Não.	Não.	Não.	Não.
<i>Suporte a multimídia</i>	Sim.	Sim.	Sim.	Sim.
<i>Uso de agentes</i>	Não.	Não.	Não.	Não.
<i>Tipos de avaliações executadas</i>	Agendadas, assíncronas, individuais.	Assíncronas, individuais.		
<i>Autorecolhimento de avaliação</i>	Não. Bloqueio de envio após tempo de solução.	Não.	Não.	Não.

Estes ambientes apresentam algumas deficiências, tais como:

- A impossibilidade de emprego de agentes ou tutores embarcados no ambiente do aluno para o tratamento imediato de situações de falha ou dúvidas durante a avaliação, por limitações impostas pelo uso de páginas HTML;
- A possibilidade de armazenamento no ambiente do aluno das páginas geradas dinamicamente, graças a recursos providos pelos próprios navegadores;
- Os sistemas não oferecem recursos que possibilitem uma avaliação formativa do aluno; e
- O professor não tem como fazer uma auto-avaliação no que diz respeito à estratégias de ensino e de avaliação adotados, recorrendo a pesquisas de opinião e análise dos resultados obtidos pelos alunos.

Com base no estudo destas deficiências, o presente trabalho desenvolveu uma ferramenta capaz de prover um ambiente de avaliação cuja interface emprega agentes especialistas, denominados tutores embarcados, capazes de acompanhar e monitorar o aluno durante o processo de avaliação.

Adicionalmente, o sistema permite monitorar aspectos comportamentais, recurso útil para que o professor possa rever enunciados, recursos disponibilizados e até estratégias de ensino adotadas.

## **2.6 Resumo do capítulo dois**

Neste capítulo foram apresentadas considerações sobre ambientes de Ensino à Distância e ferramentas disponíveis para avaliação remota.

Buscando estabelecer os modelos das principais ferramentas de avaliação para ambientes de EAD, foram feitas comparações entre eles. Inicialmente foi analisada a ferramenta Perception, da Question Mark, em seguida as ferramentas do Web CT e do ambiente AulaNet.

Finalmente, foram feitas comparações entre as principais ferramentas, buscando o levantamento de suas deficiências e vantagens, tomando-os como base para a proposta do ambiente Javal.

### 3 Considerações sobre o processo de avaliação do processo ensino-aprendizagem em EAD

Neste capítulo será apresentada uma breve descrição das idéias que nortearam o desenvolvimento do ambiente Javal, podendo servir como base para trabalhos futuros ou desenvolvimento de novos ambientes.

Serão inicialmente apresentadas considerações sobre a avaliação do processo ensino-aprendizagem em ambientes de Ensino à Distância, abordando problemas existentes e soluções propostas.

Após, são abordados aspectos ligados à concepção de agentes para este tipo de ambiente, focalizando modelos e comportamentos esperados, sendo apresentada uma breve descrição dos agentes que compõem o ambiente Javal.

Cabe ressaltar que não é objetivo do presente capítulo a apresentação de aspectos ligados à psicologia da aprendizagem tampouco abordagens técnicas sobre metodologias de avaliação e de ensino. As idéias descritas a seguir são fruto de análise e estudo de diversos trabalhos na área de Avaliação à Distância e buscam justificar o modelo de agentes desenvolvido para ser implementado e validado por este protótipo.

Para a validação deste trabalho, serão especificados detalhadamente nos capítulos 4 e 5 os módulos e API<sup>1</sup> de classes pertencentes ao ambiente Javal.

O ambiente Javal apresentado neste trabalho não tem intenção de ser um modelo monolítico, podendo ser expandido e personalizado, dentro do que preconiza Paul Dourish em *Open Implementation and Flexibility in CSCW Toolkits* [DOU96].

#### 3.1 Considerações iniciais

O processo de ensino está ligado ao conceito de mudança de comportamento, baseado no aprendizado e o desenvolvimento de aptidões. São três as áreas ligadas ao ensino, segundo a Taxionomia de Bloom [BLO64]:

- Cognitivo;
- Afetivo (comportamental); e
- Habilidade.

Cognitivo - envolve o conhecimento e o desenvolvimento de destreza intelectual. Isto inclui a lembrança ou reconhecimento de fatos específicos, padrões de procedimentos e conceitos que possam servir para o desenvolvimento de habilidades intelectuais e destreza.

Afetivo - inclui a forma na qual lidamos com as coisas emocionalmente, como sentimentos, valores, apreciações, entusiasmo, motivações e atitudes.

Habilidade - Também conhecido como domínio psicomotor, encontra-se ligado à capacidade de movimento, coordenação e uso das áreas de habilidade motora. As medições deste tipo de comportamento, encontram-se ligados aos aspectos de velocidade, precisão, distância, etc.

Para atingir estas áreas, um modelo de ambiente educacional padrão geralmente possui a seguinte estrutura, dentro dos conceitos e termos compilados em [LEI01]:

<sup>1</sup> API - Application Programming Interface – Conjunto formado por classes implementadas e sua documentação, necessários para o desenvolvimento do ambiente. O ambiente Javal propõe uma API própria denominada Kit de Desenvolvimento Javal (KDJ).

- Unidade de estudo - A matéria que se deseja ensinar ;
- Assunto - O assunto (ou módulo) que faz parte da matéria em questão; e
- Objetivo de Aprendizagem - Conjunto de metas a serem atingidas, como comportamento mínimo a ser manifestado como resultado do processo ensino-aprendizagem. É na medida do grau de aproximação deste objetivo que trabalham os sistemas de avaliação.

A avaliação do ensino-aprendizagem é a medida da relação existente entre o comportamento manifestado pelo aluno e o esperado, de acordo com os objetivos estabelecidos para aquele determinado assunto. Para cada assunto apresentado ao aluno podem corresponder um ou mais objetivos ou comportamentos. A dificuldade está em se determinar quais aspectos devem ser avaliados e qual (ou quais) deles melhor permite quantificar o grau de aprendizado do aluno sobre determinado assunto.

Segundo [BLO64], existem três modalidades de avaliação amplamente conhecidas: somativa, diagnóstica e formativa, cada uma delas com uma função específica:

- **Formativa**- Ocorre durante o processo de instrução; inclui todos os conteúdos importantes de uma etapa da instrução; fornece “feedback” ao aluno do que aprendeu e do que precisa aprender; fornece feedback ao professor, identificando as falhas dos alunos e quais os aspectos da instrução que devem ser modificados; busca o atendimento às diferenças individuais dos alunos e a prescrição de medidas alternativas de recuperação das falhas de aprendizagem.
- **Somativa**- Ocorre ao final do processo de instrução com a finalidade de verificar o que o aluno efetivamente aprendeu; inclui conteúdos mais relevantes e os objetivos mais amplos do período de instrução; visa à atribuição de notas; fornece “feedback” ao aluno (informa-o quanto ao nível de aprendizagem alcançado), se este for o objetivo central da avaliação formativa; presta-se à comparação de resultados obtidos com diferentes alunos, métodos e materiais de ensino.
- **Diagnóstica**- Ocorre em dois momentos diferentes: antes e durante o processo de instrução; no primeiro momento, tem por funções: verificar se o aluno possui determinadas habilidades básicas, determinar que objetivos de um curso já foram dominados pelo aluno, agrupar alunos conforme suas características, encaminhar alunos a estratégias e programas alternativos de ensino; no segundo momento, buscar a identificação das causas não pedagógicas dos repetidos fracassos de aprendizagem, promovendo, inclusive quando necessário, o encaminhamento do aluno a outros especialistas (psicólogos, orientadores educacionais, entre outros).

Para o ambiente de avaliação, um dos principais problemas reside na parametrização da relação ensino-aprendizagem, o que pode ser obtido através das seguintes perguntas:

- Que comportamentos evidenciados pelo aluno permitem supor que o mesmo entendeu o assunto? (respondeu precisamente? dentro do tempo previsto?);
- Como saber se o aluno demonstrou habilidade na execução da tarefa ou no manuseio de determinado recurso? (aspecto motor - como o aluno reagiu fisicamente frente ao problema? Consegue agir com a habilidade requerida

para a solução do problema?);

- O aluno pode chegar a solução com base no que aprendeu? (aspecto cognitivo- ele aprendeu a idéia?);
- Como saber se o aluno é capaz de empregar os conhecimentos obtidos para interagir com seu próprio ambiente, de forma a permitir a solução de problemas? (e atingir os objetivos propostos?);
- Como saber se o aluno encontrou dificuldades ao solucionar o problema proposto?

A resposta a algumas destas perguntas em uma situação de avaliação não presencial e assíncrona, como a existente em avaliações no ambiente de Ensino à Distância através da rede ou Internet, pode implicar no emprego de componentes de hardware específicos (sistemas de vídeo-conferência, web-câmeras) ou dispositivos de software especializados no próprio ambiente de avaliação, cuja finalidade é obter informações colhidas como fruto da interação entre o sistema e o aluno.

Desta forma, durante a execução de uma avaliação, pode-se monitorar o comportamento do aluno, perceber suas dificuldades e avaliar o resultado obtido como forma de medir a relação ensino-aprendizagem.

O modelo proposto engloba as avaliações somativa e formativa, esta última pelo emprego de um sistema de agentes para monitoração do comportamento do aluno e tratamento de possíveis falhas ocorridas durante a execução da avaliação.

A proposta envolve o desenvolvimento de um tipo de agente que, devidamente instruído, seja capaz de monitorar o comportamento do aluno durante a execução de avaliações, oferecendo-lhe suporte imediato e realimentação ao professor, buscando validar a metodologia de ensino adotada, o emprego de recurso e o ajuste de perfis de aluno, quando necessário.

### **3.1.1 Os problemas da avaliação remota e o processo ensino-aprendizagem**

Para o professor, a avaliação deve medir o aprendizado do aluno, o que pode espelhar a eficiência da estratégia de ensino adotada e dos recursos empregados. Para que esta avaliação possa ser precisa, tornam-se necessárias outras ferramentas ou recursos para a avaliação.

O ambiente de Ensino à Distância via Internet ou Intranet, por sua característica descentralizadora espacial e temporalmente, dificulta o trabalho de se obter uma avaliação informal precisa, sobretudo durante a execução de testes ou provas.

Como exemplo, estudos foram realizados por [JAQ99] no sentido de se considerar participações de alunos em fóruns, listas de discussão e chat como fontes capazes de fornecer informes que permitam quantificar a participação do aluno do processo de ensino-aprendizagem, tomando-os como base para avaliações informais.

Um dos problemas nesta área resume-se em como desenvolver um sistema informatizado que seja capaz de “perceber” o grau de dificuldade encontrado por um aluno durante a solução de uma avaliação distribuída de forma assíncrona e dispersa no espaço, em função da monitoração de seu comportamento.

Isto poderia ser útil para:

- Adequação do conteúdo ou formato da avaliação ao perfil do aluno ;
- Obtenção de *feedback* sobre a elaboração da avaliação;
- Adequação do ensino ao perfil (obtenção de feedback para traçado de novas estratégias); e
- Levantamento de adaptações, reajustes e soluções alternativas a serem implementadas.

Adequação do conteúdo da avaliação ao perfil - em ambientes de ensino tradicional, o perfil do aluno não é considerado para a elaboração da avaliação. Os exemplares são cópias de uma avaliação elaborada com base em um aluno com características-padrão. Como exemplo, deficientes visuais ou auditivos necessitam de provas diferentes de pessoas que não possuem limitações físicas. De forma análoga, perfis psicopedagógicos distintos também deveriam merecer o mesmo cuidado.

Obtenção de feedback sobre a elaboração da avaliação - as dificuldades encontradas pelo aluno podem ser fruto de falhas na estratégia de ensino adotada ou da elaboração da avaliação.

Adequação do ensino ao perfil - o sistema pode fornecer parâmetros para o professor, no sentido de permitir possíveis ajustes em estratégias de ensino adotadas.

Levantamento de adaptações, reajuste e soluções - o sistema pode fornecer subsídios importantes que permitam indicar reajustes ou adaptações na sistemática de ensino ou de elaboração de avaliações futuras.

Em avaliações presenciais, comportamentos que evidenciam situações de não compreensão ou não entendimento de questões (tiques nervosos, olhar disperso, falta de concentração, etc.) podem ser monitorados pelo professor em tempo de desencadear um processo de auxílio.

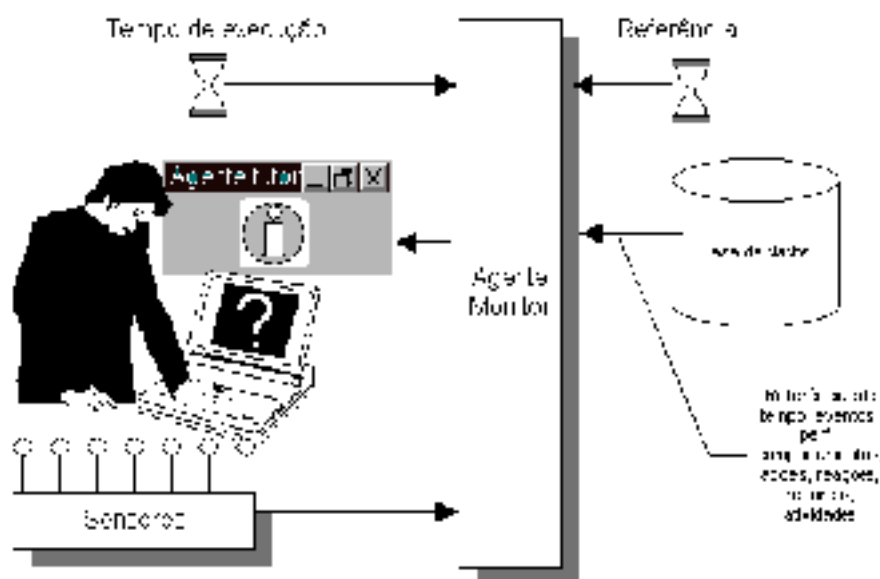
Em ambientes de avaliação de Ensino à Distância, tal procedimento torna-se impossível devido à característica de dispersão espacial e temporal destes ambientes. Para estes casos, torna-se necessário um elemento de software especializado capaz de identificar, com base no comportamento do aluno, se o mesmo deixa de responder uma questão:

- a) Por não saber a resposta (houve falha no aprendizado?);
- b) Por não ter entendido a proposição da questão (houve falha na elaboração da avaliação?);
- c) Por não ter destreza no uso do recurso disponível (houve falha da seleção da estratégia de avaliação para o perfil?);
- d) Por outros fatores locais incidentes sobre o aluno (falhas temporais - stress, desinteresse temporário, o ambiente em que ele se encontra no momento da execução da questão, etc.?);
- e) Por fatores técnicos (congestionamentos de rede, dificuldades na transmissão e recepção de pacotes, etc?).

Uma solução possível envolve o emprego de sensores (figura 3.1) que permitam aos elementos especialistas monitorar o aluno, identificar situações de falha e desencadear atividades que procurem conduzi-lo a atingir os objetivos propostos.

FIGURA 3.1 - Um aluno e o sensoreamento possível para um ambiente informatizado.





Estes especialistas, denominados agentes, atuam no ambiente do aluno de forma a não apenas avaliá-lo, mas assisti-lo, ajudando-o na compreensão da questão, favorecendo o “insight”- agindo no componente cognitivo - e, ao mesmo tempo, oferecendo *feedback* ao instrutor sobre o seu comportamento visando a aprimorar o processo de avaliação e do ensino.

Cabe ressaltar que estes “sensores” do comportamento do aluno estão limitados quanto ao espectro de seu monitoramento. Fatores como fadiga, stress, desinteresse, desmotivação, desatenção, incompreensão do assunto, etc. são difíceis de serem monitorados por um sistema informatizado, por estarem relacionados ao estado físico-psicológico do aluno e à capacidade de percepção do professor. Deste modo, o planejamento do ensino e avaliação remota deve levar sempre em consideração que o aluno está motivado e disposto a aprender, uma vez que sistemas informatizados ainda não permitem a monitoração destes aspectos.

Quanto aos agentes, estes devem ser instruídos pelo professor sobre como se apresentar e se comportar ante as dificuldades do aluno. Estas instruções referem-se:

- Ao aspecto do seu ambiente gráfico para exibição ao aluno (cor de tela, ícones, animações, sons, etc.);
- À adequação da linguagem ao aluno considerado (de forma impessoal ou personalizado, utilizando “linguajar” adaptado ao tipo de aluno-alvo da avaliação. É interessante que o agente seja instruído pelo professor sobre como “falar” com o aluno); e
- Ao emprego de recursos, como: documentação, ferramentas integradas ao ambiente (salas de chat, fóruns, etc.), páginas web, filmes, arquivos multimídia, etc, que o agente poderá disponibilizar para o aluno quando necessário.

Durante a execução da avaliação, os agentes devem permanecer atentos ao comportamento do aluno, monitorando os sensores, registrando informações e agindo conforme diretrizes do professor.

Finalmente, o modelo computacional ideal para desenvolvimento de sistemas de agentes inteligentes para avaliação em EAD deve considerar possibilidades técnicas

favoráveis ao seu emprego. Por este motivo, a solução deve ser capaz não apenas de detectar as situações de falha do aluno e iniciar processos tutores, mas de assegurar a confiabilidade e integridade do ambiente, sob pena de fragilizar o modelo, levando-o ao descrédito e falência. Para isto, deve incorporar recursos capazes de solucionar problemas julgados como “essencialmente técnicos”, uma vez que o ambiente da Internet deve ser considerado hostil para o desenvolvimento de sistemas distribuídos e avaliações remotas.

### **3.1.2 O papel do professor na avaliação remota em um ambiente que emprega agentes embarcados**

Em ambientes de avaliação à distância que empregue agentes embarcados é interessante que o professor se faça presente ao lado do aluno através dos seus agentes. Estes devem ter a capacidade de agir segundo suas orientações, oferecendo ao aluno confiança e segurança para a solução das questões.

Para que os agentes possam realizar sua tarefa de acompanhamento, torna-se necessário definir dentro de quais parâmetros os sensores devem ser mensurados e analisados sob forma de identificar a ocorrência de situações de dúvida ou falha da avaliação. Cabe ao professor fornecer estas informações ao sistema, como fruto da experiência em atividades de avaliação presenciais.

Torna-se interessante, em trabalhos futuros, o desenvolvimento de agentes ou recursos de IA capazes de auxiliá-lo nesta tarefa. Devem ser considerados alguns fatores como:

- O grau de intimidade do aluno com o ambiente - para alunos iniciantes é normal uma demora na solução das questões; e
- O tipo de perfil de aluno em questão - alunos com perfis específicos ou portadores de algum tipo de deficiência, demoram mais a responder do que outros por possuírem menos destreza ou compreensão ou necessitarem consultar recursos específicos.

O professor deve instruir os agentes durante a elaboração da avaliação, de forma a permitir um auxílio eficiente e eficaz.

O modelo do ambiente Javal emprega aspectos temporais e comportamentais como parâmetros na avaliação. Neste caso, o professor deve:

- Estipular tempos máximo e mínimo capazes de caracterizar um possível não entendimento (*timeout*) ou comportamentos de resposta precipitada (“chute”);
- Estipular quais eventos causados pelo aluno devam ser entendidos pelos agentes como “fuga”, bem como estipular o número de ocorrências toleráveis para a questão, a partir da qual o agente possa interpretar como sendo uma situação de dúvida;
- Determinar quais questões devam ter tutores para auxílio;
- Determinar como os tutores deverão abordar os alunos (personalização de mensagens, linguagem, cores e formas do ambiente);
- Informar aos tutores quando deverão entrar em ação e quais os recursos que poderão empregar para auxiliar o aluno;

- Disponibilizar recursos úteis para consulta on-line, baseados nos objetivos intermediários seguidos durante o processo de ensino. Utilizar página da disciplina como fonte de consulta e recursos como fórum de discussões e salas de chat podem ser adicionados.
- Adequar os recursos ao tipo de perfil de aluno. É conveniente que o professor crie formas alternativas de atingir o aluno, considerando-se o enunciado da questão e a forma como a mesma será respondida.

Como visto, um modelo de tutor capaz de ajudar ao aluno deve ser instruído pelo professor, ao montar a avaliação, para disponibilizar ao aluno recursos específicos, como palavras chaves (*keywords*), links, etc. capazes de levá-lo ao “insight”<sup>2</sup>.

São estes elementos que - em situações de falha - podem decisivamente influenciar o destino do aluno. Adicionalmente, deve este tutor ser capaz de emitir-lhe estímulos quando do acerto de determinadas questões, reforçando comportamentalmente o ambiente do aluno, a exemplo de sistemas de avaliação existentes como o Question Mark e o Aula Net.

### 3.2 Uma proposta de solução computacional para os problemas

As considerações abordadas daqui por diante serão colocadas de forma a apresentar o modelo concreto dos agentes, buscando validar o ambiente de avaliação apresentado.

Foram implementados dois tipos de agente cujas funcionalidades permitem monitorar o comportamento do aluno durante a execução das tarefas. Esta monitoração ocorre por conta da avaliação de “sensores” que permitem acompanhar o desenvolvimento da avaliação.

O modelo elaborado sugere que os agentes atuem quando ocorrerem os seguintes eventos:

- Demora ;
- Precipitação;
- Dúvida ; e
- Omissão.

Demora - um aluno pode demorar a responder uma questão por diversos motivos. O enunciado ou abordagem da questão de forma inadequada para o perfil do aluno considerado podem inibir a sua resposta. Por este motivo, com base em um tempo estimado pelo professor para a sua solução, agentes devem ser capazes de identificar situações de “timeout” e entrar em ação, quer exibindo dicas ou enunciados alternativos disponibilizados previamente pelo professor;

Precipitação - alguns alunos podem se deixar levar pelo tradicional “chute”, marcando a alternativa julgada correta de forma apressada, como se quisessem se livrar do incômodo de resolver a questão. A importância do agente neste tipo de situação requer o emprego de estratégias no sentido de acalmar o aluno, fazendo-o buscar conceitos e rever idéias, antes de marcar a alternativa julgada correta;

<sup>2</sup> Segundo o trabalho de Eynseck [EYS77], é o momento em que o caminho para a solução de um problema torna-se claro. O termo “Eureka”, criado por Arquimedes refere-se à esta experiência.

Dúvida - um aluno que marca uma questão e depois muda sua resposta pode não ter entendido a questão ou realmente não saber completamente o assunto. Neste caso, o sistema deve prover recursos para evitar que a má interpretação do enunciado venha a atrapalhar a elaboração da resposta correta.

Omissão - pular a questão é algo normal para quem responde a questão pela primeira vez, pois às vezes o aluno não possui certeza imediata da resposta e desejaria respondê-la posteriormente. Cabe ao sistema saber quantos comportamentos desta natureza podem ser aceitos para um tipo de questão, antes de desencadear os processos de auxílio.

Os sensores considerados no presente modelo são:

- Eventos da interface gráfica do aluno; e
- Tempo de execução da avaliação.

Os agentes são capazes de desencadear um procedimento de auxílio, visando o tratamento imediato de situações que possam evidenciar falhas na elaboração da avaliação ou da compreensão do aluno, lembrando-lhe dicas ou idéias capazes de lhe provocar o “insight”.

Desta forma, o modelo propõe agentes cujas funcionalidades englobam:

- Monitoramento do aluno (durante a solução); e
- Atividades de tutoramento (em situações de falha).

Por este motivo, os agentes desenvolvidos são denominados agente monitor e agente tutor. Para o ambiente Javal, os agentes monitores desenvolvidos são o Agente Comportamento (ACA) e Agente Tempo (ATA). A relação entre os agentes e o ambiente aparece conforme modelo da figura 3.2

Considerando o comportamento evidenciado frente às tarefas e com base no perfil do aluno e nos objetivos e estratégias disponíveis, os agentes deverão ser capazes de iniciar procedimentos específicos visando:

- Apresentação de dicas ou recursos;
- Apresentação da questão sob novo ambiente (ou alternativo, incluindo gráficos e imagens); e
- Permitir interação direta com o professor.

Apresentação de dicas ou recursos - o professor deve instruir o agente sobre palavras-chaves ou idéias relacionadas com o enunciado ou as proposições. Enunciados elaborados de forma mais simples e mais detalhada, quando exibidos oportunamente, podem ajudar a resolver problemas de dúvidas durante a fase de análise da questão.

Apresentação da questão sob novo ambiente - o sistema deverá apresentar recursos multimídia (som, imagens, filmes, etc.), visando o estímulo de outros sentidos capazes de permitir uma adaptação do ambiente à necessidade do aluno. A apresentação da questão pelo tutor deve permitir ao aluno total controle sobre quais tópicos deseja rever através de opções de navegação disponíveis, bem como quais recursos disponibilizados deseja utilizar.

Permitir interação direta com o professor - esgotados os recursos anteriores, deve encaminhar o aluno ao contato com o professor, preferencialmente em um

ambiente virtual de aula, como uma sala de chat ou sala virtual para a retirada de possíveis dúvidas sobre o enunciado da questão (problema: disponível apenas para avaliações síncronas).

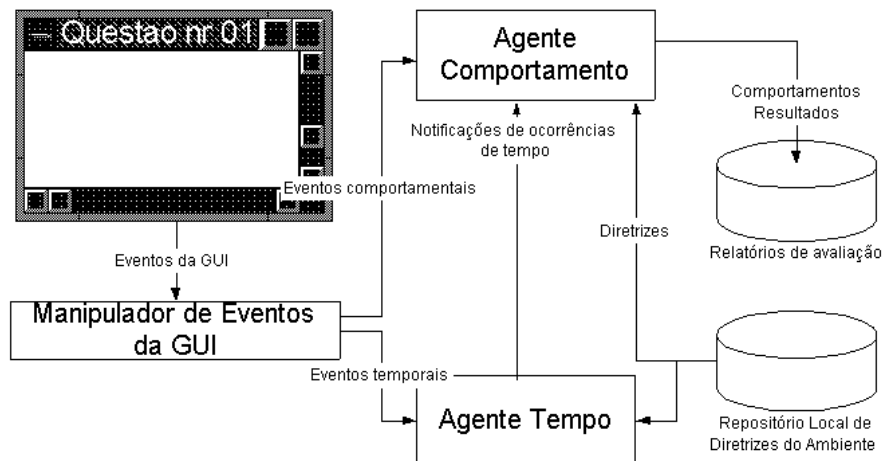


FIGURA 3.2 - Relacionamento entre agentes do Módulo de Avaliação Remota (MAR).

Pelas características de que se reveste este trabalho de pesquisa, espera-se a implementação de novas classes de agentes especialistas, permitindo a personalização de agentes para ambientes de avaliação.

Inúmeros outros agentes podem ser implementados de forma a realizar um completo sensoriamento e monitoramento do comportamento do aluno frente à realização destas atividades, devendo servir como sugestão para a modelagem e implementação de novos agentes em trabalhos posteriores.

A modularidade e o suporte a threads providos pela plataforma Java permitem a criação e emprego de agentes de forma fácil e imediata, justificando esta plataforma como alternativa para seu desenvolvimento.

### 3.3 O desenvolvimento de agentes

O desenvolvimento de agentes para este tipo de ambiente levou em consideração a sua interação com o aluno, através da interface gráfica que lhe é apresentada. As idéias a seguir conduziram a pesquisa e o desenvolvimento dos agentes de tempo (ATA) e comportamento (ACA) presentes no ambiente Javal.

Como visto, os agentes devem ser capazes de acompanhar a interação do aluno com o ambiente de avaliação. Para isto, os agentes devem permitir:

- Monitoração de movimento e cliques de mouse;
- Monitoração de eventos de janela e caixas de texto; e
- Monitoração de tempos de execução de tarefas.

Em nível de tratamento de falhas, os agentes devem ser capazes de:

- Sugerir intervenção na modelagem de perfis, solicitando o ajuste do sistema ao perfil do aluno avaliado;
- Decidir sobre o emprego de recursos com base em premissas (estratégias) específicas designadas pelo professor;

- Permitir interação, de forma a permitir coerência no ajuste de informações, através de troca de mensagens entre agentes.

Para uma correta avaliação de parâmetros e tomada de decisão, os agentes devem possuir acesso a informações sobre a estrutura do sistema no qual estejam inseridos, de forma a permitir que os mesmos conheçam:

- O perfil cognitivo do aluno em questão;
- Os recursos disponíveis (páginas Web, salas de chat, figuras, filmes, etc.) para a atividade em execução;
- Os critérios para emprego destes recursos, com base em diretivas (estratégias) explícitas para cada tipo de perfil ; e
- Os objetivos que servem como base para a avaliação do agente e sua tomada de decisão.

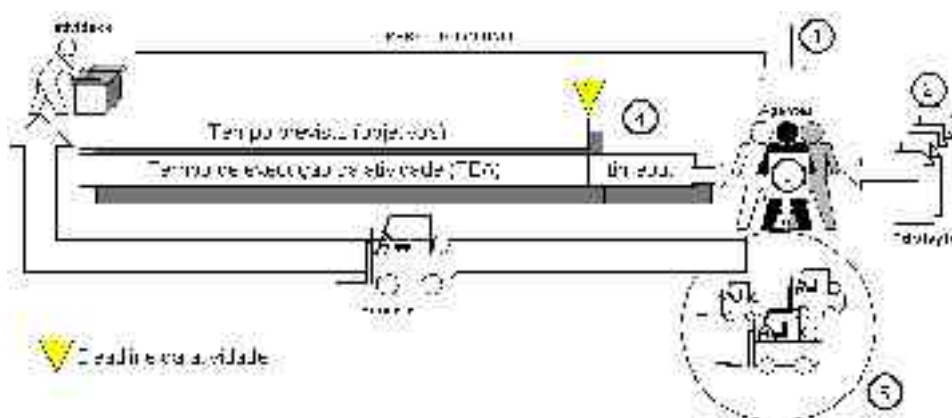


FIGURA 3.3 - A dinâmica de funcionamento da avaliação empregando agentes.

No ambiente Javal, a tarefa do agente pode ser resumida em cinco passos, conforme a figura 3.3:

1. Carga dos dados do aluno (informações pessoais, perfis) (1);
2. Carga do padrão de execução da atividade, como tempo de execução da tarefa, com base na estratégia estabelecida pelo professor (2);
3. Início da atividade proposta e marcação do tempo de duração;
4. Comparação de parâmetros temporais, com base nos objetivos definidos para o perfil, pelos agentes (3); e
5. Caso o tempo de execução da tarefa tenha acabado (4), os agentes detectam uma situação de falha e, nestes casos e com base na estratégia definida pelo professor, são selecionados os recursos adequados (5) para auxiliar o aluno na solução do problema.

De acordo com os perfis disponíveis e *inputs* avaliados, os agentes, em uma situação de falha, devem ser capazes de informar agentes especializados em selecionar as estratégias de avaliação adequadas para cada caso.

### 3.4 Os agentes do ambiente Javal

Como visto, o modelo de avaliação empregado pelo ambiente Javal busca principalmente a avaliação somativa e o acompanhamento e correção da aprendizagem durante este processo, através do emprego de agentes especializados (tutores).

As questões disponibilizadas nestes ambiente são as questões clássicas (verdadeiro-falso, marque alternativa correta, etc.).

O problema maior reside em estabelecer um modelo de agente capaz de supor uma situação de dúvida do aluno e que seja capaz de esclarecê-lo sobre a questão ou colaborar com sua solução, através do estabelecimento de vínculos com assuntos vistos na matéria, através de recursos disponibilizados pelo professor.

Corrigir a aprendizagem é tão importante quanto efetivar o ensino. Mas tão importante quanto corrigir é conduzir o aluno aos objetivos propostos. A técnica empregada por este modelo é o de condução por reforços sucessivos, da qual o agente tutor é o elemento-chave.

O objetivo do agente tutor em avaliações remotas é encaminhar o aluno para o alvo, principalmente nas situações de suspeita de desvio do objetivo esperado. Desta forma, através de recomendações (lembra-se da aula sobre tal assunto?) pode-se conduzir o aluno a obter o “insight” desejado, permitindo-lhe, ao mesmo tempo de que é avaliado, a consolidação de conceitos visando a construção do seu conhecimento [DUF92].

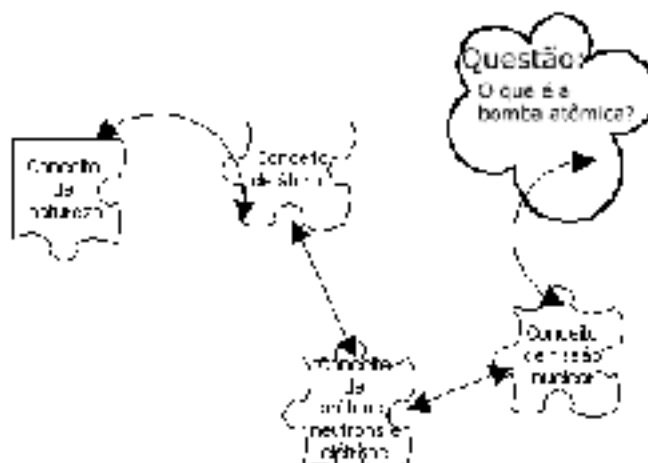


FIGURA 3.4 - Um exemplo de um sistema de dicas.

Para a elaboração de um sistema de dicas eficiente, o processo de avaliação formal deve ser pautado em objetivos bem definidos, de forma que um questionamento possa estar fundamentado em conhecimentos adquiridos, baseados em objetivos intermediários.

Ao aluno, um simples lembrete sobre um assunto visto em aula (sem a necessidade de revê-lo novamente) poderá gerar o “insight” desejado para conduzi-lo ao objetivo proposto.

A ocorrência de falha do aluno é constantemente monitorada. Conforme visto anteriormente, situações do tipo: demora ao responder (desde que previamente testada e não detectada uma situação de falha de rede), questões respondidas de forma errada (apressadamente ou não) são situações de falha, que alertam ao agente tutor de que alguma atitude imediata deva ser tomada.

Nestas situações, o professor e o sistema de EAD ao qual o ambiente que contém os agentes encontra-se integrado devem ser notificados em seguida, dependendo do tipo de instrução que os mesmos receberem do professor que elaborou as questões.

O modelo do ambiente Javal propõe um agente que verifica se o aluno tem certeza de sua resposta e lhe oferece opções de ajuda e até enunciados alternativos, além da consulta ao professor (pois o próprio enunciado pode não estar completamente compreensível).

Os eventos considerados são:

1. Download da avaliação;
2. Oferecimento da questão;
3. Início dos agentes monitores da avaliação;
4. Situação de falha por tempo - iniciar tutor;
5. Situação de falha por erro - iniciar tutor; e
6. Correção

Download da avaliação - o aluno efetua login no site onde se encontra a avaliação e realiza o seu download. A applet de avaliação carrega as classes dos objetos relativos ao ambiente enviados para o aluno.

Oferecimento da questão - a questão é selecionada e exibida ao aluno. As opções são expostas para escolha e os agentes inicializados.

Início do agente avaliação - O agente avaliação entra em ação visando monitorar o tempo de solução da questão e comportamentos estabelecidos como elementos sensores.

Falha por tempo - Se o tempo de solução for curto demais para a resposta (o professor estabelece um tempo médio para a solução de cada questão) ou se o tempo for excessivamente longo, o agente avaliação inicializa o tutor, que visa auxiliar o aluno em sua possível dificuldade.

Falha por erro - Alguns alunos erram questões por responderem rápido demais ou por confiarem em sua intuição. Isto leva o aluno a corrigir respostas já emitidas, o que pode levar a crer que encontrou dificuldades para solucionar. Por este motivo o agente tutor entra em ação, visando evitar o desvio de trajetória, direcionando o aluno para o alvo. Desta forma, a menos que o aluno realmente não tenha entendido a questão, deve-se evitar que decisões impensadas ou incertas conduzam-no ao erro.

Correção - O ambiente corrige a avaliação e entrega-o ao agente Comportamento, gerando relatórios úteis ao professor no traçado de estratégias futuras e adaptação de perfis.

### 3.4.1 O agente tutor

Como visto, o conceito de tutor está ligado a um componente que simula a presença do professor ao lado do aluno, conferindo-lhe segurança e tranquilidade para resolver as questões. Seu objetivo é ter em memória os tópicos relativos ao tema da avaliação para lembrar ao aluno os assuntos vistos relacionados, buscando fazer com que o mesmo consiga o *insight*, solucionando o problema proposto.

Futuramente, espera-se o tutor possa analisar o perfil do aluno, solicitando ao



professor adaptações de recursos a serem disponibilizados de acordo com a real necessidade do aluno.

Em ambientes CSCW (ambientes colaborativos de trabalho) o tutor pode servir para orientar o grupo na organização dos conceitos responsáveis pela construção do conhecimento.

O ambiente Javal emprega um agente tutor simples, que, após ter sido acionado pelo agente monitor quando detecta uma situação de falha (por exemplo o *timeout* de um item de uma questão), exibe uma tela para o aluno, conforme figura 3.5.

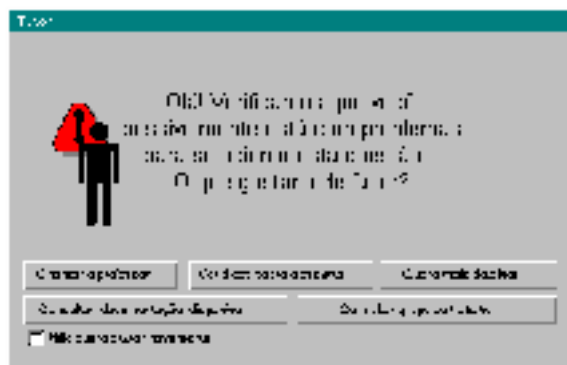


FIGURA 3.5 - Uma interface gráfica de agente tutor especialista em tempo (detecção de timeout).

As opções a serem disponibilizadas ao aluno são ajustadas pelo professor, durante a elaboração da avaliação, de forma que o tutor apenas possa disponibilizar os recursos julgados realmente necessários para atender ao aluno na questão considerada.

As ações possíveis de execução pelo tutor são:

- Chamar o professor - permite que o professor seja acionado imediatamente em situação de dúvida do aluno, entretanto, este recurso estaria disponível apenas para ambientes de avaliação síncrona<sup>3</sup>;
- Ver dicas sobre o assunto - permite o acesso a dicas disponibilizadas pelo professor durante a elaboração da avaliação;
- Quero mais detalhes - o aluno pode acessar um conteúdo mais detalhado para a solução da questão, quando disponibilizado pelo professor;
- Consultar documentação disponível - o professor pode disponibilizar documentos on line para consulta ou outros recursos possíveis como áudio, vídeo, etc.;
- Consultar grupo de trabalho - esta opção permite a integração entre elementos de um grupo de trabalho. O aluno pode questionar ao grupo através do acesso a uma lista de discussão on line ou sala de chat.

Caso o aluno manifeste um comportamento que possa indicar situação de dúvida, como a marcação e alteração de respostas, o agente entrará em ação, apoiando-o, visando provocar o insight.

Isto pode ser feito através da exibição de uma interface gráfica, conforme figura 3.6. Esta interface oferece ao aluno as seguintes funcionalidades:

<sup>3</sup> O autor deste trabalho classificou a avaliação quanto a execução em síncrona e assíncrona. Na primeira, todos os participantes iniciam e encerram ao mesmo tempo, enquanto que, na segunda, o aluno escolhe a data e hora de início. Em ambos os casos o tempo de duração da avaliação considerado é o mesmo.

- Chamar o professor, permite o aluno um contato direto com o professor, o que pode ser feito por chat ou e-mail;
- A opção “sim, estou certo” marca a resposta como certa e inicializa o agente correção para enviar a solução ao professor e emitir o *feedback* ao aluno;
- A opção “não tenho certeza” inicializa um agente especialista, que buscará no banco de dados (através de métodos embutidos no agente local) quais tópicos relativos àquele assunto foram empregados pelo professor para montar aquela questão. Os tópicos (dicas) serão apenas lembrados, não sendo exibidos seus conteúdos.



FIGURA 3.6 - Uma interface gráfica de agente tutor especialista em evento (detecção de dúvida).

- A opção “gostaria de mais detalhes” permite a exibição de um detalhamento da questão oferecida ao aluno. Neste caso, cabe ao professor elaborar um enunciado alternativo, para estas situações, com base no perfil do aluno avaliado; e
- A opção “consultar grupo de trabalho” permite consultar grupo de trabalho para buscar a solução da questão (disponível apenas em situações de avaliação em grupo, realizadas de forma síncrona).

Para trabalhos futuros nesta área, sugere-se que, os tutores devam ter linguagem e uma interface gráfica próprios, com cores e botões que devem estar ligados ao tipo de perfil (visando manter a harmonia do ambiente de ensino, tornando-o mais agradável ao aluno), devendo-se levar em conta aspectos metodológicos e psicológicos, visando permitir a elaboração de uma interface capaz de realizar uma perfeita interação homem-máquina.

Os agentes devem ser capazes de identificar características do aluno em questão, como o nome, sexo, grau de instrução, preferências, forma de tratamento, etc. Estas informações podem ser retiradas de uma base de dados de perfis desenvolvida especificamente para este fim, visando o desenvolvimento de agentes tutores (animados ou não) com características de personalidade. Com base neste contexto, os tutores devem ser capazes de modificar seu próprio aspecto e ambiente, em função destas informações.

Cabe ao sistema o desenvolvimento de personalização de ambientes de tutor, mas seu comportamento ante as dificuldades encontradas pelo aluno deve ser definido pelo professor.

### 3.4.2 O agente monitor

O monitor é um tipo de agente cujo objetivo é acompanhar a atividade do aluno, sendo inicializado durante a avaliação e cuja principal atividade é acionar os tutores quando necessário.

Monitores especializados para ambiente em grupo (colaborativo) podem permitir a interconexão entre tutores, visando criar um ambiente de avaliação multiagentes integrado.

Algumas das características desejáveis para o monitor e incorporadas ao ambiente:

- ⊗ Capacidade de manipular eventos;
- ⊗ Baseado em threads; e
- ⊗ Instruído pelo professor sobre como e quando atuar no ambiente aluno.

O monitor é capaz de tratar eventos da interface gráfica da questão (cliques de mouse, entrada e saída de áreas da tela, minimização de janelas, etc.), lembrando que alguns comportamentos podem evidenciar situações de dúvida do aluno.

Funcionalmente os monitores executam em threads e, por serem baseados em objetos, possuem atributos que definem ações e métodos capazes de definir comportamentos. O emprego do modelo de orientação a objetos permite que recursos como a herança e agregação possibilitem a criação de agentes monitores especialistas.

Durante a elaboração das avaliações no Módulo de Montagem de Avaliações (MMA), o professor será capaz de instruir o monitor sobre como se comportar em situações de falha, bem como este deve localizar e chamar os tutores.

## 3.5 Os agentes do ambiente de avaliação Javal

Os agentes desenvolvidos para uso no Javal tem por atividades específicas acompanhar a interação homem-máquina, no sentido de se aplicar prazos para o cumprimento das tarefas e prover o sistema de uma lógica capaz de ajustar-se ao perfil do aluno de acordo com o andamento do ensino.

Para este modelo, são propostos inicialmente os seguintes agentes:

- Agente Tempo da Avaliação (ATA) - monitoração de tempo de execução de cada atividade, *timeouts* e encerramento da avaliação; e
- Agente Comportamento da Avaliação (ACA) - monitoração de áreas da interface gráfica e eventos do sistema.

O agente de tempo é um tipo de agente cuja finalidade principal é detectar o *timeout* de uma atividade ou tarefa..

O ambiente Javal considera dois tempos distintos para monitoração. O tempo global, que envolve o tempo total de execução da avaliação e o tempo local, ligado ao tempo de execução de cada item da avaliação.

Este agente tem sua funcionalidade baseada em *threads*, sendo inicializado sempre que o aluno inicia uma determinada avaliação no sistema. Ele permite verificar se o aluno encontra-se desenvolvendo suas atividades dentro do prazo especificado para tal, de acordo com o estabelecido como meta durante a elaboração dos objetivos de ensino e o perfil disponível.

Quando o aluno extrapola o prazo previsto (*timeout*), o agente falha e notifica o agente comportamento, que tem por principal atividade a elaboração de relatórios sobre a avaliação.

### **3.6 Modelo de Objetos Distribuídos como base para avaliação em EAD e CSCW**

O ambiente Javal é concebido no ambiente de programação Orientada a Objetos, especificamente objetos distribuídos. Neste conceito, a linguagem Java [EMI00] foi escolhida como base para elaboração de protótipo, conforme descrito no capítulo 4, graças a suas características de segurança, portabilidade e neutralidade, além de oferecer considerável suporte a multitarefa, base imprescindível do trabalho dos agentes

Com relação ao modelo de objetos distribuídos, segundo [ALB98], este modelo oferece as seguintes vantagens:

- Aproveitamento de código, trazendo desenvolvimento rápido de aplicações (*Rapid Application Development - RAD*) com interação *plug-and-play* entre os objetos distribuídos;
- Manutenção efetiva de código incluindo aumento de código e distribuição sistemática de atualizações;
- Interfaces cliente mais leves (*thin*) que se conectam a aplicações servidoras e repositórios de dados em múltiplas localidades.

Este conjunto de vantagens serviu de base para a escolha da linguagem Java para desenvolvimento deste modelo.

### **3.7 Resumo do capítulo três**

O capítulo três apresentou aspectos motivacionais relacionados à necessidade de desenvolvimento de um ambiente informatizado capaz de suprir necessidades de acompanhamento de avaliações à distância, face à dificuldade de controle que lhe é peculiar.

Foram explorados os conceitos de agentes e, em especial, de tutores embarcados, cuja finalidade é conduzir o aluno a uma situação de insight que lhe seja favorável para a solução da avaliação.

Em seguida foram apresentados os modelos e funcionalidades desejadas para dois agentes específicos, tempo e comportamento, bem como sugeridos novos agentes como valida perfil e seleciona recurso de avaliação, visando a exibição de enunciados alternativos ao aluno durante a resolução de provas.

Por último foram apresentados aspectos ligados à tecnologia de objetos, base para o desenvolvimento deste trabalho.

## 4 O modelo funcional do ambiente Javal

Neste capítulo será apresentada uma descrição geral do ambiente de avaliação Javal. Para validação do modelo de agentes embarcados proposto, foi implementado apenas o Módulo de Avaliação Remota (MAR), cujas classes serão empregadas como ambiente de interface do aluno, do sistema SEMEAI. A descrição da implementação destas classes está disponível no capítulo 5.

O modelo proposto para o ambiente Javal oferece um pacote de classes voltadas para a criação de um ambiente de avaliação remota, empregando a tecnologia de orientação a objetos, com classes escritas em Java. A composição dos módulos e classes do sistema será apresentada da seguinte forma:

Inicialmente, serão apresentados os módulos componentes do ambiente Javal e suas funcionalidades básicas, visando situar o protótipo implementado em um contexto mais amplo. Em seguida, será feita a descrição conceitual e funcional da API de classes que o compõem.

Não é interesse deste trabalho esgotar todos os recursos capazes de serem implementados neste ambiente, sendo sua API disponibilizada para a personalização e criação de novos elementos visando incrementar este ambiente, tornando-o flexível, customizável e escalável.

### 4.1 Composição do ambiente

O modelo do ambiente Javal (figura 4.1) é concebido por módulos que englobam um conjunto de classes de objetos, visando assegurar modularidade e independência entre eles, de forma a torná-lo uma estrutura portátil, segura e escalável.

Para validar o modelo de avaliação proposto, sua implementação abrangerá apenas o Módulo de Avaliação Remota (MAR) do ambiente Javal. Visando emoldurá-lo no contexto de um ambiente de avaliação para Ensino à Distância, serão feitas descrições sobre o projeto do ambiente Javal.

Os módulos componentes do ambiente Javal são os seguintes:

**Módulo de Controle de Avaliações (MCA)** - Responsável pelo controle geral do ambiente Javal. Localizado no servidor, coordena as atividades de comunicação, autenticação e segurança, elaboração de avaliação, armazenamento e distribuição dos Módulos de Avaliação Remota (MAR) aos alunos, sobre uma estrutura de rede ou Internet.

**Módulos de Avaliação Remota (MAR)** - unidades de avaliação remota distribuídas aos alunos através do ambiente de rede, empregando o recurso de applets em páginas Web. Os MAR incorporam agentes responsáveis pelo acompanhamento do aluno durante a execução da avaliação.

**Módulo de Montagem de Avaliações (MMA)** - Responsável por apresentar ao professor uma interface gráfica para desenvolvimento do seu ambiente de avaliação. Permite a criação de avaliações e suas questões, bem como de toda a sua estrutura de pontuação, disponibilização de recursos e configuração de comportamento de agentes. Como baseia-se em ambiente Web, dispensa qualquer tipo de instalação de software no lado do professor, facilitando a criação de avaliações em qualquer computador da rede ou da Internet.

**Módulo de Armazenamento de Dados (MAD)** - Consiste de um banco de dados

que tem por finalidade armazenar avaliações, informações para agentes, perfis, dicionários, relatórios de avaliação e questões, bem como prover a infraestrutura necessária para a criação de objetos avaliação e armazenamento.

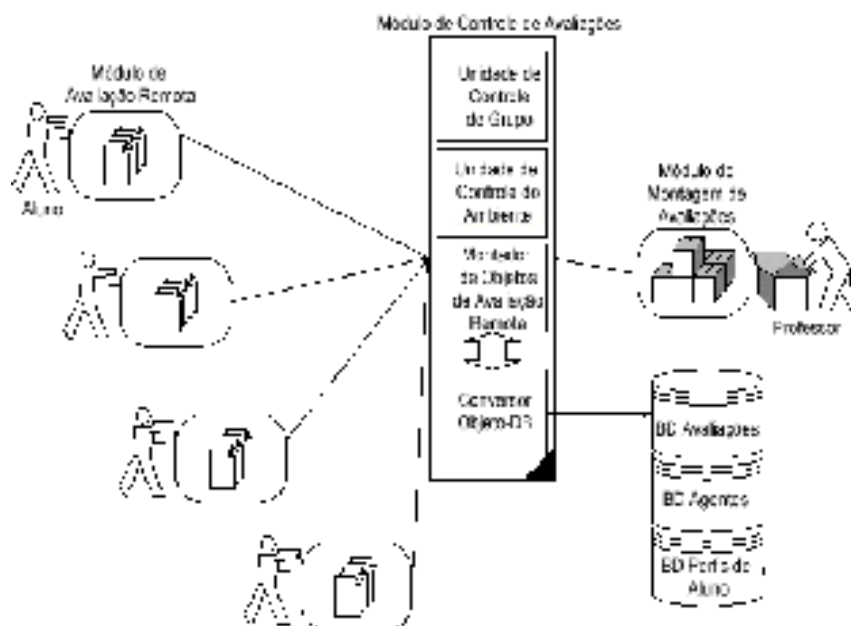


FIGURA 4.1 - O ambiente proposto pelo modelo Javal.

Os módulos envolvem um conjunto de classes organizadas em pacotes de acordo com suas funcionalidades, conforme descrito no capítulo 5.

O modelo conceitual do ambiente Javal considera uma série de entradas e saídas de informação, sendo capaz de tratar cada uma com seus módulos. Para uma análise de suas funcionalidades básicas, o modelo do sistema foi desenvolvido para interagir com dois tipos de atores definidos: professor e aluno.

As interações com o professor envolvem aspectos operacionais do sistema e pedagógicos da avaliação, enquanto que as interações com o aluno envolvem aspectos cognitivos e comportamentais.

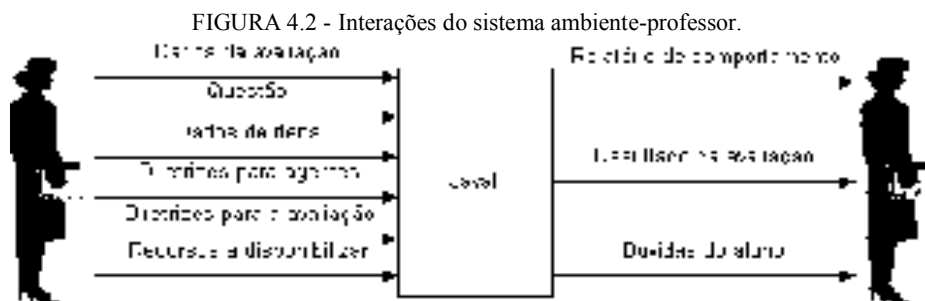
As interações que o professor realiza com o sistema (figura 4.2), envolvem os seguintes fluxos de informação:

- A avaliação;
- Diretrizes operacionais;
- Diretrizes temporais;
- Diretrizes comportamentais;
- Orientações para execução; e
- Recursos a disponibilizar.

A avaliação - compreende o objeto de avaliação propriamente dita, com informações relativas à instituição, professor, turma de aula, matéria, assunto, disciplina, valor das questões e tudo que diz respeito à avaliação, com seus aspectos didáticos e formais.

Diretrizes operacionais - compreende informações que permitem instruir os alunos e agentes específicos para a execução da avaliação. Englobam a definição da

avaliação quanto a sua execução (individual ou em grupo), o tempo destinado à sua realização, a data de início e de término, a ativação do recurso de recolhimento automático da avaliação, tutores para os itens da questão e disponibilização ou não de recursos durante a execução.



**Diretrizes temporais** - compreende aspectos ligados aos agentes que acompanharão o aluno durante a execução da avaliação, analisando o componente tempo de execução. Permite estabelecer tempo máximo e mínimo para a solução de cada item, dentro do qual os agentes monitoram situações de demora do aluno e que desencadeiam procedimentos de auxílio, dentro do estabelecido pelas diretrizes operacionais.

**Diretrizes comportamentais** - contempla aspectos comportamentais relacionados aos sensores que estarão presentes no ambiente do aluno, especialmente na interface gráfica. Minimizações de janela, modificação de respostas são aspectos monitorados e, caso ocorram fora da margem prevista pelo professor, desencadeiam procedimentos de auxílio ao aluno, dentro das diretrizes operacionais.

**Orientações para a execução** - compreende instruções sobre a execução da prova, que o professor pode disponibilizar para o aluno, orientando-o sobre como proceder durante sua realização.

**Recursos a disponibilizar** - compreende o conjunto de recursos que o professor pode disponibilizar para que o aluno os consulte, em situação de dúvida, durante a avaliação. Envolve páginas da web, serviços de comunicação em grupo, lista de discussão, sala de chat, página web da disciplina, arquivos de áudio, vídeo, imagens, etc.

O sistema disponibilizará ao professor uma série de informações, as quais envolvem :

- Relatórios comportamentais;
- Dúvidas ocorridas; e
- Resultado da avaliação.

**Relatórios comportamentais** - visam apresentar ao professor aspectos ligados ao comportamento do aluno frente a cada item que lhe foi apresentado, como questões “puladas” ou resolvidas fora do tempo previsto, uso de recursos disponibilizados, e outras informações necessárias para que o professor adeque recursos ou tempo ao tipo de perfil considerado, em avaliações futuras.

**Dúvidas ocorridas** - envolve o conjunto de questões cuja resposta não ocorreu dentro do limite de tempo previsto e que, para sua solução, o aluno tenha sido obrigado a consultar recursos disponibilizados pelo professor. São informações importantes para um reestudo sobre a elaboração de questões ou disponibilização de recursos pelo professor.

Resultado da avaliação - envolve os aspectos formais da avaliação, informando o grau obtido pelo aluno como fruto da realização da questão.

As interações do sistema com o aluno visam o monitoramento dos seus aspectos comportamentais, apresentar informações disponibilizadas pelo professor, avisos e alertas do sistema, além da avaliação e seus componentes.

Estas interações englobam aspectos temporais e comportamentais, como interações com a interface gráfica e solução de itens.

As principais interações do aluno com o sistema são:

- Ajustes do ambiente;
- Seleção de recursos;
- Seleção de questões e itens;
- Eventos de interface gráfica;
- Elaboração de respostas; e
- Tempo de solução.

Ajustes do ambiente - o aluno pode interagir com o ambiente da avaliação ativando recursos capazes de facilitar a tarefa de sua execução, como tutores, alarme de fim da avaliação, etc.

Seleção de recursos - envolve as ações que permitem ao aluno a solução de questão pela consulta a informações disponibilizadas previamente pelo professor, para o item considerado.

Seleção de questões e itens - compreende a seleção das questões e de seus itens para serem respondidos.

Eventos de interface gráfica - compreende os eventos que o aluno desencadeia ao interagir com a interface gráfica do ambiente da avaliação, como cliques de mouse, entradas e saídas do ambiente, minimização de janelas, clique em botões, etc. Estes são monitorados por agentes específicos que podem desencadear processos de auxílio, quando preciso.

Elaboração de respostas - envolve as respostas elaboradas pelo aluno, com o objetivo de obtenção de conceitos e avaliação formal.

Tempo de solução - envolve o tempo gasto pelo aluno na solução dos itens da avaliação.

Na interação sistema-aluno (figura 4.3), o ambiente disponibiliza informações e recursos úteis para a execução da avaliação.

Estas informações estão ligadas a:

- Orientações;
- Correção;
- Tutores;
- Recursos;
- Questões e itens; e
- Alertas.



Orientações - orientações disponibilizadas pelo professor para elaboração da avaliação.

Correção - as questões são corrigidas e o resultado da correção é apresentada ao aluno, sob forma de correção da aprendizagem.

Tutores - envolvem a interface gráfica dos tutores que o acompanharão durante a avaliação.

Recursos - os recursos disponibilizados pelo professor.

Questões e itens - os elementos componentes da avaliação.

Alertas - alertas e mensagens de erro do sistema para o aluno, servindo-lhe como orientação para utilização do ambiente ou para reportar-lhe erros de execução da aplicação.

Os módulos componentes do ambiente interface do aluno, tratam especificamente dos aspectos relacionados com a interação ambiente-aluno, base deste trabalho.

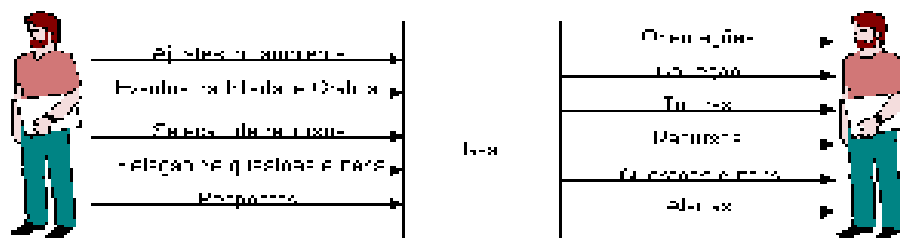


FIGURA 4.3. - Interações do sistema ambiente-aluno.

#### 4.1.1 Módulo de Controle de Avaliações (MCA)

Embora a implementação do modelo relativo ao MCA seja objeto do presente trabalho, será feita uma breve descrição de suas funcionalidades de forma a permitir o entendimento da dinâmica do sistema e da interação necessária entre o ambiente de avaliação e o ambiente do aluno.

O MCA é o módulo responsável pelo controle do ambiente de avaliação. Localizado no servidor, é acessível via Web tanto pelos alunos, para a execução das avaliações, quanto pelo professor, para sua montagem e verificação.

Este módulo tem por finalidades:

- ⊗ Permitir ao professor a criação de avaliações, a instrução de agentes e a seleção de recursos a serem empregados pelos mesmos, pelo emprego do Módulo de Montagem de Avaliações (MMA);
- ⊗ Efetuar autenticações e validações de acesso pelo aluno e professor;
- ⊗ Controlar a comunicação do sistema servidor com os Módulos de Avaliação Remota (MAR);
- ⊗ Monitorar o trabalho dos agentes embarcados, sob forma de obter informações necessárias ao acompanhamento das avaliações e manipulação de perfis, quando necessário;
- ⊗ Permitir comunicação entre os MAR, quando for desejável para o professor (atividades em grupo, por exemplo); e

- ⑩ Armazenar objetos em uma base de dados e sua fácil recuperação quando desejado.

Para exercer estas funções, o MCA é basicamente composto pelos seguintes unidades:

- ⑩ Unidade de Controle de Grupo (UCG) - Permite o controle pelo professor das atividades de grupo, bem como a comunicação entre os participantes e o ambiente ou os participantes entre si. Encapsula todos os aspectos relacionados à nomeação de grupos, à identificação e ao gerenciamento de recursos no ambiente;
- ⑩ Unidade de Controle de Ambiente (UCA) - Controla o ambiente de avaliação, permitindo o agendamento de avaliações, distribuição, recolhimento (manual ou automático), geração de relatórios de acesso ao sistema, controle de senhas e serviço de suporte de comunicação para o sistema;
- ⑩ Montador de Objetos de Avaliação Remota (MOA) - Permite a instanciação dos objetos que irão compor a avaliação descarregada para o ambiente do aluno, após sua solicitação ao sistema. Possui recursos que permitem sua conexão ao banco de dados através do conversor de objetos/banco de dados e a recuperação dos dados necessários para montagem destes objetos; e
- ⑩ Conversor de objetos/banco de dados (COD) - Permite a conversão dinâmica de objetos em banco de dados relacional e vice-versa, sem perda de suas características operacionais. Juntamente com o Módulo de Armazenamento de Dados (MAD), permite armazenamento persistente das informações. Espera-se com o uso de banco de dados orientado a objeto que este conversor simplesmente realize operações de armazenamento e busca de objetos.

#### **4.1.2 Módulo de Avaliação Remota (MAR)**

Será feita a descrição do Módulo de Avaliação Remota (MAR), objeto do presente trabalho. Suas características permitem o trabalho dos agentes e tutores vistos no capítulo 3.

O MAR (figura 4.4) compõe o módulo que será distribuído pela rede aos participantes da avaliação. Para este modelo é proposto o uso de uma applet, devido às suas características de modularidade e seus aspectos de segurança [EMI00], além de permitir o emprego de múltiplas threads, necessárias ao funcionamento dos agentes nele embarcados.

Será apresentada no capítulo 5 a implementação do MAR.

Diferentemente dos diversos sistemas comerciais de avaliação à distância analisados, cuja lógica de funcionamento é baseada em scripts existentes no servidor, o ambiente Javal embarca toda sua lógica de operação no ambiente do aluno, através do uso de classes, executadas sobre a plataforma proporcionada pela applet.

O MAR possui as seguintes funcionalidades:

- Carregar em um Repositório de Objetos Locais (ROL) o objeto avaliação e seus objetos componentes;
- Permitir a instanciação de objetos agente no ambiente do aluno, com suas diretrizes operacionais estipuladas pelo professor;

- Informar ao aluno a composição da avaliação, permitindo ao aluno flexibilizar a realização da avaliação, facultando-lhe escolher por qual questão deseja iniciar, de forma aleatória;

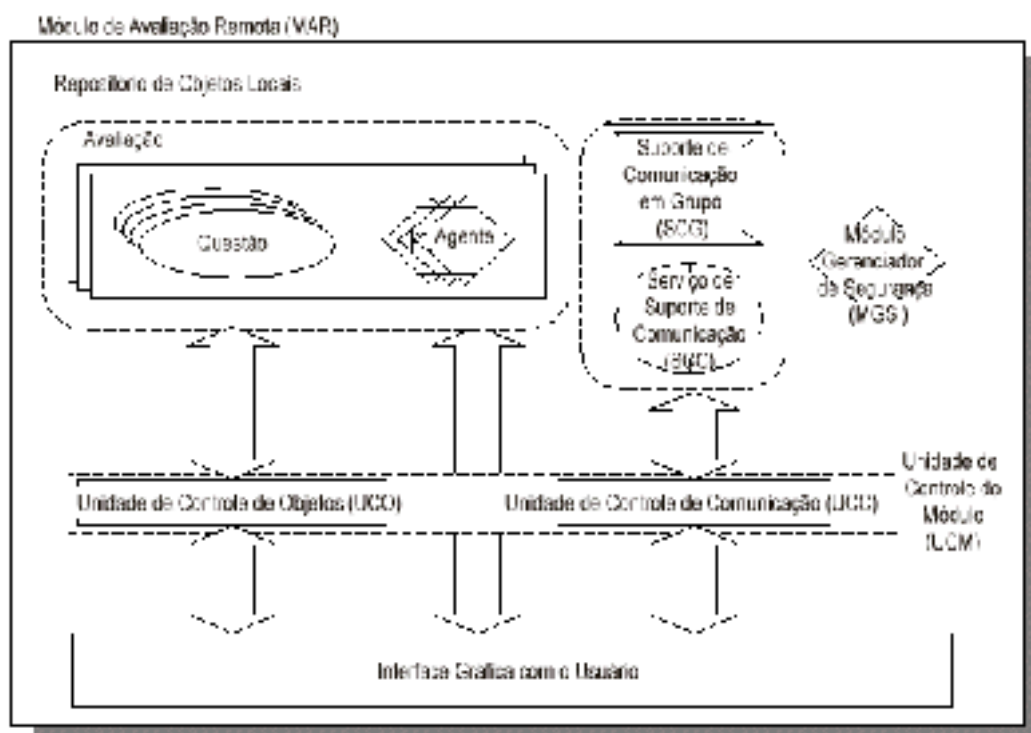


FIGURA 4.4 - Descrição em blocos do Módulo de Avaliação Remota (MAR), empregado pelo Javal.

- Permitir a ativação de alarme da avaliação, visando informar ao aluno com antecedência do término previsto da avaliação;
- Permitir ao aluno optar ou não pelo auxílio do agente tutor durante a avaliação;
- Permitir ao professor obter informações e relatórios sobre a realização de sua avaliação, especialmente quanto a aspectos comportamentais do aluno, através da monitoração realizada pelo Agente Comportamento; e
- Permitir ao aluno acessar recursos disponibilizados pelo agente, bem como interagir com outros elementos do grupo, quando autorizado pelo professor.

Para isto, o MAR é composto pelos seguintes elementos:

- Repositório de Objetos Locais (ROL);
- Unidade de Suporte de Comunicação (USC);
- Módulo Gerenciador de Segurança (MGS); e
- Unidade de Controle do Módulo (UCM).

#### 4.1.2.1 Repositório de Objetos Locais (ROL)

Estrutura dentro da Applet que garante armazenamento em memória dos objetos avaliação e seus componentes recebidos via rede. Está localizado dentro da Unidade de Controle do Módulo (UCM).

Durante a carga do MAR, a UCM recebe objetos descarregados do MCA e possui métodos que permitem selecionar e exibir o objeto selecionado.

Está elaborado sob forma de um objeto do tipo *Vector* (*java.util.Vector*), permitindo ao aluno o acesso aleatório a qualquer tipo de questão que deseje realizar.

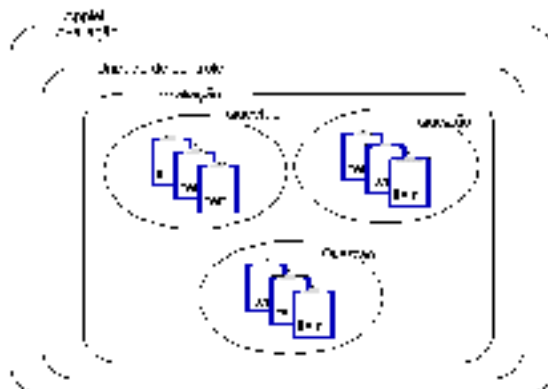


FIGURA 4.5 - A estrutura de Repositório de Objetos Locais (ROL) e sua hierarquia de armazenamento.

Os objetos do tipo questão e item possuem seus próprios repositórios locais de objetos, garantindo o armazenamento de instâncias dos elementos que agregam, segundo a hierarquia de armazenamento do sistema (figura 4.5).

Desta forma, uma avaliação pode ser composta por uma ou mais questões que, por sua vez, podem agregar um ou mais itens, oferecendo flexibilidade na criação e apresentação de questões.

#### 4.1.2.2 Unidade de Suporte de Comunicação (USC)

Unidade responsável pelo estabelecimento de comunicações do módulo MAR com o Módulo de controle da Avaliação (MCA) e outros módulos, visando assegurar o envio e recebimento de informações.

É através do USC que o sistema conecta-se ao MAD, no servidor, visando buscar os objetos para armazenamento no ROL.

É formado por :

- Suporte de comunicação de grupo (SCG); e
- Serviço de suporte à comunicação (SSC).

O suporte de comunicação de grupo (SCG) é um módulo que permite a execução de atividades em grupo. Permite a comunicação entre alunos, através da disponibilização de uma interface gráfica específica chamada Interface de Comunicação de Grupo (ICG), que oferece um ambiente semelhante a um chat (figura 4.6). O ambiente permite a participação do professor de quaisquer das salas dos diversos grupos, de forma que possa acompanhar o andamento das conversações e intervir quando necessário.

O serviço de suporte de comunicação (SSC) é composto por uma série de classes capazes de prover a infraestrutura de comunicação necessária para o sistema. A plataforma de comunicação (RMI, CORBA) não foi definida neste trabalho, mas estudos e protótipos estão sendo desenvolvidos para serem incorporados em trabalhos futuros.



FIGURA 4.6 - Aspecto da Interface de comunicação de grupo (ICG) .

#### 4.1.2.3 Unidade de Controle de Objetos (UCO)

A Unidade de Controle de Objetos é o elemento responsável por controlar a instanciação dos diversos objetos existentes no ROL, quando solicitados através da GUI. Controla o funcionamento das threads dos agentes, bem como destrói objetos avaliação por solicitação do Agente Tempo ou do professor.

Adicionalmente, permite o controle sobre os objetos da interface gráfica, sob forma de evitar múltiplas instâncias da mesma janela.

#### 4.1.2.4 Unidade de Controle do Módulo (UCM)

Tem por finalidade controlar o funcionamento geral do MAR, envolvendo atividades de manipulação de objetos, segurança e controle de dispositivos de comunicação.

Suas principais funções são:

- Permitir a instanciação local da avaliação;
- Inicializar o agente de tempo da avaliação;
- Controlar o armazenamento do objeto avaliação no ROL, com suas questões e agentes;
- Destruir o objeto avaliação quando solicitado pelo agente ou determinado pelo professor;
- Instanciar elementos da interface gráfica, através do UCO;
- Instanciar objetos de suporte à comunicação em grupo, disponíveis no USC;
- Identificar políticas de segurança implementadas pelos navegadores do ambiente aluno, respeitando e impondo o respeito à caixa de areia de execução;
- Bloquear a instanciação de objetos, quando encerrada a avaliação;
- Permitir a comunicação inter-agentes, visando uma ação conjunta e dinâmica;

- Controlar o funcionamento da interface gráfica da avaliação com o aluno; e
- Tratar exceções ocorridas no sistema e enviar ao ACM os problemas detectados ligados à avaliação e suporte de comunicação.

### **4.1.3 Módulo de Montagem e Acompanhamento de Avaliações (MMA)**

O MMA é o módulo que permite ao professor a criação dos objetos questões a serem disponibilizadas para o aluno, bem como o controle sobre o ambiente da avaliação e sobre os agentes destinados ao trabalho junto ao aluno, durante a execução de sua avaliação.

A monitoração pode ocorrer de forma síncrona ou assíncrona:

- Para avaliações síncronas, a monitoração pode ser feita on-line através do Console de Monitoração de Avaliação (CMA), disponível no ambiente; e
- Para avaliações assíncronas, os agentes podem ser instruídos pelo professor para direcionar informações para o banco de dados do sistema ou para um serviço de correio eletrônico.

Para a montagem da avaliação, suas principais funções são:

- Criar questões do tipo Verdadeiro-Falso, Múltipla Escolha, Ordenação, Responda, Preenchimento de Lacunas e Associação;
- Permitir definições de data de início e término, bem como tempo de duração total da avaliação e tipo de recolhimento (síncrono ou destruição automática);
- Definir avisos e instruções pertinentes à avaliação, a serem exibidos por ocasião do carregamento da mesma pelos alunos;
- Definir mensagens de motivação ou diálogos entre o agente e o aluno (quando não for escolhido o agente padrão);
- Permitir ao professor instruir o agente comportamento quanto à sua atuação diante das ações realizadas pelo aluno;
- Indicar para o agente os recursos a serem empregados pelo aluno, em situação de necessidade; e
- Permitir o recebimento de solicitações enviadas pelos agentes, ligados a possíveis modificações nos perfis dos alunos.

Para esta dissertação, não está prevista a implementação do MMA, devendo servir como sugestão para trabalhos futuros.

### **4.1.4 Módulo de Armazenamento de Dados (MAD)**

Consiste em um serviço de banco de dados SQL, podendo estar executando no mesmo equipamento (com acesso via ODBC) ou em um sistema remoto (utilizando serviço de conexão provido pelo banco de dados).

Compõe-se de tabelas capazes de armazenar os objetos instanciados pelo professor, durante a montagem da avaliação no MMA. Para a validação do modelo da Interface do Aluno, foi empregado um banco de dados MySQL.

Visando permitir a conversação entre as Applets e o MCA, foi desenvolvida e

encontra-se em testes uma aplicação chamada RAT (*Remote Attendant*), cuja finalidade é oferecer uma interface de comunicação entre applets e o banco de dados (local ou remoto), contornando problemas de segurança localizados no emprego de applets em ambiente Internet Explorer 5.0 e superiores.

## **4.2 Descrição funcional do Módulo de Avaliação Remota (MAR)**

O MAR consiste de uma applet que é descarregada no ambiente do aluno. Seus objetos são instâncias de classes organizadas em pacotes, com funcionalidades, atributos e métodos específicos para cada módulo.

Uma descrição sumária sobre seu funcionamento é feito conforme o fluxograma descritivo da figura 4.7 e compreende os seguintes passos:

1. O MAR é descarregado para o ambiente do aluno, e, após sua validação, instancia o objeto avaliação e suas questões;
2. Informações relativas à avaliação, como tempo de duração, valor total da avaliação e as questões disponíveis são exibidas ao aluno. O aluno pode optar pelo uso ou não de tutores durante a avaliação, e, adicionalmente, pode ajustar o timer do sistema, visando alertá-lo da proximidade do final da prova;
3. As questões são objetos que podem ser escolhidos pelo aluno de forma aleatória, e, uma vez escolhidos, apresentam ao aluno a interface gráfica correspondente;
4. Uma interface gráfica específica (Navegador de Questões) permite ao aluno escolher qual item da questão escolhida deseja resolver, com base no enunciado que lhe é exibido;
5. O agente responsável por monitorar o tempo de execução da avaliação é inicializado, contando o tempo para sua solução, e alertando o aluno quanto à proximidade do seu final;
6. Uma vez escolhido o item, o objeto é instanciado e a interface gráfica correspondente ao tipo é exibida;
7. Neste instante, são inicializados agentes que controlam o tempo de execução do item e o monitor do comportamento do aluno;
8. Enquanto o item respondido está dentro do tempo previsto pelo professor, o Agente Comportamento monitora a correção de questões, as modificações na resposta, as minimizações de janela, de forma que, caso o agente registre o número de modificações ou interações acima do previsto pelo professor, seja inicializado um agente tutor, para auxiliá-lo na solução da questão.

Os eventos considerados para o agente Comportamento são:

- Interações com a resposta - o aluno elaborou a resposta e a modificou antes de enviá-la;
- Minimizações da janela da questão, sem envio da resposta - o aluno pode minimizar janelas de questões, para respondê-las em seguida; e
- Respostas precipitadas - o aluno responde o item apressadamente, abaixo do tempo mínimo de solução proposto pelo professor;

9. Neste momento, o sistema apresenta ao aluno um agente tutor, cujo objetivo é a representação do professor junto ao aluno, visando auxiliá-lo na solução da questão. Quando o aluno envia a resposta ao sistema, o agente Comportamento gera um objeto do tipo relatório, contendo o resumo das atividades ocorridas durante a execução do item;
10. Ao atingir o tempo limite para a execução do item previsto pelo professor, o sistema aciona um agente tutor visando auxiliar o aluno na sua execução. Este agente é inicializado, permitindo ao aluno o acesso aos recursos disponibilizados pelo professor, durante a montagem da avaliação;
11. O agente tutor recebe instruções do professor sobre quais tipos de recursos devem ser disponibilizados para o aluno nas situações de falha durante a avaliação. Estas instruções são passadas pelo professor ao agente, durante a elaboração do item, ao montar a avaliação, de forma que o mesmo saiba quais recursos devem ser disponibilizados, como dicas e recursos úteis, comunicação em grupo, páginas web, etc; e
12. Ao término do tempo previsto da avaliação, um objeto relatório é gerado e remetido ao MCA. O relatório traz informações úteis sobre o comportamento do aluno e sua performance durante a realização da avaliação, podendo servir ao professor para alterações na estratégia de avaliação ou de ensino adotados.

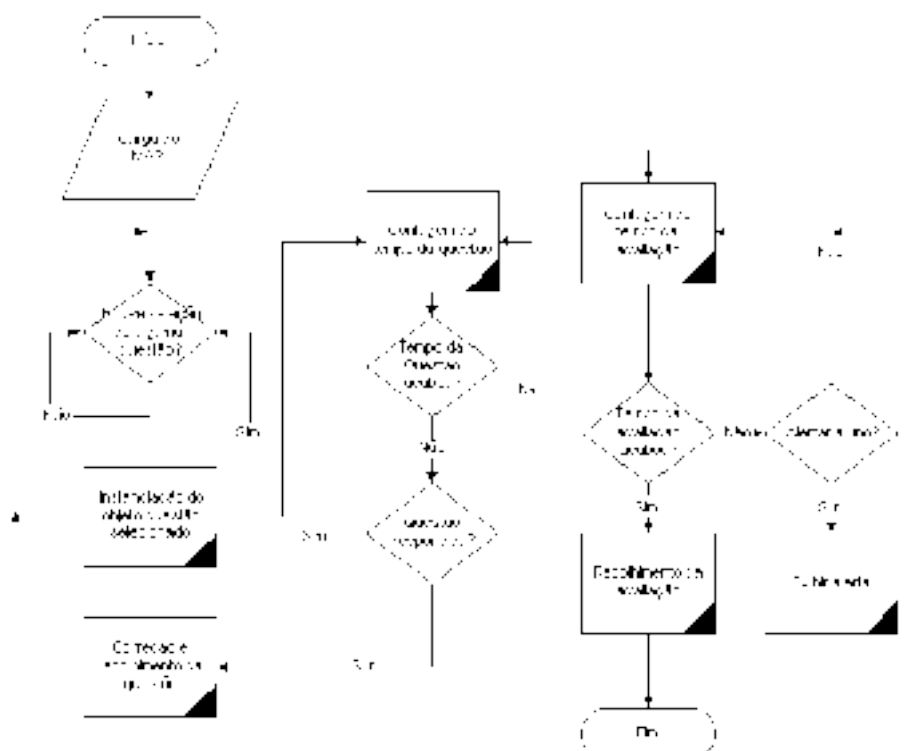


FIGURA 4.7 - Fluxograma de funcionamento do MAR..

O funcionamento do MAR envolve quatro estados básicos (figura 4.8):

1. Carga do ambiente e ajuste;
2. Execução da avaliação;
3. Avaliação do tempo da execução; e
4. Encerramento da avaliação.



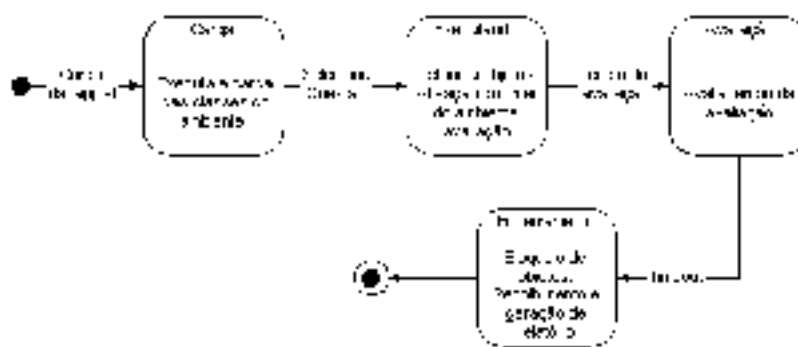


FIGURA 4.8 - Diagrama de estados de funcionamento do MAR.

### 4.2.1 Carga do ambiente e ajuste

Inicialmente, o MAR, que consiste em um applet embutido em uma página HTML, é descarregado para o ambiente do aluno e é inicializado. Ao término de sua carga, a interface gráfica (figura 4.9) será exibida ao aluno.

O aluno digitará suas informações de login e senha. Ao clicar no botão “Entrar”, será realizada a conexão com o banco de dados através do MCA, visando sua validação. O MCA então selecionará o objeto avaliação correspondente, que será serializado e exibido no MAR. .

A Unidade de Controle instanciará em seu ROL o objeto avaliação, e neste momento o aluno poderá:

- Consultar instruções relativas à execução da prova ;
- Habilitar o auxílio de agentes durante a avaliação; e
- Ajustar alerta de final de prova.

Consultar instruções relativas à execução da prova - procedimentos a serem tomados para a solução da prova ou outras informações adicionais podem ser fornecidas pelo professor e exibidas pelo sistema em uma janela específica ou no espaço destinado na applet.

Habilitar o auxílio de agentes durante a avaliação - o aluno poderá optar pelo uso ou não dos agentes designados pelo professor para auxílio durante a avaliação. Tais agentes são instruídos pelo professor para agir em situação de dificuldade do aluno durante a avaliação.

Ajustar alerta - o aluno poderá utilizar um objeto Alarme específico que será capaz de avisá-lo da proximidade do término da avaliação, emitindo um alerta visual quando este tempo for atingido.

Para iniciar a prova, o aluno deve escolher por qual tipo de questão deseja iniciar seu trabalho.

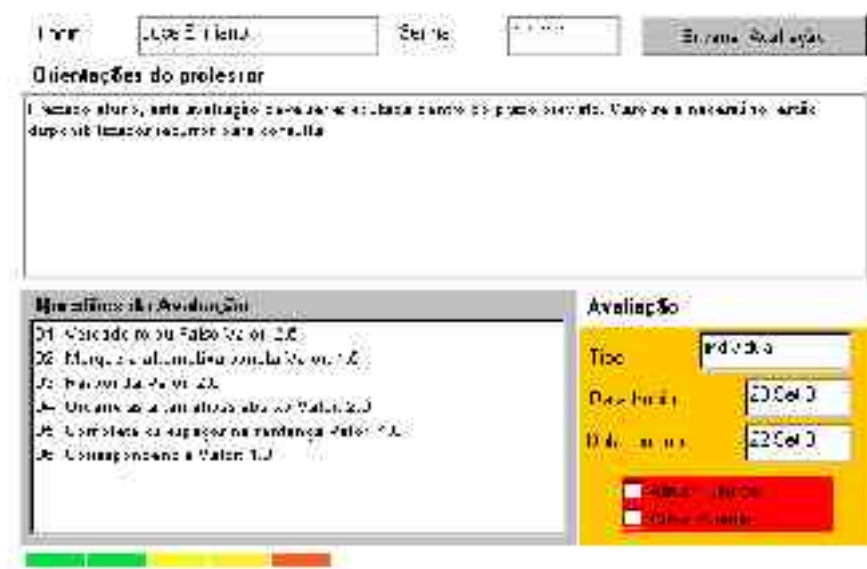


FIGURA 4.9 - Aspecto da interface gráfica do Módulo de Avaliação Remota (MAR).

As diversas ações e módulos envolvidos estão descritos na tabela 4.1:

TABELA 4.1 - Eventos da fase de carga e ajuste.

<i>Nr</i>	<i>Evento</i>	<i>Descrição</i>	<i>Obs.</i>
1	Descarga da applet avaliação do MCA.	A applet contida na página HTML é descarregada para o ambiente do aluno e inicializada.	
2	Logon no sistema.	O aluno preenche sua identificação (login, senha) e, ao clicar no botão “Entrar”, é realizada a carga da avaliação.	A Unidade de Controle do Módulo solicita a unidade ConnectDB a conexão com banco de dados e a busca da avaliação a ser entregue ao aluno.
3	Carga da avaliação.	A Unidade de Controle do Módulo instancia o objeto tipo avaliação, questões e itens.	A Unidade de Controle do Módulo efetua a chamada nos métodos dos objetos da Unidade de Suporte à Comunicação, visando a descarga dos elementos da avaliação.
4	Exibição.	Exibe na applet informações sobre a questão (orientações iniciais, informações sobre o tipo de avaliação e duração).	O aluno pode ajustar seu alarme de aviso da proximidade de término da avaliação.
5	Instanciação dos objetos no ROL.	A Unidade de Controle do Módulo instancia localmente a avaliação, armazenando-a no ROL. Posteriormente os objetos questões são instanciados e armazenados no VetorQuestao, o repositório existente no objeto Avaliacao.  Em seguida, os itens correspondentes são instanciados e armazenados dentro de cada questão, em seu repositório local.	A Unidade de Controle do Módulo chamará a carga da avaliação. Em seguida, as questões pertencentes àquela avaliação. Por fim, os itens pertencentes à cada questão.

## 4.2.2 Execução da avaliação

Após a carga da applet, o ambiente permanece aguardando o início da avaliação. Quando o aluno seleciona a questão desejada, a Unidade de Controle do Módulo instancia o agente especialista (AgenteTempo) que desencadeia um processo de contagem de tempo da avaliação, com base no tempo de execução previsto pelo professor.

A execução da avaliação é dividida em duas fases: seleção e solução do item.

- A fase de seleção envolve a seleção da questão a ser solucionada, com a exibição do navegador de questões correspondente; e
- A fase de solução do item tem início com a interação com o navegador de questões e instanciação dos objetos e interface gráfica do item desejado.

A sequência de eventos ocorridos durante esta fase pode ser vista mais adiante na figura 4.11. e estão relacionados na tabela 4.2.

O agente responsável pelo tempo (ATA) contará o tempo total da avaliação, definido pelo professor e calculado pelo somatório dos tempos previstos para cada questão do sistema. Estas definições do tempo de avaliação são elaboradas pelo módulo MMA, durante a montagem da avaliação.

### 4.2.2.1 Fase de seleção

Durante a fase de seleção, o aluno poderá escolher a questão desejada, de forma aleatória, sendo lhe apresentado a interface gráfica do Navegador de Questões (figura 4.10), que permite navegação aleatória dos itens componentes de cada questão.

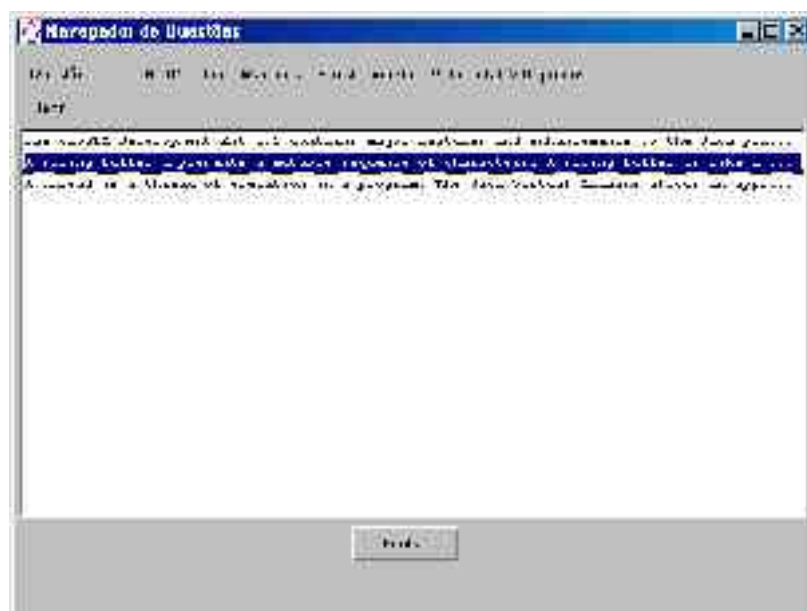


FIGURA 4.10 - Interface gráfica do Navegador de Questões.

O Navegador de Questões exibe informações sobre a composição da questão, valor e os enunciados parciais dos itens que o aluno pode escolher para solucionar. Ao clicar sobre o enunciado do item pretendido, o sistema exibe sua interface gráfica e entra na fase de solução do item.



TABELA 4.2 - Eventos ocorridos na fase de execução da avaliação.

<i>Nr</i>	<i>Evento</i>	<i>Descrição</i>	<i>Obs.</i>
1	Seleção	O aluno escolhe uma questão que deseja realizar. Ao clicar, a applet informa a Unidade de Controle do Módulo qual questão foi selecionada. O navegador de questões é exibido, contendo o enunciado de cada item da questão.  Ao clicar no item desejado, é iniciado a fase de solução.	A Unidade de Controle do Módulo inicializa a prova. O agente tempo de avaliação (ATA) é inicializado.
2	Solução do item	A interface gráfica do item selecionado é exibido. Os agentes de tempo e comportamento são inicializados com parâmetros de funcionamento definidos pelo professor.  A notificação das ações sobre a resposta é feita pela interface gráfica do item, sendo monitorados pelo agente comportamento.	Os agentes carregam informações sobre que recursos devem ser disponibilizados ao aluno em situação de falha.
3	Envio de Resposta	Se a questão foi respondida, ao clicar no botão “Enviar Resposta”, o objeto da interface gráfica em questão chama o método “corrigir ()”, que permite corrigir a questão, informando a quantidade de acertos.	Cada questão possui uma implementação distinta do método corrigir. A resposta é verificada e o relatório é gerado e armazenado em um objeto Relatorio.
4	Geração de relatório	Todo objeto que estenda a classe Item possui como agregado um objeto da classe Relatorio (ri – relatório do item). Este relatório incorpora informações sobre o comportamento do aluno durante a solução da questão, além de informações sobre acertos e comportamentos manifestados.	O objeto do tipo relatório possui um método Ler() que permite consultar todas as informações contidas em seu objeto.

### 4.2.3 Avaliação do tempo de execução

O módulo MAR baseia o funcionamento do agente tempo na análise de tempos gastos para a solução da avaliação. Para isto, foram considerados tempos para solução de cada item (tempo local), cujo somatório define o tempo de solução da avaliação (tempo global). Estes valores de tempo são monitorados por instâncias da classe Agente Tempo, que executam ações específicas, conforme descrito a seguir.

#### 4.2.3.1 Agente de tempo da avaliação global

O agente de tempo da avaliação global é um agente de tempo, instanciado com um construtor específico, responsável por verificar se o tempo de execução da avaliação está dentro do previsto e se o mesmo atingiu o tempo de alarme definido pelo o aluno.

Caso o tempo de execução da avaliação atinja o tempo previsto pelo o aluno para ativação do alarme, uma tela de aviso é exibida.

Caso o tempo global tenha esgotado, a avaliação é encerrada:

- O navegador de questões é bloqueado. Caso sua interface gráfica esteja sendo exibida, receberá uma mensagem da Unidade de Controle de Objetos para fechamento da sua janela;

- O relatório de avaliação é gerado;
- Objetos questões disponibilizados no ROL são destruídos; e
- Inicia comunicação entre o MAR e o MCA, o relatório é enviado e o aluno é informado do *timeout* da avaliação.

#### 4.2.3.2 Agente tempo da avaliação local

Consiste em um agente de tempo responsável por verificar se o tempo de execução do item está dentro do previsto, visando o acionamento dos tutores didáticos.

Os tempos de execução do item são monitorados dentro de uma faixa entre valores máximo e mínimo de tempos de solução. Soluções enviadas em tempo superior ao tempo máximo previsto pelo professor ou inferior ao tempo mínimo desencadeiam atividades de auxílio ao aluno, visando tratar situações de dúvida (*timeout*) ou de possível “chute”.

Caso o tempo estipulado para a execução de cada item seja atingido, um agente tutor será instanciado visando apresentar ao aluno os recursos disponibilizados pelo professor. O agente comportamento do item (ACA) é notificado da ocorrência de *timeout* do item, visando a notificação no relatório do item.

Caso o tempo estipulado para a execução de cada item esteja abaixo do valor definido pelo professor como um tempo “mínimo” o agente tutor intercede, buscando evitar precipitações por parte do aluno. Neste caso, o agente comportamento também é notificado.

A contagem do tempo de execução do item é suspensa quando o aluno minimiza uma questão para respondê-la posteriormente. A retomada ocorre quando é restaurada a janela da questão.

O Agente Tempo que monitora a execução de cada item possui dois estados básicos (figura 4.12):

- No estado contando, ele carrega e inicia a contagem do tempo até o limite previsto pelo professor; e
- No estado pausado, ele interrompe a contagem do tempo e notifica o Agente Comportamento.

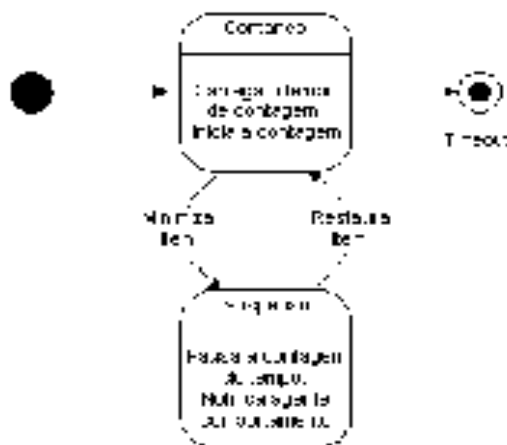


FIGURA 4.12 - Estados do Agente Tempo.

#### 4.2.3.3 Interação entre agentes e o sistema

Os Agentes Tempo e Comportamento comunicam-se através de mensagens, visando ativar funcionalidades e cooperar no sentido de oferecer ao professor as informações precisas sobre os eventos ocorridos durante a execução da avaliação. Estas informações são compiladas e entregues ao objeto relatório, conforme item 4.2.4.

A comunicação não se limita ao ambiente dos agentes. Outros objetos, como interface gráfica e elementos de controle, podem atuar sobre os agentes informando e solicitando ações específicas (figura 4.13).

A interface gráfica de cada item fornece a estrutura para monitorar e informar as interações com o aluno aos agentes.

Os seguintes eventos são registrados e enviados aos agentes:

- Minimização e maximização de janelas;
- Correções ou modificações na resposta do item; e
- Envio da resposta.

Minimização e maximização de janelas - quando o aluno, ao executar uma questão, resolve minimizar um item para posterior solução, o agente tempo que controla o tempo local da execução é interrompido. O agente comportamento é notificado da ocorrência. Quando a janela do item é restaurada, a contagem do agente tempo é retomada, prosseguindo até o tempo previsto pelo professor, quando aciona os agentes tutores.

Correções ou modificações na resposta do item - ocorre quando o aluno responde uma questão e clica sobre o botão “limpar” da questão, ou modifica sua resposta (nos casos de questões com alternativas para escolha). O agente comportamento é notificado.

Envio da resposta - Quando o aluno envia a resposta, ao clicar no botão “Enviar Resposta”, a interface gráfica notifica o agente comportamento sobre o acerto ou não da questão, bem como a quantidade de acertos do item solucionado.

O Agente Comportamento da Avaliação é o elemento chave do suporte de tutoramento remoto implementado pelo ambiente Javal. Como sugestão para trabalhos futuros, pode-se adicionar ferramentas de mineração de dados visando levantar informações comportamentais precisas para o ajuste de perfis, estratégias e metodologias adotadas para o aluno, tomando por base as informações prestadas por este agente.

#### 4.2.4 Encerramento da avaliação

O encerramento da avaliação dá-se com a remessa do seu relatório para o Módulo de Controle da Avaliação (MCA). Cada objeto do tipo item agrega uma instância do objeto Relatório, gerado pelo Agente Comportamento da Avaliação (ACA) com base nas informações fornecidas pelo Agente Tempo e pelos eventos de interface gráfica monitorados.

Ao término da avaliação, a Unidade de Controle do Módulo busca o objeto relatório de cada item da avaliação, visando sua remessa ao MCA e o levantamento de informações ligadas à avaliação, como:

- Tipo do item e número;

- Tempo de solução do item e tempo máximo previsto; e
- Relatório de comportamento da questão, composto por:
  - ⊗ Quantidade de interações com a questão (máx);
  - ⊗ Quantidade de acertos do item; e
  - ⊗ Ocorrências de *timeout*, precipitações ou minimizações de questão.

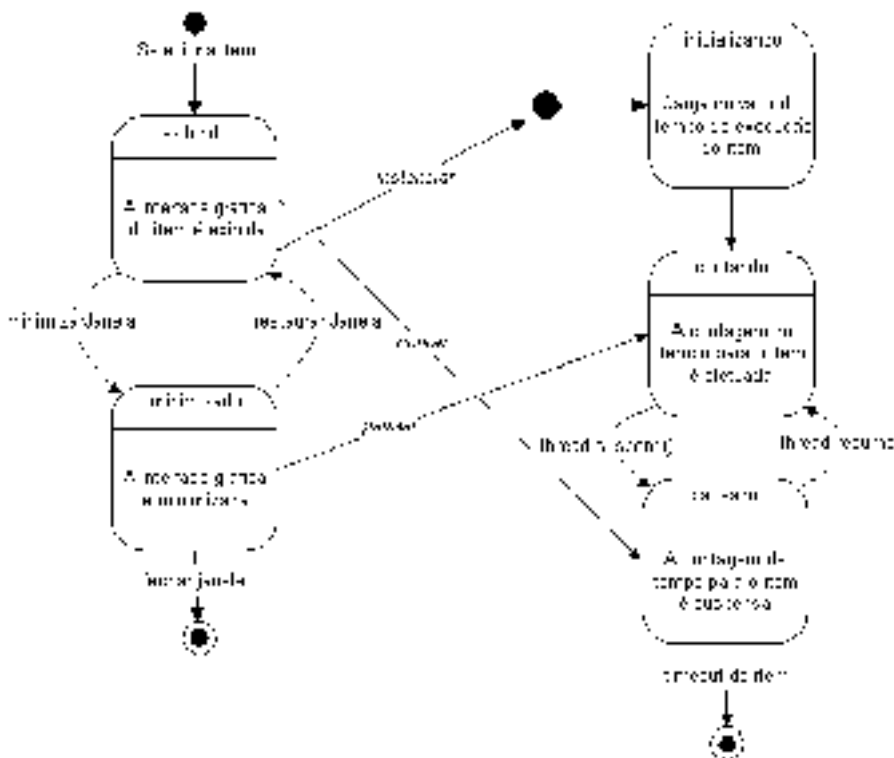


FIGURA 4.13 - Comunicação entre a interface gráfica do item e o AgenteTempo

Maiores detalhes sobre o objeto Relatório estão descritos no capítulo 5.

O relatório da avaliação é um objeto instanciado pela Unidade de Controle do Módulo, cujo conteúdo contém um sumário da avaliação realizada, com os seguintes itens (a tabela 4.5 é preenchida como exemplo):

- Item (ITEM);
- Tempo previsto (TP);
- Tempo de solução (TS);
- Status (STAT);
- Valor (VLR);
- Observações (OBS); e
- Recursos utilizados (RU).

Item - define o tipo de elemento componente da avaliação.

Tempo previsto (TP) - é o tempo de solução previsto pelo para cada item, seja questão ou avaliação. O tempo geralmente é associado a um de grau de dificuldade atribuída à cada questão, de forma a garantir coerência na montagem das avaliações, uma vez que questões com maior grau de dificuldade geralmente tenham tempo maior



de solução do que questão mais fáceis.

Tempo de solução (TS) - engloba o tempo de solução de cada item realizado pelo aluno. Quando este tempo é superior ao TP, considera-se a ocorrência de *timeout*, sendo lançado na coluna observações.

Status (STAT) - informa ao professor o status de cada tipo de item. Os códigos previstos são mensagens descritas na tabela 4.3:

TABELA 4.3 - Códigos de controle do campo STAT do relatório.

<i>Código</i>	<i>Descrição</i>	<i>Tipo de Item</i>
ACERTO	Sinaliza quando o aluno tenha acertado sua resposta.	Questões.
ERRO	Sinaliza caso o aluno tenha errado.	Questões.
NRS	Código para questões não respondidas, caso o aluno não tenha solucionado a questão.	Questões.
REALIZADO	Código para avaliação, caso o aluno tenha concluído dentro do tempo previsto.	Avaliação.

Valor (VLR) - informa ao professor o valor individual de cada item. Para questões, é calculado com base nos itens corretos de cada questão. Para avaliação corresponde ao somatório dos pontos obtidos em cada questão.

TABELA 4.4- Códigos de controle do campo OBS do relatório.

<i>Código</i>	<i>Descrição</i>	<i>Tipo de Item</i>
OK	Item concluído dentro do tempo previsto para sua solução.	Questões
TIMEOUT	Item não concluído dentro do tempo previsto para sua solução (TP).	Avaliação Questões
AGN(TOL)	Indica que o aluno utilizou agentes para realizar a avaliação. O parâmetro TOL define a tolerância de tempo para a questão, em função do uso de agentes para sua solução.	Avaliação
NAGN	Indica que o aluno não utilizou agentes para realizar a avaliação.	

Observações (OBS) - apresenta ao professor informações importantes sobre emprego ou não de tutores (caso o aluno tenha desativado no início da avaliação) e sobre a execução em relação ao prazo. As mensagens previstas para este campo são as previstas na tabela 4.4.

Recursos Utilizados (RU) - registra os recursos disponibilizados pelo professor e que foram utilizados pelo aluno. Permite ao professor adequar recursos, ajustando recursos úteis e aperfeiçoando recursos pouco utilizados, tornando-os úteis para o aluno durante a solução da avaliação

O exemplo da tabela 4.5 ilustra o relatório gerado para uma avaliação realizada, cujo tempo previsto pelo professor foi de 20 minutos.

Pode-se observar que:

- O aluno realizou a avaliação dentro do tempo previsto (28 minutos) e obteve

grau final igual a 8 ;

- Houve a extrapolação do tempo na questão 2 e que o recurso página web foi utilizada como suporte para a solução de duas questões (2 e 5); e
- O aluno errou uma questão (nr. 4), embora tenha respondido dentro do prazo e consultado a lista de discussão sobre o tema.

TABELA 4.5 - Um exemplo de relatório de questão realizada por aluno no MAR..

<i>ITEM</i>	<i>TS</i>	<i>TP (min)</i>	<i>STAT</i>	<i>VLR</i>	<i>OBS</i>	<i>RU</i>
AVALIAÇÃO	28	30	REALIZADO	8	AGN(20)	
QUESTÃO 1	4	5	ACERTO	1	OK	
QUESTÃO 2	7	5	ACERTO	2	TIMEOUT	R 01 - PAGINA WEB
QUESTÃO 3	9	10	ACERTO	1	OK	
QUESTÃO 4	2	3	ERRO	0	OK	R 02 - LISTA DE DISCUSSAO
QUESTÃO 5	6	7	ACERTO	2	TIMEOUT	R 01 - PAGINA WEB

### 4.3 Resumo do capítulo quatro

Neste capítulo foi apresentada a funcionalidade do ambiente Javal, focalizando o ambiente do Módulo de Avaliação Remota (MAR). Visando uma situação do MAR no ambiente Javal, foram descritos os principais módulos e funcionalidades dos diversos componentes deste ambiente.

Em seguida, foi apresentada a estrutura interna do Módulo de Avaliação Remota, visando identificar os componentes dos módulos e funcionalidades específicas.

Em seguida foi feita a descrição funcional do MAR, identificando seus estados de funcionamento e os eventos envolvidos. Foram abordados aspectos comportamentais dos agentes, bem como a interação existente entre os principais objetos do sistema.

No capítulo 5 será apresentada a descrição de cada classe implementada pelos diversos pacotes do ambiente.

## 5 A implementação do MAR

O ambiente Módulo de Avaliação Remota (MAR) é composto por uma biblioteca de classes cujas funcionalidades garantem o armazenamento local e instanciação de objetos do tipo avaliação, descarregados do módulo MCA.

Apresenta uma interface gráfica amigável, capaz de permitir que o aluno possa interagir com o ambiente, resolvendo questões, interagindo com agentes tutores e consultando recursos disponibilizados pelo professor, visando a solução de questões.

Para que o MAR seja flexível e modular, suas classes de objetos foram reunidas em pacotes com funcionalidades específicas, que poderão ser estendidas pela inclusão de novas classes, dentro da estrutura proposta pelo ambiente.

Devido às funcionalidades modeladas para o MAR, tornou-se necessário criar classes especialistas, classificadas em dois grupos:

- As classes do ambiente de avaliação - são capazes de instanciar objetos relacionados com o aspecto pedagógico da avaliação como avaliação, questões, agentes e tutores.
- As classes de suporte e controle - instanciam objetos responsáveis pelas funcionalidades do ambiente, como controle, comunicação e segurança.

### 5.1 Objetos do ambiente de avaliação

A estrutura de classes do ambiente de avaliação, modela os objetos e funcionalidades descritos abaixo :

- Objetos do tipo avaliação;
- Objetos do tipo questão;
- Objetos do tipo item;
- Objetos de interface gráfica;
- Objetos do tipo recurso; e
- Objetos do tipo agente.

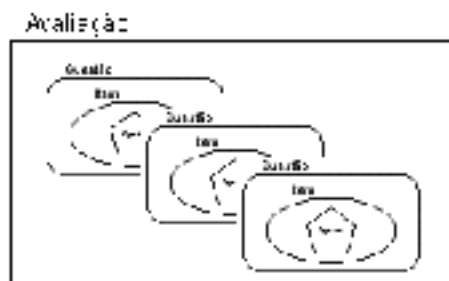


FIGURA 5.1 - Uma descrição conceitual da estrutura de avaliação adotada pelo modelo.

Objeto do tipo avaliação - representa o ambiente da avaliação elaborada pelo professor, com suas questões, agentes e recursos alocados para a sua solução. Seus métodos são capazes de lhe prover as seguintes funcionalidades:

- Carregar os objetos questões e os respectivos objetos itens;
- Inicializar os objetos questões, instanciando objetos de sua interface gráfica;

- Inicializar os agentes capazes de monitorar o controle do tempo da avaliação e de eventos da interface gráfica;
- Permitir a correção da avaliação e o envio do relatório da avaliação ao professor;
- Permitir o alerta ao professor das situações de falha no ambiente do aluno; e
- Permitir o recolhimento da avaliação por destruição dos seus objetos.

Objeto do tipo questão - é um objeto responsável por guardar em armazenamento local as instâncias dos diversos objetos do tipo item componentes da avaliação. Como em uma avaliação tradicional, uma questão pode conter um ou mais itens, cujo somatório dos pontos define o total da questão.

Objeto do tipo item - é o objeto que representa os tipos de questões do ensino tradicional. A classe Item é a classe raiz dos objetos empregados para o desenvolvimento de classes que implementem este tipo de objeto. Cada objeto do tipo item possui um objeto do tipo interface gráfica do pacote `java1.jstudent` correspondente.

Objeto de interface gráfica - o pacote de classe Jstudent define as interfaces gráficas para os diversos tipos de objetos que oferecem interação com o aluno. Esta interface gráfica agrega o tipo de item da questão e incorpora os agentes tempo e comportamento a serem inicializados durante a solução de cada tipo de item.

Objeto do tipo recurso - o pacote de classes de recursos Jresource define os objetos que podem ser empregados pelos agentes como recursos a serem exibidos ao aluno. Englobam URLs de páginas Web, recursos multimídia, ajuda on line e ferramentas para comunicação em grupo. Para este modelo, cabe ao professor, através do Módulo de Montagem de Avaliações (MMA), a agregação destes objetos à avaliação elaborada e a definição para os agentes de como e quando empregá-los.

Objeto do tipo agente - o pacote de classe de agentes Jagent define os agentes do sistema são responsáveis pelo acompanhamento do aluno durante a avaliação.

## 5.2 Objetos de suporte e controle

Os objetos de suporte ao ambiente são objetos com funções específicas, ligados ao controle do ambiente. São eles:

- Objeto do tipo Controle - objeto que agrega funcionalidades específicas do ambiente, como controle de instanciação de objetos, monitoração de eventos, etc.; e
- Objeto do tipo Suporte de Comunicação - objeto que agrega funcionalidades de suporte de comunicação do ambiente.

## 5.3 A API de classes de Javal

Como visto, os pacotes do ambiente procuram agrupar classes cujos objetos são capazes de desempenhar funcionalidades específicas dentro do sistema. Inicialmente as classes são organizadas em dois grandes grupos:

- O primeiro compõe as classes básicas do ambiente de avaliação. Suas instâncias são objetos cuja funcionalidade está intimamente relacionada com o objetivo do sistema, ou seja, objetos avaliação e seus componentes; e

- O segundo compõe um conjunto de classes de suporte ao ambiente, cujos objetos implementam a infraestrutura necessária para o seu funcionamento.

Para as classes do ambiente de avaliação temos os pacotes:

- Ⓜ Jquest;
- Ⓜ Jstudent;
- Ⓜ Jagent;
- Ⓜ Jresource; e
- Ⓜ Jtools.

O pacote Jquest implementa objetos do ambiente de avaliação à distância. É composto por uma estrutura de classes cujos objetos representam o ambiente de ensino tradicional, em forma de questões do tipo verdadeiro, falso, múltipla escolha, resposta, correspondência e completamento.

O pacote Jstudent é composto por um conjunto de classes cujas instâncias permitem a criação da interface gráfica com o aluno de cada um dos objetos de avaliação.

O pacote Jagent implementa os objetos agentes embarcados, que acompanharão o desenvolvimento de avaliações e efetuarão detecção de erros ou levantarão situações de dúvida, agindo como auxiliares didáticos.

O pacote Jresource possibilita o emprego de recursos a serem consultados pelo aluno durante a execução da avaliação. Seus objetos representarão recursos como páginas da web, arquivos de áudio ou filmes de vídeo, fóruns de discussão, recursos multimídia e outros que possam ser implementados em trabalhos futuros.

O pacote Jtools prevê o emprego futuro de classes que permitem a disponibilização de ferramentas que possam ser úteis durante a solução da questão como dicionários on line, máquinas de calcular, rascunhos, etc.

Para as classes de suporte ao ambiente, temos os pacotes :

- Jcontrol; e
- Jcomm.

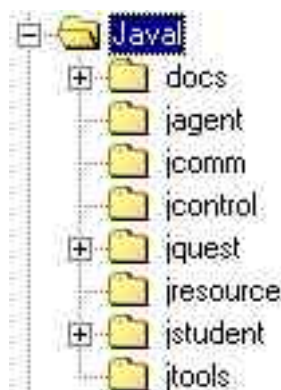


FIGURA 5.2 - Os pacotes da API do ambiente Javal

O pacote Jcontrol implementa classes que oferecem suporte às funcionalidades básicas do ambiente, como controle sobre janelas e tempo da avaliação, busca e encerramento da avaliação e remessa dos relatórios ao Módulo de Controle da Avaliação (MCA), entre outros.

O pacote Jcomm implementa classes que oferecem suporte às funções de comunicação do ambiente. Pretende-se que, em versões futuras do Javal, este pacote inclua ferramentas que permitam a realização de avaliações de forma colaborativa e em grupo.

## 5.4 API do ambiente de avaliação

Serão apresentadas as classes que fazem parte dos módulos relacionados com o ambiente de avaliação. A documentação da API apresenta suas classes organizadas por pacotes, atributos e métodos das classes implementadas.

### 5.4.1 O pacote Jquest

O pacote Jquest (figura 5.3) foi modelado visando permitir a criação de objetos capazes de representar o ambiente de avaliação, tendo por base os tipos de questão descritos em [PER99]. A seguir, serão descritas suas classes componentes e a relação existente entre elas.

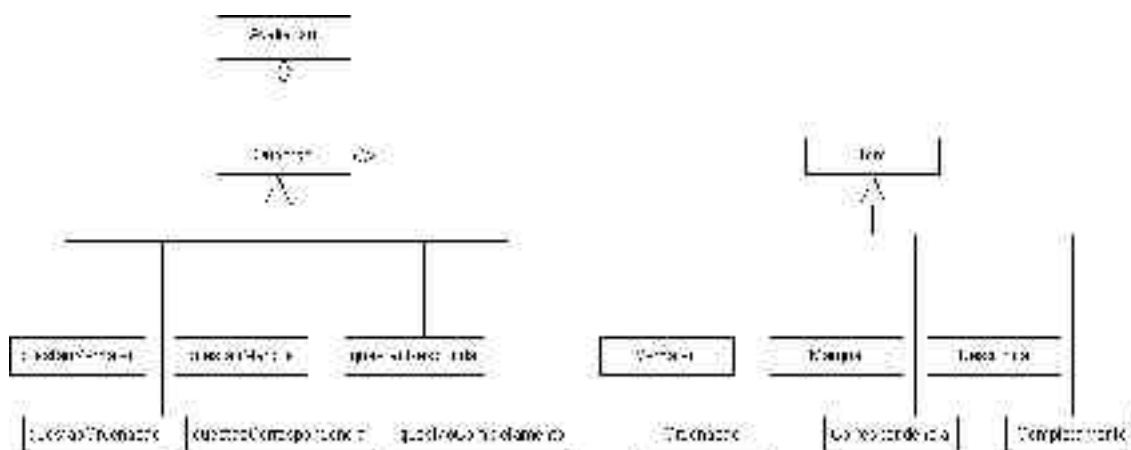


FIGURA 5.3. Modelo de objetos do ambiente Jquest

A classe Avaliacao foi modelada para agregar objetos do tipo questão, que, por sua vez, agrega objetos do tipo item, formando uma estrutura encapsulada, como apresentado na figura 5.1.

A seguir, serão apresentadas as diversas classes implementadas neste pacote, a descrição sumária de seu funcionamento, com seus principais atributos e métodos.

#### 5.4.1.1 Classe Avaliacao

A classe Avaliacao instancia um objeto cuja estrutura implementa um ambiente de avaliação. Seus atributos guardam informações relativas ao tipo de avaliação a ser realizada (individual ou em grupo), seu tempo para solução, bem como o contexto escolar (instituição de ensino, nome do professor, turma de aula, etc.) na qual está inserida.

O objeto Avaliacao agrega os diversos tipos de objetos questão que podem ser utilizados pelo professor para se obter um indicador do conhecimento adquirido pelo aluno durante o curso.

O protótipo implementado para a validação deste modelo, buscou apresentar a distribuição de avaliações assíncronas no tempo e individuais, sendo sugestão para

trabalhos futuros o desenvolvimento de módulos capazes de incrementar as funcionalidades de distribuição síncrona e em ambientes colaborativos de atividades de grupo.

#### **5.4.1.2 Classe Questao**

O objeto Questao é o objeto raiz para a hierarquia de classes que compõem os tipos de questões disponíveis para o ambiente. Suas subclasses implementam as questões-tipo do ambiente de ensino tradicional.

Os objetos questões-tipo estendem a classe Questao e armazenam os tipos de questões disponíveis. Como exemplo, o objeto questaoCompleto é um objeto questão-tipo, em cujo repositório estão guardados instâncias deste tipo de item (Completo).

A classe Questao agrega as instâncias do tipo Item, de forma que, em uma avaliação, podem haver várias questões e cada uma possui um conjunto de itens que lhe confere.

Os atributos de questão guardam informações importantes sobre sua identificação na avaliação, o tipo e o valor total de pontos que possui. A contabilização de pontos para cada questão é calculada segundo métodos específicos para cada tipo de item elaborado.

Visando aumentar a flexibilidade na solução de questões, o ambiente possui uma interface gráfica de navegação de questões denominada Navegador de Questões, que permite ao aluno navegar por questões e selecionar seus itens para a solução.

Quando o aluno seleciona uma questão na applet, a Unidade de Controle é notificada, e os objetos questão armazenados na avaliação são exibidos, dependendo da questão-tipo selecionada.

Subclasses implementadas: questaoCompleto, questaoCorrespondencia, questaoMarque, questaoOrdenacao, questaoResponda, questaoVerFalso.

#### **5.4.1.3 Classe Item**

A classe Item é uma classe abstrata, origem da hierarquia de classes para os objetos do tipo item do ambiente de avaliação. O objeto Item é a base pedagógica do ambiente avaliação, permitindo a verificação do aluno quanto ao seu aprendizado sobre determinado tópico.

Durante a montagem da avaliação, o professor pode selecionar e disponibilizar recursos que, por intermédio dos agentes, possam ser exibidos visando auxiliar o aluno na elaboração de sua resposta.

Os itens entregues ao aluno inicializam sensores que são constantemente monitorados, de forma a permitir que agentes, agindo como tutores ou auxiliares didáticos [LEI01], detectem situações de falha por possível não entendimento da questão, ocasionando a chamada de procedimentos de ajuda ao aluno.

Os agentes recebem instruções sobre quais recursos devam ser disponibilizados, de acordo com os parâmetros informados durante a montagem da avaliação.

Este auxílio tem por objetivo o esclarecimento da questão, através da exibição de sugestões, dicas ou outros recursos previamente disponibilizados pelo professor

como sites web, lista de discussão, arquivos de áudio e vídeo, etc.

O método `corrigelItem`, descrito pela classe `Item` e sobrescrito pelas diversas classes que o estendem permitem implementar a forma de correção peculiar a cada tipo de item. Novas formas de correção podem ser implementadas, pela modificação da implementação deste método.

Subclasses implementadas: `Completamento`, `Correspondencia`, `Ordenacao`, `Responda`, `VerFalso`.

A atribuição de graus de dificuldade a questões aplicadas, visa o uso desta classe em trabalhos futuros, no Módulo de Montagem de Avaliações (MMA) do ambiente Javal, não implementado nesta versão.

### 5.4.2 Recursos

O ambiente foi desenvolvido para permitir que um conjunto de recursos disponibilizados pelo professor ofereçam ao aluno auxílio para a solução dos itens componentes das questões. Os agentes podem ou não ter autorização para sua exibição automática ao aluno, quando julgar que este esteja passando por dificuldades na solução da questão.

Para o presente modelo, os recursos que o professor pode tornar disponíveis são:

- O professor (ele próprio, quando em avaliações síncronas ou através de mail ou serviço de fórum, quando em avaliações assíncronas);
- Palavras Chave (Dicas);
- Páginas Web;
- Arquivos de Áudio;
- Imagens; e
- Vídeo.

O aluno, no entanto, pode escolher entre utilizar estes recursos ou outros que em trabalhos futuros possam ser agregados, aumentando o poder de abrangência da ferramenta ora proposta:

- Consulta a grupo de trabalho;
- Acesso a lista de discussão sobre a matéria; e
- Outras ferramentas úteis para auxílio na avaliação.

Para o presente trabalho este pacote (*javal.jresource*) não está implementado, sendo interessante seu estudo e desenvolvimento em trabalhos posteriores.

### 5.4.3 O pacote Jstudent

Modela os objetos que representam as diversas interfaces gráficas do ambiente com o usuário. Seus objetos permitem ao aluno interagir com o ambiente de avaliação, selecionando questões, escolhendo e respondendo aos diversos tipos de itens componentes da avaliação.

O pacote `Jstudent` possui íntima relação com os objetos que estendem a classe



Item, uma vez que suas instâncias agregam os objetos correspondentes dos diversos tipos de itens existentes no ambiente.

Quanto à sua funcionalidade, as classes do pacote Jstudent estendem duas classes básicas (figura 4.16):

- Java.awt.Frame - Necessária para a criação de objetos do tipo janela; e
- Java.applet.Applet - Para a criação da applet-base do ambiente.

A classe Frame (*java.awt.Frame*) é a classe raiz para os objetos do pacote ligados à instanciação de objetos do tipo Frame, permitindo “saltar” da janela do navegador as diversas interfaces gráficas dos objetos item do ambiente.

Desta classe foram criadas as subclasses:

- FrameItem é a classe raiz para criação de objetos do tipo interface gráfica para cada tipo de item; e
- FrameAgente é a classe root para a criação de objetos do tipo interface gráfica para os agentes.

A classe Applet (*java.applet.Applet*) é a classe raiz da classe *appletAvaliacao*, a applet descarregada para o ambiente do aluno que consiste na base do módulo MAR. A partir desta applet são carregados objetos do tipo avaliação (*Javal.Jquest.Avaliacao*), que contém os diversos componentes da avaliação como questões, itens, etc.

A integração entre as classes que instanciam objetos da avaliação (Item) e os objetos que instanciam a interface gráfica pode ser analisada, conforme a figura 5.5.

Os objetos de interface gráfica que estenderem a classe FrameItem, agregam funcionalidades específicas e necessárias para a execução das avaliações, como os agentes Tempo (ATA) e Comportamento (ACA), para cada tipo de item.

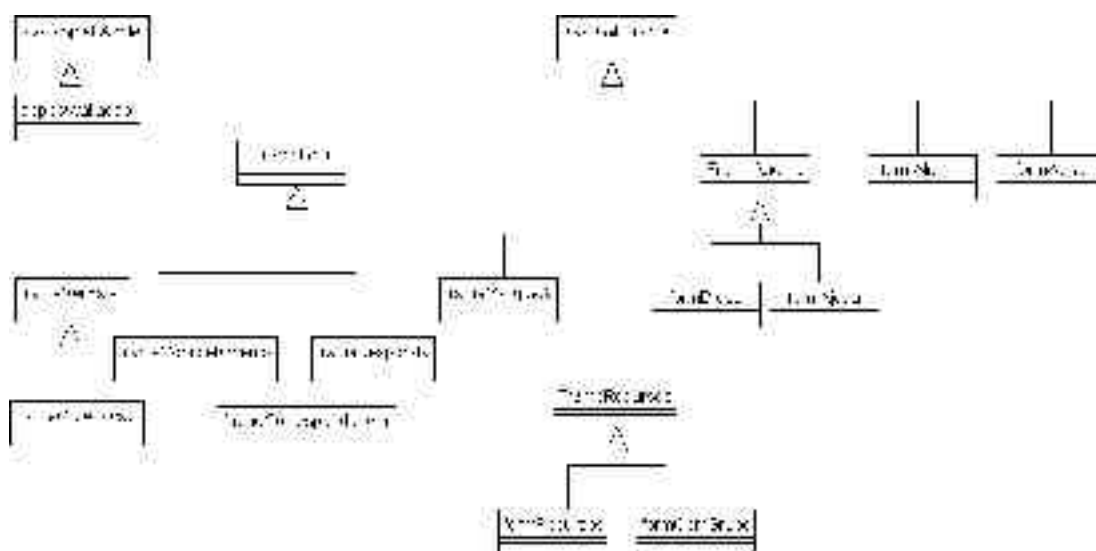


FIGURA 5.4 - Modelo de objetos do ambiente Jstudent.

Desta forma, cada objeto da interface gráfica que represente uma questão-tipo, ao ser instanciado, carregará consigo os agentes necessários para sua operação.

Dinamicamente, os objetos das questões-tipo agregam objetos das interfaces gráficas correspondentes. Estas incorporam seus objetos correspondentes do tipo de Item considerado. Esta construção facilita as ações da interface gráfica sobre os métodos

do item e dos agentes que lhe estão agregados .

A classe *Jstudent* incorpora classes que instanciam interfaces gráficas diversas, para os diversos objetos do ambiente. Desta forma temos:

- O pacote *java.jstudent.rsr* implementa os objetos interface gráfica para os objetos do tipo recursos; e
- O pacote *java.jstudent.tools* implementa os objetos de interface gráfica para os eventos e avisos do sistema.

#### 5.4.3.1 Classe *FrameItem*

Um *FrameItem* é a janela básica para desenvolvimento de interface gráfica para objetos da classe *Javal*. A classe *FrameItem* é a classe raiz para objetos da interface gráfica. Seus métodos permitem a criação de objetos que incorporam objetos do tipo agente. Todo o objeto que estender esta classe terá incorporados o agente comportamento e agente tempo.

O método *corrige()* descritos na classe *Frame Item* e sobrescritos pelas classes que a estendem permitem a chamada do método *corrigeItem()*, disponível em cada objeto do tipo *Item*.

Subclasses implementadas: *frameCompletamento*, *frameCorrespondencia*, *frameResposta*, *frameVerFalso*.

#### 5.4.3.2 Classe *frameVerFalso*

A classe *frameVerfalso* (figura 5.5) instancia a interface gráfica para o item que representa questões do tipo Verdadeiro-Falso. Seus métodos permitem informar ao agente comportamento correspondente as interações realizadas entre o aluno e a interface gráfica durante a sua solução.

A questão verdadeiro ou falso exibe um enunciado e uma série de idéias em um conjunto de 4 alternativas. A cada alternativa corresponde uma caixa de combinação onde o avaliado escolherá se o significado da alternativa é correto (V) ou errado (F) .

Subclasses implementadas : *frameMarque*, *frameOrdenacao*.



FIGURA 5.5- Interface gráfica do ambiente Verdadeiro-Falso.

### 5.4.3.3 Classe frameMarque

A classe frameMarque instancia a interface gráfica (figura 5.6) para o item que representa questões do tipo “marque a alternativa correta”.

Neste tipo de questão, a exemplo da questão Verdadeiro-Falso, são exibidos um enunciado e quatro alternativas para o aluno. O aluno deverá responder clicando no componente de botão de rádio que estiver ao lado da alternativa julgada correta.

Visando o seu aperfeiçoamento, este tipo de questão permite que o professor ative recursos de diálogo do agente com o aluno, cumprimentando-o, se a solução estiver correta ou realizando a correção imediata da aprendizagem, caso a resposta emitida esteja errada.



FIGURA 5.6 - Interface gráfica do ambiente Marque.

### 5.4.3.4 Classe frameOrdenacao

A classe frameOrdenacao instancia a interface gráfica para o item que representa questões do tipo Ordenação (figura 5.7)

Na questão do tipo ordenação, é apresentado um enunciado e uma série de eventos que devem ser ordenado segundo uma seqüência de 1 a 4. A questão é julgada como certa quando o aluno acerta toda a seqüência prevista pelo professor.

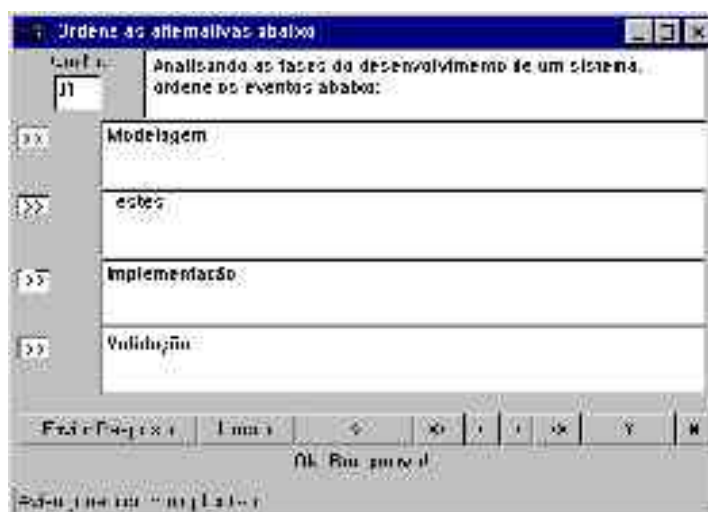


FIGURA 5.7- Interface gráfica do ambiente Ordenação

### 5.4.3.5 Classe frameCorrespondencia

A classe frameCorrespondencia instancia a interface gráfica (figura 5.8) para o item que representa questões do tipo Correspondencia.

A correspondência é realizada pela relação entre duas colunas, chamadas A e B, com idéias chave e seu respectivo significado. Ao aluno cabe relacionar o conceito entre as colunas.

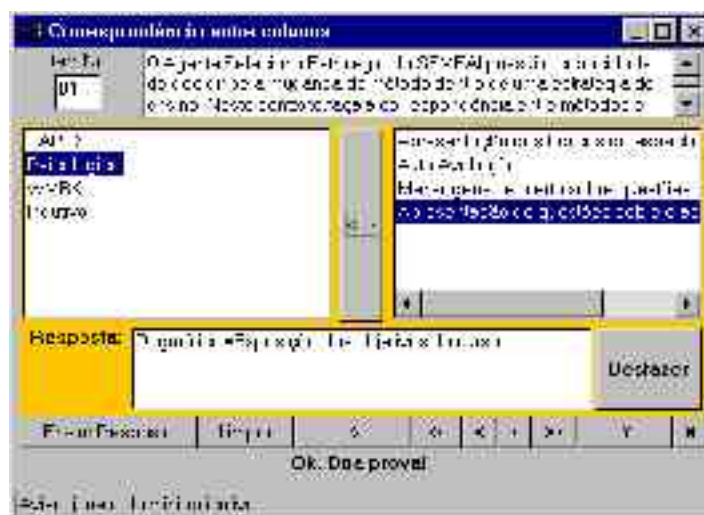


FIGURA 5.8- Interface gráfica do ambiente Correspondência.

A pontuação do item é obtida contando-se a quantidade de relações corretas.

### 5.4.3.6 Classe frameResposta

A classe frameResposta instancia a interface gráfica (figura 5.9) para o item que representa questões do tipo Resposta.

Por se tratar de questão subjetiva, não há implementação do método corrigeItem para este tipo de objeto. Neste caso, todo o conteúdo da resposta é passado para o banco de dados, sendo corrigido apenas pelo professor.

Como sugestão para trabalhos futuros, pode-se implementar uma lógica de correção baseada em análise de texto e de conteúdo, não implementados nesta versão.

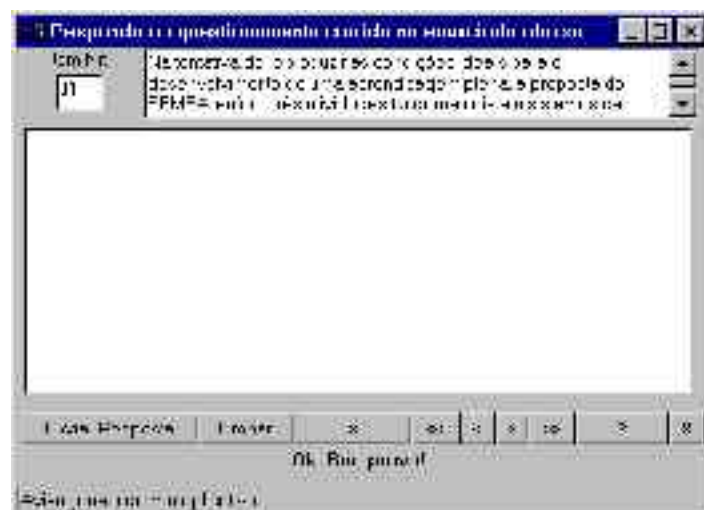


FIGURA 5.9- Interface gráfica do ambiente Resposta.

### 5.4.3.7 Classe frameCompletamento

A classe frameCompletamento instancia a interface gráfica para o item que representa questões do tipo Completamento, conforme figura 5.10.

Uma questão do tipo completamento é composta por uma coluna com o enunciado, onde existem lacunas a serem preenchidas pelo aluno. Este deve selecionar a lacuna que deseja preencher e inserir o conceito ou idéia capaz de tornar aquela afirmação correta. O botão desfazer permite desfazer a inserção realizada.

A pontuação do item é obtida contando-se a quantidade de inserções corretas.

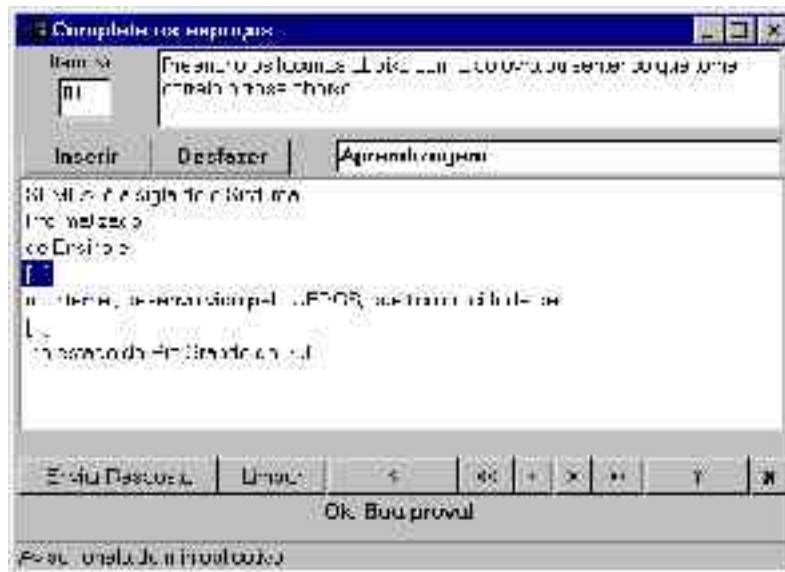


FIGURA 5.10 - Interface gráfica do ambiente Completamento.

### 5.4.4 O pacote Jagent

Modela os objetos representativos dos agentes que intercedem junto ao aluno durante a execução de avaliação. Seu diagrama de classes é apresentado na figura 5.11.

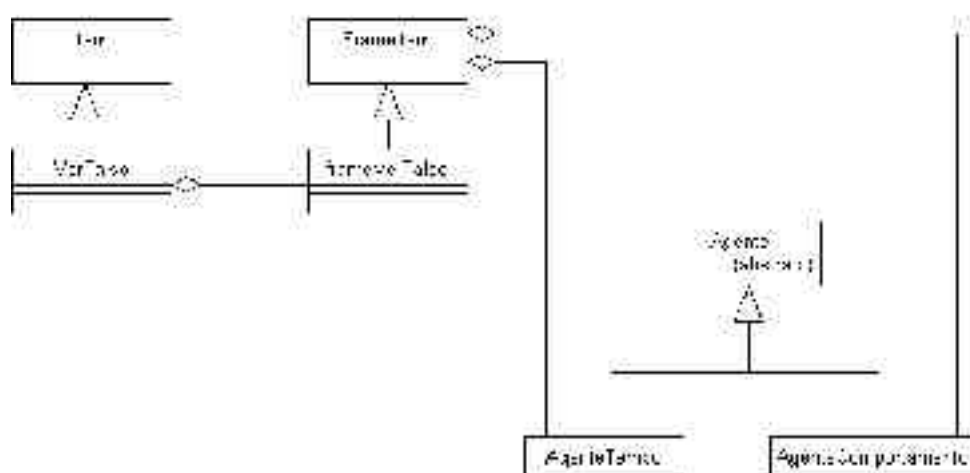


FIGURA 5.11 - Diagrama de Classes da pacote Jagent.

O modelo proposto emprega um recurso denominado agentes embarcados (monitor e tutor), uma vez que são serializados e executados junto ao ambiente aluno,

de onde permanecem monitorando seu comportamento durante a avaliação.

Os agentes entram em ação tão logo a interface do ambiente (o item propriamente dito) esteja disponibilizada para o aluno. A interação com o aluno é discreta, cessando no momento em que o aluno encerra a avaliação ou esta é terminada automaticamente por ordem de algum dos agentes.

Para validar o modelo, os agentes embarcados são especialistas em controle de tempo e de comportamento, sendo denominados AgenteTempo (AT) e Agente Comportamento (AC), cujas finalidades são descritas a seguir:

- O primeiro tem por finalidade acompanhar o tempo de execução de uma avaliação ou de parte dela, onde é capaz de comparar o tempo previsto pelo professor com o tempo de execução, identificando possíveis situações de dúvida ou falha do sistema; e
- O segundo tem por finalidade monitorar o ambiente gráfico, procurando identificar cliques de mouse, correção ou cancelamento de respostas, enfim, interações do aluno com o ambiente gráfico que sejam capazes de evidenciar situações de dúvida ou falha.

A importância de agentes cresce na medida em que eles são responsáveis pelo acompanhamento da avaliação no ambiente aluno, atuando não apenas como monitores de comportamento, mas como auxiliares do professor durante a avaliação.

Quanto à taxinomia de agentes, citado em [FRA 97], os agentes para o modelo proposto podem ser classificados como :

- ⊗ Do tipo reativo (isto é, sente e atua), responde na hora a mudanças no ambiente (respostas incorretas ou demoradas);
- ⊗ Temporariamente intermitente, está processando apenas durante as atividades de solução da avaliação; e

Comunicativo (sociável), comunica com outros agentes, e com pessoas (alunos) através de interface amigável com linguagem adaptável ao tipo de aluno.

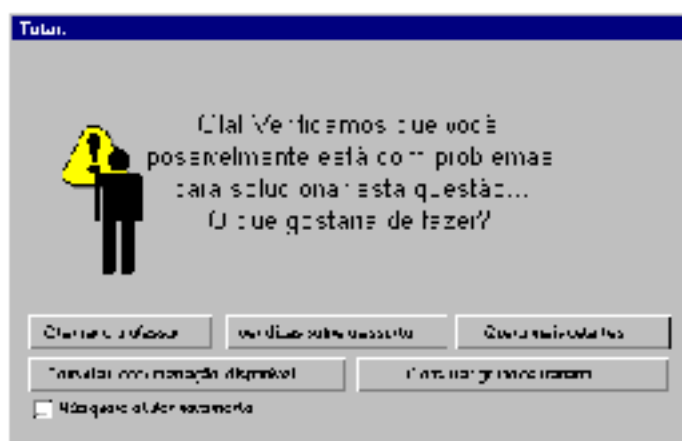


FIGURA 5.12 - Interface gráfica de um agente.

Os agentes possuem uma interface gráfica amigável (figura 5.12), capaz de interagir com o aluno, de forma a conduzi-lo na execução da tarefa proposta.

Como sugestão para trabalhos futuros, sugere-se o estudo visando a adequação no uso de cores e imagens, bem como da linguagem de comunicação do agente, tomando por base o perfil do aluno.

#### 5.4.4.1 Classe AgenteComportamento

O objeto da classe Agente Comportamento captura eventos da interface gráfica, visando a monitoração do comportamento do aluno durante as avaliações. Pode receber mensagens da interface gráfica sobre comportamentos, correções ou modificações realizadas pelo aluno antes do envio de sua resposta.

O Agente Comportamento pode enviar mensagem para o Agente Tempo da avaliação ou do item correspondente, visando a suspensão do seu tempo de contagem, quando necessário.

É o responsável pela geração de um objeto relatório, disponível em cada tipo de item da avaliação.

#### 5.4.4.2 Classe AgenteTempo

Implementa o agente de tempo para avaliação. O agente é um elemento inicializado juntamente com uma questão ou avaliação.

Após o tempo de execução de cada item, o agente aciona um tutor, que é exibido ao aluno, de forma a auxiliar-lhe na sua solução.

A classe Agente Tempo implementa o objeto ATAV, que é o agente tempo da avaliação, empregado pela Unidade de Controle para a monitoração do tempo de execução da avaliação, denominado tempo global.

### 5.5 Interação entre elementos do ambiente

O presente item do capítulo visa apresentar aspectos relacionados à interação entre os objetos das diferentes classes componentes do sistema, e constitui a base para o funcionamento do ambiente MAR.

Como o ambiente Javal é Orientado a Objetos, a comunicação entre seus módulos é feita através da passagem de parâmetros (para a criação de objetos) ou emprego de mensagens, para comunicação entre instâncias ou criação de novos objetos .

As mensagens podem ter origem em eventos do sistema ou interações com a interface gráfica do ambiente, conforme figura 5.13.

As mensagens do sistema podem ser classificadas em 3 categorias:

- Mensagens entre agentes;
- Interações ambiente-aluno; e
- Interações aluno-ambiente.

As mensagens entre agentes - envolvem o conjunto de mensagens entre objetos dentro do ambiente da avaliação. Estas mensagens podem ser frutos de eventos do sistema ou interações externas, vindo do aluno ou professor.

As interações ambiente-aluno envolvem mensagens ou interações cuja percepção para o aluno pode se fazer de forma visual ou sonora. Envolve o grupo de mensagens ligadas à exibição de ambientes gráficos, tutores, recursos e quaisquer outras formas de comunicação capaz de ser feita com o aluno.

As interações aluno-ambiente envolvem um conjunto de mensagens fruto da interação do aluno com a interface gráfica do ambiente. Envolve seleção de questões, itens, ativação e desativação de recursos do ambiente, seleção e emprego de recursos

disponibilizados pelo professor.

As classes componentes dos diversos pacotes do ambiente Javal incorporam métodos capazes de implementar as funcionalidades previstas para o sistema, conforme descritos na documentação do ambiente Javal, em anexo.

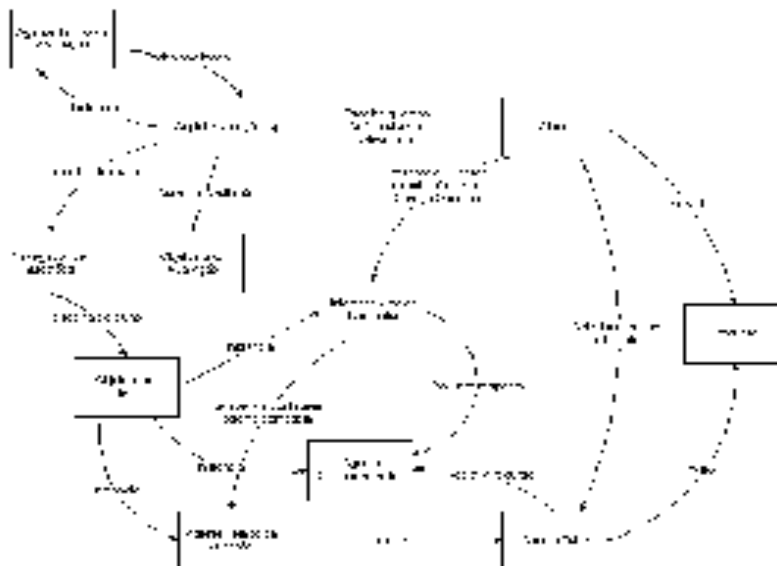


FIGURA 5.13 - Mensagens entre objetos componentes do ambiente MAR.

### 5.5.1 Mensagens entre agentes

Durante o funcionamento do ambiente Javal, os agentes se comunicam, passando, recebendo parâmetros ou instanciando novos objetos.

Os principais elementos envolvidos são o agente tempo da questão, o agente tutor e agente Comportamento.

As principais mensagens entre agentes estão na tabela 5.1 :

TABELA 5.1 - Principais interações entre agentes do MAR

<i>Mensagem</i>	<i>Origem</i>	<i>Destino</i>	<i>Funcionalidade</i>	<i>Obs.</i>
Inicializa	Agente Tempo da questão	Agente Tutor	Inicializar o agente tutor	Habilita para o aluno os recursos disponibilizados pelo professor.
Inicializa	Agente Comportamento			
Exibe	Agente Tutor	Recurso	Exibir recursos para o aluno	De acordo com as diretrizes operacionais definidas pelo professor
Registra recurso	Agente Tutor	Agente Comportamento	Informar ao agente comportamento qual o recurso utilizado para consulta pelo aluno	



<i>Mensagem</i>	<i>Origem</i>	<i>Destino</i>	<i>Funcionalidade</i>	<i>Obs.</i>
Registra resposta	Interface gráfica	Agente comportamento	Informa ao agente a interação do aluno com o ambiente gráfico da resposta, ao modificá-las	Permite monitorar mudanças de respostas ocorridas.
Suspende contagem	Interface gráfica	Agente Tempo	Informa ao agente tempo que deve suspender a contagem do tempo para o item.	A janela do item foi minimizada pelo aluno.
Retoma contagem	Interface gráfica	Agente Tempo	Informa ao agente que deve retomar a contagem do tempo para o item	A janela do item foi restaurada pelo aluno

### 5.5.2 Interações aluno-ambiente

Consistem de mensagens que agem como inicializadores de novos processos dentro do sistema, fruto da interação entre o aluno e o ambiente. Estão ligados a eventos da interface gráfica do ambiente.

A tabela 5.2 ilustra as principais interações e seus efeitos.

TABELA 5.2 . Principais interações entre o aluno e o ambiente.

<i>Evento</i>	<i>Mensagem</i>	<i>Destino</i>	<i>Funcionalidade</i>	<i>Obs.</i>
Escolha da questão na applet avaliação	Escolha questão	Applet Avaliação	Seleciona a questão desejada para solução.	Inicializa a interface gráfica do Navegador de Questões.
Escolha do item da questão no Navegador de Questões	Seleciona item	Navegador de Questões	Permite escolher o item a ser solucionado dentro de uma questão.	Exibe a interface gráfica do item selecionado.
Ajuste do alarme de proximidade do fim de prova	Define Alarme	Applet Avaliação (Agente Tempo da Avaliação)	Define para o agente tempo da avaliação o tempo necessário para o alerta de proximidade do fim de prova.	Applet avaliação instancia objeto Agente Tempo, passando a informação como parâmetro.
Ativação / desativação de tutores	Ativa / Desativa Tutor	Applet Avaliação	Permite ativar / desativar tutor para a avaliação.	Alerta ao aluno sobre a ativação/desativação do tutor.
Seleção de recursos para consulta	Seleciona recurso	Recurso	Permite selecionar o recurso desejado.	Permite a exibição do recurso.
Minimização ou Maximização de Janelas	Minimiza / Maximiza Item	Interface gráfica da questão	Permite minimizar / maximizar o item em execução.	O agente comportamento é notificado. O agente tempo é pausado/reiniciado.

<i>Evento</i>	<i>Mensagem</i>	<i>Destino</i>	<i>Funcionalidade</i>	<i>Obs.</i>
Envio de resposta da questão	Envia resposta	Interface gráfica da questão (Questão)	Permite ativar o método que faz a correção da questão.	O aluno é notificado sobre o acerto ou não da questão. O relatório do item é gerado.

### 5.5.3 Mensagens ambiente-aluno

Consistem em mensagens que o sistema exhibe para o aluno, sob forma de telas de aviso, tutores, ou de eventos do sistema.

Os principais eventos considerados estão na tabela 5.3:

TABELA 5.3 - Principais interações entre o ambiente e o aluno.

<i>Evento</i>	<i>Mensagem</i>	<i>Origem</i>	<i>Funcionalidade</i>	<i>Obs.</i>
Proximidade do fim da avaliação	Alarme avaliação	Agente Tempo de Avaliação	Informa ao aluno que o final da avaliação está próximo.	Depende da ativação do timer para a solução da questão.
Encerramento da avaliação	Fim avaliação	Agente Tempo de Avaliação	Notifica que a avaliação está encerrada.	Exibição do módulo Navegador de questões é bloqueada. Se sua janela está ativa, é encerrada.
Acerto ou erro da questão	Exibe mensagem	Objeto tipo Item	Informa ao aluno que acertou a questão.	Depende do tipo de item.
Exibição do agente tutor	Exibe tutor	Agente Tempo de Avaliação Agente Comportamento	Exibe ao aluno o tutor para o item	Depende de ativação pelo professor. Depende de ativação pelo aluno, na applet Avaliação.

## 5.6 API de objetos de suporte e controle

Envolve os pacotes de classes relacionados com o suporte e controle do ambiente. Por questões didáticas, será apresentado apenas a documentação relativa às classes Unidade de Controle e Applet Avaliação, uma vez que são as classes com funcionalidades significativas para o ambiente.

### 5.6.1 Classe unidadeControle

A Unidade de Controle é a classe base da lógica operacional do Módulo de Avaliação Remota. Suas funcionalidades englobam o controle sobre os agente de tempo da avaliação, alarme da avaliação, controle do Navegador de Questões, Unidade de Suporte à Comunicação e controle sobre tutores.

A unidade de Controle exerce o controle direto sobre a avaliação, bloqueando sua execução, caso programado pelo professor durante a montagem da avaliação.

### 5.6.2 Classe appletAvaliacao

A classe appletAvaliacao instancia a applet suporte do ambiente Interface do

Aluno. A partir desta applet, o aluno poderá consultar orientações iniciais do professor sobre a execução da prova, bem como ativar alarmes e tutores para emprego durante a avaliação.

Para iniciar a avaliação o aluno insere no sistema o seu login e senha previamente distribuídos pelo professor.

A applet descarrega a avaliação do Módulo Controle da Avaliação, armazenando-a em um repositório de objetos locais (ROL) localizado na Unidade de Controle. O aluno é alertado quanto ao tempo disponível para a solução da questão, quando for questão com tempo marcado e com auto recolhimento.

A avaliação e seus objetos (questões e itens) são exibidos à medida que solicitados pelo aluno. A partir da seleção de um item da interface gráfica, o Navegador de Questões é exibido, sendo iniciada a avaliação. Os agentes Tempo da Avaliação, disponibilizados na Unidade de Controle são ativados e a avaliação é iniciada.

O tempo da avaliação pode ser monitorada visualmente, através do conjunto de “leds” na barra inferior horizontal, onde as cores correspondem ao tempo de execução da avaliação.

O checkbox Ativar Tutores permite ao aluno ter acompanhamento dos agentes tutores de avaliação. Caso não seja desejado, estes não auxiliarão o aluno de forma automática.

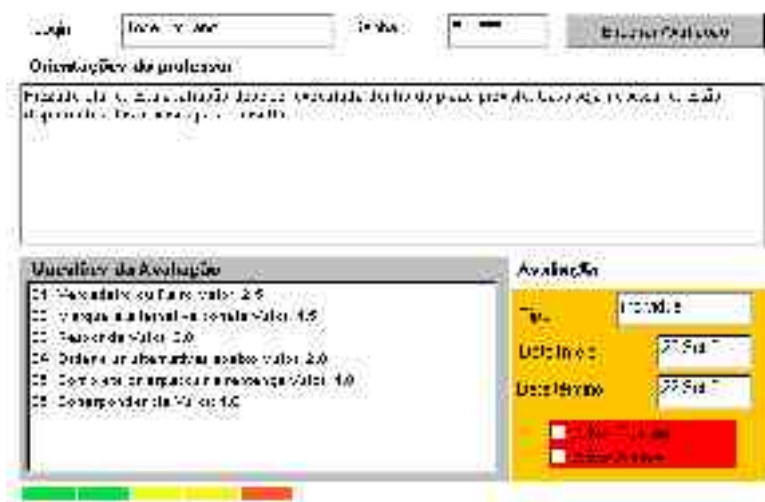


FIGURA 5.14 - O ambiente gráfico da Applet Avaliação.

O checkbox Ativar Alarme permite ao aluno definir com que antecedência deseja ser alertado pelo agente tempo da avaliação quanto à proximidade do final da avaliação.

### 5.6.3 Classe frameQuestao

A classe frameQuestão instancia um ambiente gráfico, chamado Navegador de Questões, capaz de permitir ao aluno visualizar os itens componentes de cada questão da avaliação.

A interface gráfica do Navegador de Questões é exibida na figura 5.15.

Ao ser carregado, o Navegador de Questões permite o acesso aleatório aos itens elaborados pelo professor.

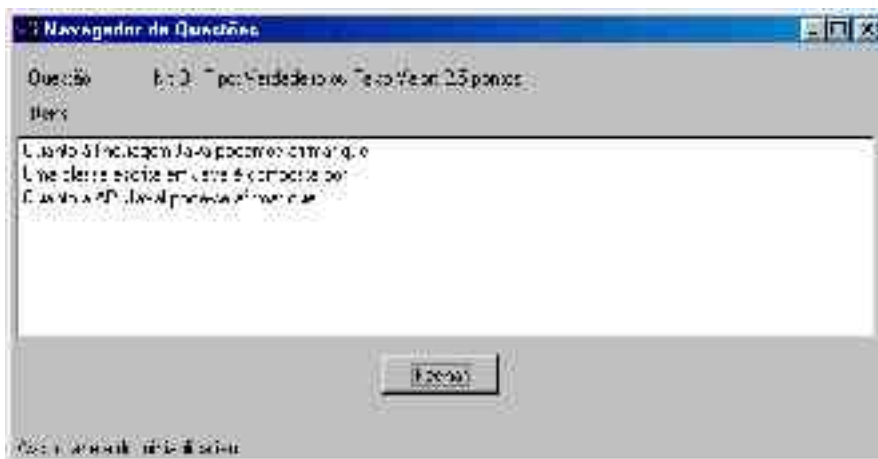


FIGURA 5.15 - O ambiente gráfico do Navegador de Questões.

## 5.7 Aspectos de implementação do ambiente Javal

As classes do ambiente Javal foram escritas tomando por base a API do JDK 1.1.8, tendo em vista problemas de compatibilidade existentes entre a API SDK 2 e as classes componentes da máquina virtual Java da Microsoft (equivalentes à versão 1.0) que equipa os navegadores Internet Explorer 5.0.

Visando evitar o emprego do HTML Converter e a instalação do runtime Java como plugin pelos alunos, resolveu-se abrir mão de algumas facilidades desta nova API.

Foram realizados testes da applet em ambientes de navegadores diversos como StarOffice, Opera, Netscape, Internet Explorer 5.0 com resultados satisfatórios. O tempo de carga da applet (embora empregando arquivos JAR) é um pouco maior que a carga de uma aplicação baseada em script, mas a redução do tráfego observado na rede durante a execução da avaliação pode vir compensar a demora.

Observou-se no entanto, a necessidade de memória para as máquinas cliente. O emprego de múltiplas *threads* em applets e o uso de conexões com a rede consomem recursos do sistema operacional, cujos efeitos podem ser atenuados empregando-se uma memória superior a 32 MB (embora não se note diferença muito grande na performance global).

Este ambiente foi prototipado com base no banco de dados MySQL, mas suas configurações permitem manipular qualquer banco que empregue SQL ANSI como linguagem de consulta. Foram executados testes para uso de bancos Access via ODBC, mas a ponte JDBC-ODBC não se mostrou eficiente para esta conexão e os drivers Java-Access conseguidos não eram gratuitos.

Estuda-se o emprego de uma estrutura baseada em Objetos (RMI) que execute esta tarefa, através de um objeto remoto que efetue a consulta ao banco de dados e retorno dos resultados.

## 5.8 Resumo do capítulo cinco

Neste capítulo foram apresentadas as classes componentes dos diversos pacotes do ambiente Módulo de Avaliação Remota (MAR), visando a apresentação de sua API.

Foi apresentada uma visão geral do ambiente, com o objetivo de identificar grupos com funcionalidades específicas que justificassem a divisão do ambiente em

pacotes. Em seguida, foram apresentadas as principais classes de cada pacote, com seus atributos e métodos, buscando apresentar uma visão sistêmica do conjunto.

Foram apresentados ainda a relação entre as diversas classes da API, procurando identificar as principais mensagens existentes no ambiente e a relação entre os seus agentes, necessária para garantir as funcionalidades desejadas.

Por último foram comentados alguns aspectos de implementação julgados interessantes quanto a necessidades e problemas a corrigir.

O próximo capítulo apresenta um pequeno estudo de caso, elaborado com base na ferramenta desenvolvida.

## 6 Testes realizados e resultados obtidos

O ambiente Javal foi empregado para a avaliação em uma amostragem de 10 alunos que se voluntariaram para o teste, como parte da disciplina de sistemas distribuídos.

As questões abordavam temas ligados à disciplina previamente ministrada pelo professor e as questões eram inéditas. O valor total da prova foi de 4 pontos, considerada como conceito para a disciplina.

Foi empregado o laboratório do NIED (Núcleo de Informática Educativa) para a execução da avaliação, com o material empregado abaixo:

- ⑩ 01 Servidor Pentium II 450 MHz para servidor Web empregando windows NT;
- ⑩ 10 estações Pentium 200 MMX para estações clientes (64 MB RAM);
- ⑩ Internet Explorer 5.0
- ⑩ Não foi utilizado banco de dados (a avaliação foi carregada por scripts em formato texto e os relatórios enviados por e-mail).

A avaliação entregue aos alunos possuía o seguinte formato:

TABELA 6.1 -Formato de avaliação aplicada como teste.

<i>Tipo de questão</i>	<i>Qtde itens</i>	<i>Valor (item)</i>	<i>Tempo Proposto</i>	<i>Tempo Mínimo</i>	<i>Valor total do item</i>
Verdadeiro ou Falso	5	0,4	90 s	20 s	2,0 pontos
Marque X	5	0,4	40s	20s	2,0 pontos
Tempo total da avaliação minutos				15	4.0 pontos

Os resultados obtidos foram mapeados , conforme a tabela 6.2 :

TABELA 6.2 - Mapeamento de resultados obtidos.

<i>Observações</i>	<i>Quantidade</i>
Pontuação	Graus obtidos (máximo 4 pontos): <ul style="list-style-type: none"> <li>⑩ Entre 0 e 2 - 1 alunos;</li> <li>⑩ Entre 2,1 e 3,0 - 4 alunos; e</li> <li>⑩ Acima de 3,0 - 5 alunos;</li> </ul>

O mapa de avaliação da montagem da prova, segundo o relatório gerado pelo ambiente, evidenciou possíveis falhas na elaboração da avaliação, conforme pode-se observar na tabela 6.3.

Como sugestão para trabalhos futuros, ferramentas de mineração de dados podem ser aplicados ao ambiente, visando detecção de problemas pedagógicos ligados à montagem da avaliação.

A seguir, algumas informações rapidamente extraídas dos resultados da avaliação.

TABELA 6.3 - Mapa de resultados da avaliação.

Eventos ocorridos na avaliação.	Identificação dos alunos									
	1	2	3	4	5	6	7	8	9	10
<i>Questões Puladas</i>	4	5	2	7	3	4	4	5	2	8
<i>Respostas Precipitadas</i>	2	1	1	0	0	1	1	0	2	0
<i>Timeouts de item</i>	2	0	3	2	0	0	0	3	1	4
<i>Questões não respondidas</i>	1	0	0	1	0	0	0	0	0	1
<i>Timeouts da avaliação</i>	0	0	0	1	0	0	0	0	0	0
<i>Acertos em VF</i>	16	14	18	11	20	17	16	16	16	10
<i>Acertos em Marque</i>	4	5	4	3	3	3	3	2	4	2
<i>Total de pontos</i>	3,2	3,4	3,4	2,3	3,4	2,9	2,8	2,4	3,2	1,8

Observações realizadas pelo professor:

1. O item nr. 3 da questão nr. 1 deve ser revisada, pois apresentou maior índice de *timeout* (4 alunos estouraram o tempo de solução do item. Apenas 2 acertaram);
2. O item nr. 1 da questão nr. 2 deve ser revisada, pois o tempo para sua solução foi superestimado em 90 segundos, enquanto a média dos alunos a solucionou em 30 segundos;
3. Os itens nr. 5 e nr. 3 da questão 2 . Foram respondidas de forma precipitada por 5 alunos (50%), onde apenas 1 aluno acertou. Possível questão com enunciado impreciso ou prolixo ( “pegadinha”);
4. Apenas 1 aluno ultrapassou o tempo total da avaliação (10%), o que indica que a questão tem seu tempo de solução previsto coerente;
5. Um aluno esqueceu de responder uma questão da avaliação; e
6. Um aluno pulou 80% das questões e, extrapolou o tempo de solução proposto em 4, não respondeu 1 questão. Seu comportamento espelha o grau obtido pelo aluno, o menor grau da avaliação.

Observações dos alunos:

- Em dois computadores a carga da applet demorou mais tempo que o normal, possivelmente devido à tráfego na rede do laboratório;
- Em um deles o Internet Explorer teve de ser reiniciado, devido ao trancamento da applet.

## 6.1 Vantagens do modelo proposto

O modelo proposto pelo ambiente Javal oferece algumas vantagens em relação às

ferramentas de avaliação para EAD comerciais, baseados em tecnologias como ASP, PHP e QML:

TABELA 6.4 - Vantagens do modelo proposto por Javal

<i>Item</i>	<i>Descrição</i>	<i>Sistemas de Avaliação Comerciais</i>	<i>Ambiente Javal</i>
Sigilo das avaliações	Permitir que o aluno possa armazenar questões em seu computador, possibilitando a criação de bancos de questões “paralelo”.	As páginas HTML geradas dinamicamente através de ASP, PHP ou JSP podem ser armazenados pelo browser cliente.	As applets destruídas após as avaliações não podem ser salvas.  O método <i>destroy</i> da applet permite que o objeto seja destruído. A página HTML que contém a applet não armazena a avaliação.
Controle de execução da avaliação	Característica que permite o controle do servidor sobre os clientes durante a execução da avaliação	Até o presente momento, não implementado pelos sistemas avaliados.  Os controles existentes estão ligados à restrições de submissão de respostas, após o tempo de execução da avaliação.	As applets possuem métodos que permitem o autorecolhimento após esgotado o tempo de avaliação.
Tráfego de informações na rede	Quantidade de informações que trafegam na rede durante a execução de uma avaliação	Para cada item da avaliação é efetuado o download da referida página, originando tráfegos servidor/cliente tantos quantos forem os itens da avaliação.  A cada resposta, efetuada pelo aluno, trafega uma inserção da resposta no Banco de Dados no servidor	A applet descarrega todo o conteúdo da avaliação de uma única vez.  O tráfego de informações do cliente para o servidor ocorre apenas ao término da avaliação. A conexão telefônica pode ser desfeita e o aluno não perde a avaliação.
Inteligência embarcada	Característica que permite introdução de agentes inteligentes capazes de executar atividades no ambiente do aluno durante a avaliação	Não disponível	Agentes especialistas em tempo e comportamento permitem controlar a execução da avaliação e auxiliar na solução das questões, agindo como tutores embarcados.
Suporte à atividades colaborativas	Característica que permite a execução de uma atividade em grupo, controlada automaticamente por um agente	Não disponível	O ambiente MAR oferece suporte à avaliações colaborativas ou em grupo, pelo desenvolvimento de agentes especialistas.

## 6.2 Melhorias no modelo proposto

O modelo apresentado no presente protótipo necessita de um trabalho de pesquisa visando torná-lo mais funcional. Seguem relatados alguns problemas identificados bem como sugestões para futuras versões.

Foi observado um problema na conexão de applets diretamente com o servidor de banco de dados MySQL, empregado para simulação. Este problema deve-se a limitações de segurança imposta pelo ambiente de runtime da máquina virtual Java do



Internet Explorer 5. Para contornar este problema, iniciou-se o desenvolvimento de uma aplicação servidora, denominada RAT, capaz de prover a necessária comunicação, recebendo conexões das applets e realizando consultas no BD. Está sendo estudado a viabilidade do emprego de RMI como suporte para este tipo de conexão ao banco de dados.

Estuda-se a possibilidade de incluir sons no ambiente, tornando o alarme do agente tempo não apenas visual, mas sonoro. Da mesma forma, mensagens de estímulo para o aluno nas questões que possuam resposta única (marque a alternativa correta, por exemplo) também estão sendo estudados.

Torna-se necessário a inclusão da funcionalidade que permita ao aluno “marcar” questões que tenha dúvida para posterior solução, visando identificar questões não resolvidas ou que o mesmo deseje rever novamente, selecionadas através da função “marcar”.

Torna-se necessário um estudo no sentido de se personalizar a linguagem do agente, através de dicionários de linguagens, conforme descrito neste trabalho, de forma a permitir uma adaptação ao tipo de aluno.

Avaliações síncronas - Este protótipo foi desenvolvido para a execução de avaliações assíncronas. Torna-se necessário o desenvolvimento de classes capazes de permitir o desenvolvimento de avaliações sincronizadas, onde o professor possa estar presente on-line para auxiliar os alunos.

Comunicações grupais - O presente modelo propõe, mas não implementa soluções para comunicações grupais. A unidade de suporte de comunicação presente neste protótipo implementa apenas classes que permitem a comunicação do ambiente com o servidor da aplicação. Torna-se necessário estudos capazes de permitir a agregação de ferramentas (salas de reuniões, chats em Java, white boards, etc.) visando permitir maior interoperabilidade entre agentes de forma colaborativa, bem como permitir a comunicação e interação entre alunos.

## 7 Conclusão

O presente trabalho buscou apresentar um modelo de ambiente para avaliação em Ensino à Distância. Diferentemente dos sistemas comerciais, o modelo proposto por Javal oferece uma interface gráfica baseada em Java, cuja inovação está no suporte a um conjunto de agentes e na lógica baseada no cliente.

Os agentes possuem funcionalidades capazes de permitir a monitoração do comportamento do aluno durante a realização de avaliações à distância, visando auxiliá-lo na solução de questões. Adicionalmente, fornecem ao professor “*feedback*” sobre a avaliação entregue ao aluno, não apenas em aspectos somativos, mas formativos, contemplando o aspecto comportamental do aluno.

Estes agentes foram capazes de interagir com o aluno através de sensores disponibilizados na interface gráfica do ambiente, capturando eventos e calculando intervalos de tempo, visando mensurar aptidões e detectar possíveis situações de fuga ou dúvida do aluno. As informações obtidas podem servir de base para o início de procedimentos de auxílio e proposta de modificações na estratégia de ensino ou da avaliação, bem como para ajuste de perfis de alunos.

A proposta apresentada por Javal permite que a lógica seja embarcada e entregue ao aluno através do emprego de applets, capazes de oferecer suporte necessário para a execução das questões, aplicação, correção e funcionamento dos agentes. O seu emprego, em vez de páginas HTML geradas dinamicamente, oferece algumas vantagens interessantes, como :

A possibilidade de se introduzir novos agentes pedagógicos embarcados no ambiente aluno;

- A redução do tráfego de informações pela rede, haja visto que a avaliação é descarregada uma vez e apenas o relatório é retornado ao servidor;
- Segurança e maior controle na aplicação de avaliações, pois a avaliação pode ser automaticamente recolhida ao seu término; e
- Menor custo de desenvolvimento e implementação.

A originalidade do modelo encontra sua base na redescoberta do uso de applets, antes utilizadas apenas para poucas aplicações específicas, como preenchimento de formulários em páginas Web e animações de imagens, vislumbrando seu emprego em uma interface “inteligente” para o aluno, capaz de estender para a *Internet* os sentidos do professor, contribuindo para avaliações remotas precisas e adaptadas ao perfil do aluno.

### 7.1 Novos conhecimentos

O presente trabalho constituiu-se em uma ferramenta capaz de introduzir um novo vetor no ambiente de avaliação remota: o componente comportamental. Com base nos experimentos realizados, pode-se deduzir que nem sempre o professor é capaz de produzir enunciados para questões totalmente compatíveis com os perfis do aluno.

Esta diferença, em tempos de avaliação presencial, era considerada como interpretação do aluno, dentro da idéia de que “a interpretação faz parte da questão”.

O módulo MAR permitiu mensurar a necessidade do aluno de realizar consultas durante a solução de questões, visando a construção do seu próprio

conhecimento. O professor também pode avaliar a precisão de suas questões, a clareza da elaboração de seus enunciados e a sintonia da avaliação com os objetivos propostos e a estratégia adotada.

A quantidade de intervenções feitas por agentes durante a solução de questões sinaliza que devam ser conduzidos esforços visando o seu aprimoramento, acrescentando-lhes mais funcionalidades e permitindo amplo espectro de atuação, como a sua atuação de forma colaborativa.

Finalmente, foi possível utilizar applets como suporte para o desenvolvimento deste tipo de ambiente. Suas características operacionais permitiram a elaboração do ambiente Javal e abrirão espaço para novos trabalhos a serem desenvolvidos nesta área.

## **7.2 Trabalhos futuros**

O sistema Javal é um ambiente de arquitetura aberta. As melhorias descritas no presente trabalho são sugeridas para trabalhos futuros. Adicionalmente, podem ser incorporadas ao ambiente as seguintes idéias:

- Adaptatividade dos agentes tutores a perfis de aluno - um estudo sobre aspectos e comportamentos ligados ao perfil considerado;
- Ferramentas de mineração de dados podem ser aplicados ao ambiente, visando detecção de problemas pedagógicos ligados à montagem da avaliação;
- Emprego de agentes animados para uso em tutores embarcados - como extensão do tema anteriormente descrito; e
- Interoperabilidade entre agentes tutores para avaliações em grupo - visando a solução combinada e conjunta de avaliações remotas.

## Bibliografia

- [ALB98] ALBERTSON, Tom. **Best practices in distributed object application development: RMI, CORBA and DCOM**. 1998. Disponível em <[http://www.developer.com/news/techfocus/022398\\_dist1.html](http://www.developer.com/news/techfocus/022398_dist1.html)>. Acesso em 20 maio 1999.
- [BLO64] BLOOM, Benjamin S. ; MESIA, Bertram B. ; KRATHWOHL , David R. **Taxonomy of Educational Objectives**. New York. David McKay,1972.
- [BRO97] BRODY, Linda E.; MILLS, Carol J. Gifted Children with Learning Disabilities:A Review of the Issues. **Journal of Learning Disabilities**, London , v.30 , n.3, p.282-286. Ed Maio/Junho 1997.
- [CAR01] CARDOSO, Rogerio F.,LIMA, José V. **AvalWeb – Sistema interativo para gerência de questões e aplicação de avaliações na Web**. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 12., 2001, Espírito Santo. **Anais...** Espírito Santo: UFES, 2001.
- [DOU96] DOURISH, Paul. **Open Implementation and Flexibility in CSCW Toolkits**. 1996. Tese de Doutorado. University College London, London.
- [DRI99] DRINGUS, L.; TERREL, S. The Framework for Directed Online Learning Environment. **The Internet and Higher Education**.ISSN: 1096-7516. Florida,Vol 2, Nr 1, p 58, May 1999.
- [DUF92] DUFFY,T.M.; JONASSEN, D.H. **Construtivism and the technology of instruction : a conversation**. London: Lawrence Erlbaum, 1992.
- [EDU00] EDUWEB. **Guia de Referência do AulaNet**. 2000. Disponível em : <<http://www.eduweb.com.br>>. Acesso em maio 2001.
- [EMI00] EMILIANO, José P. **Um estudo comparativo sobre as estratégias de segurança implementadas pela linguagem Java**.2000. Trabalho Individual (Mestrado Profissional em Engenharia da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- [EYS77] EYSENCK , Michael W. ; KEANE, Mark T. Review : **Cognitive Psychology - A students handbook** . 3 rd ed. London. Psychology Press, 1997. p. 355 - 370.
- [FRA97] FRANKLIN, Stan; GRAESSER, Art. **Is it an Agent, or just a Program? A Taxonomy for Autonomous Agents**. Memphis. Institute for Intelligent Systems University of Memphis, 1997. Disponível em: <<http://www.msci.memphis.edu/~franklin/AgentProg.html>>. Acesso em : 05 jun 2001.
- [GAR89] GARDNER, H.; HATCB, T. Multiple intelligences go to school: educational implications of the theory of Multiple Intelligences. **Educational Researcher**. London, v.18, n.8. p.4-10, June 1989.

- [GEY01] GEYER, Cláudio F. R.; RODRIGUES, Alessandra P.; EMILIANO, José P. et al. **SEMEAI-Sistema Multiagente de Ensino Aprendizagem na Internet**. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 12., 2001, Fortaleza. **Anais...** Fortaleza: UFC, 1998. Artigo apresentado no Simpósio Brasileiro de Informática na Educação (SBIE), 2001.
- [GOL98] GOLDBERG, Murray W. **An Update on WebCT (World-Wide-Web Course Tools) – a Tool for the Creation of Sophisticated Web-Bases Learning Enviroments**. Disponível em: <<http://homebrew1.cs.ubs.ca/webct/papers/nawweb/full-paper.html>> (1998)
- [HIL98] HILTZ, S. **Collaborative Learning in Asynchronous Learning Networks: Building Learning Communities**, In: WEB 98 SYMPOSIUM, 1998, Orlando, Fl. **Proceedings...** Disponível em: <[http://eies.nijt.edu/~hiltz/collaborative\\_learning\\_in\\_asynch.htm](http://eies.nijt.edu/~hiltz/collaborative_learning_in_asynch.htm)>. Acesso em 10 out 2000.
- [JAC99] JACQUES, P.; **Agentes de Software na Monitoração da Colaboração em Ambientes Telemáticos de Ensino**. 1999. Dissertação de mestrado. (Mestrado em Ciência da Computação) – PUCRS, Porto Alegre. Disponível em: <<http://www.inf.ufrgs.br/~pjaques/>>. Acesso em: 20 mai 2001.
- [LEI01] LEIVA, W.D. et al. **Glossário de Termos Educacionais**. Disponível em: <<http://www.icmc.sc.usp.br/~wdl/sapiens/educglos.htm>>. Acesso em 20 dez 2001.
- [PER98] PEREIRA, Adriana Soares; D'AMICO, Carmem B.; GEYER, Claudio F.R. **Uma Aplicação de Ensino Orientada a Agentes na Internet**. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 9., 1998, Fortaleza. **Anais...** Fortaleza: UFC, 1998.
- [PER99] PEREIRA, Adriana Soares. **Um agente para seleção de estratégias de ensino em ambientes educacionais na internet**. 1999. Dissertação de Mestrado. (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- [QMP99] QUESTION MARK COMPUTING. **Question Mark Perception Documentation**. 1999. Disponível em <<http://www.qmark.com>>. Acesso em 12 out 2000.
- [REI00] REIS, Alessandro Boeira. **Um modelo do Aluno Adaptativo para Sistemas na Web**. 2000. Trabalho Individual (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre
- [ROD00] RODRIGUES, Alessandra Pereira. **O processo avaliação do ensino e aprendizagem em Educação à Distância**. 2000. Trabalho Individual (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- [TAR99] TAROUCO, L. M. R. **Ambiente de Suporte para Educação à Distância**. In: WORKSHOP EM INFORMÁTICA NA EDUCAÇÃO, SBC, Porto Alegre, 1999. Programa de Pós-Graduação em Informática na Educação da Universidade Federal do Rio Grande do Sul. Disponível em <<http://penta.ufrgs.br/pgie/workshop/ambiente.htm>>. Acesso em 13 fev 2001.

- [TUR95] TUROFF, Murray. **Designing a Virtual Classroom [TM]**. In : INTERNACIONAL CONFERENCE ON COMPUTER ASSISTED INSTRUCTION, ICCAI, 1995. **Proceedings...** Disponível em: <<http://www.njit.edu/njit/Department/CCCC/VC/Papers/Design.html> >. Acesso em 3 out 2000.