

Um Sistema de Alertas Inteligentes para Ambientes de Ensino a Distância na Internet

Daniela Leal Musa, José Palazzo Moreira de Oliveira

Instituto de Informática – Universidade Federal do Rio Grande do Sul (UFRGS)
Caixa Postal 15.064 – 91.501-970 – Porto Alegre – RS – Brazil

{musa, palazzo}@inf.ufrgs.br

Abstract

This work proposes an alert system to help in the process of evaluation of students in a Web-based distance learning environment. Such system detects problems occurred in the students activities and proceeds with corrective actions to solve the detected problems. The work is integrated to the Tapejara environment developed at the Institute of Informatics at UFRGS - Web-based Teaching Intelligent Systems - which consists in a system for construction and assistance of courses available in the Web.

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

Resumo

O presente trabalho propõe um sistema de alertas inteligentes para auxílio ao processo de avaliação dos alunos em ambientes de Ensino a Distância na Web, que detecta problemas nas atividades dos alunos e realiza ações corretivas adequadas. Este trabalho está integrado ao ambiente Tapejara do Instituto de Informática da UFRGS – Sistemas Inteligentes de Ensino na Web - que consiste em um sistema de construção e acompanhamento de cursos disponibilizados via Internet

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

1 Introdução

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

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

Na maioria dos ambientes ou cursos a distância disponíveis na Web [14,17], um dos problemas detectados é a falta de mecanismos que possibilitem o professor realizar um acompanhamento completo e abrangente das atividades dos alunos, diagnosticando o seu nível de conhecimento bem como seu ritmo de aprendizagem.

Normalmente, este processo de avaliação dos alunos na Web é realizado apenas através de informações referentes a testes, exercícios ou relatórios extensos que apresentam um histórico dos acessos feitos às páginas do curso [17].

Por esta razão, é muito importante a existência de ferramentas apropriadas que possibilitem o monitoramento do que está sendo feito por cada estudante, detectando situações de exceção e avisando ao professor os problemas que estão ocorrendo com alguns alunos. Com isso, o professor terá acesso às peculiaridades de cada um, e ser-lhe-á possibilitado adequar as aulas às individualidades podendo, ele, dessa forma, exercer um acompanhamento mais próximo, estimulando e orientando o aluno, sem que isso acarrete um ônus maior.

Em alguns ambientes de ensino via Internet, o comportamento dos alunos é avaliado pelo uso da tecnologia de agentes da Inteligência Artificial, mais especificamente Sistemas Multiagentes [6], [19]. Um agente monitora os processos de interação dos alunos com o ambiente do curso, que reflete o seu processo de aprendizagem no curso [24]. Quanto maior for a quantidade de informações sobre o comportamento de cada aluno, os tutores virtuais poderão auxiliar os alunos mais adequadamente. Na maioria dos ambientes de ensino que utilizam técnicas de monitoração do comportamento dos alunos, as informações necessárias para esta descoberta são retiradas de uma análise dos registros dos *logs* de acessos ao servidor Web de ensino a distância [1,16, 21]. Tais registros podem fornecer o comportamento de cada aluno no tocante aos acessos aos arquivos (páginas), armazenados no servidor Web. Porém, é preciso realizar várias consultas nestes registros para se obterem informações que sejam importantes e que possam auxiliar a descobrir o comportamento do aluno. Um sistema de alertas pode ser usado para realizar estas consultas e monitorar o aluno no momento em que ele está realizando o curso, o que auxiliará os tutores virtuais a avaliarem o comportamento de cada aluno.

O objetivo deste trabalho é a criação de módulos de alertas inteligentes que irão monitorar as atividades dos alunos em Sistemas de Ensino na Web, mais especificamente no projeto Tapejara da Universidade Federal do Rio Grande do Sul. O projeto Tapejara – Sistemas Inteligentes de Ensino na Internet [22], desenvolvido dentro do Programa ProTeM-CC do CNPq, é executado em parceria entre pesquisadores da área de Sistemas de Informação e Inteligência Artificial do Instituto de Informática da Ufrgs, Faculdade de Psicologia da Unisinos e por profissionais da área de

capacitação de recursos humanos da CRT Brasil Telecom, os quais visam desenvolver cursos de treinamento a serem ministrados, remotamente, à seus funcionários em diferentes cidades e em diferentes períodos. A proposta acadêmica, deste projeto consiste em projetar e implementar um sistema de construção e acompanhamento de cursos a serem disponibilizado na rede Internet, permitindo ensino a distância de forma assíncrona. Esta metodologia destaca o uso de técnicas pedagógicas que levem em conta aspectos cognitivos de aprendizagem, bem como a arquitetura multiagente para construção de sistemas tutores inteligentes [20].

Este artigo descreve o Sistema de Alertas, que monitora o aluno na tentativa de descobrir e tratar comportamentos fora do previsto, para assegurar que ele aproveite o curso satisfatoriamente. O artigo está estruturado da seguinte forma: Inicialmente, na seção 2, é apresentada a descrição do Sistema de Alertas desenvolvido. Na seção 3 o funcionamento do Sistema de Alertas é explicado, assim como a forma como ele interage com os demais componentes que compõem o ambiente Tapejara. A seção 4 apresenta o diagrama de classes do protótipo implementado. O Sistema de Alertas e os aspectos da implementação do protótipo são descritos na seção 5. A seção 6 apresenta o estudo de caso realizado para a validação do mecanismo de alertas proposto neste trabalho. Finalmente, as conclusões deste estudo, assim como os trabalhos futuros a serem realizados com base nos estudos e implementação desenvolvidos nesta etapa, são apresentados na seção 6.

2 Descrição do Sistema de Alertas

Um sistema de alerta é um programa que monitora um banco de dados e informa para alguns usuários ou para um programa quando uma determinada condição ocorre, oferecendo recursos para tomada de decisões [11].

Um sistema de alertas inteligentes é tipicamente um sistema ativo. Isto significa que o sistema deve reagir quando alguma situação especial ocorre e gerar avisos (*warnings*) que podem ser úteis no processo decisório [4]. O usuário que especifica as condições que devem ser monitoradas e as ações que devem ser realizadas na ocorrência destas condições. Segundo [8], alertas também podem ser enviados se um determinado evento não ocorrer.

O Serviço de alertas desenvolvido é composto pelos seguintes módulos: Editor de Alertas, Base de Conhecimento, Base de Mensagens, Monitor de Eventos, Servidor de Alertas e Base de Dados. A Figura 1 apresenta a arquitetura do Sistema de Alertas.

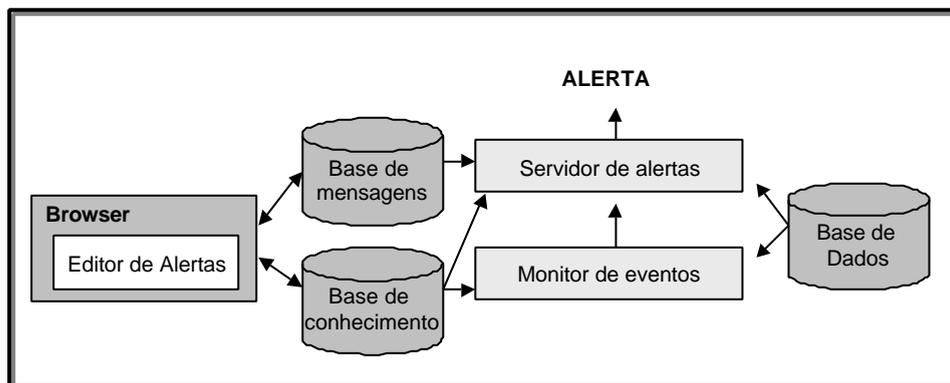


Figura 1 - Arquitetura do Sistema de Alertas

Editor de alertas: módulo de construção e gerenciamento dos alertas. Através de uma página Web, o usuário (professor, coordenador do curso, etc.) pode configurar e ajustar as regras de geração de alertas, identificando os tipos de alertas desejados. Isto deve ser realizado por um especialista de conhecimento, o qual deve conhecer muito bem as regras e condições a serem criadas. É aqui que ele registra no sistema se necessita ser informado quando uma certa condição de exceção ocorrer.

A condição de exceção e a ação de envio da mensagem aconselhável, assim como o receptor e o meio de recebimento da mensagem, também são configuradas neste formulário. As regras podem ser agendadas para serem executadas em qualquer intervalo de tempo.

Base de conhecimento: representa as situações de exceção que devem ser detectadas e geralmente são representadas através de regras. Uma regra é, simplesmente, um par condição-ação: dada a existência da condição expressa, faça a ação. As regras são expressas da seguinte forma: SE <condição> ENTÃO <ação> e utilizam uma gramática criada especificamente para o Sistema de Alertas. Esta gramática pode ser encontrada na dissertação de Mestrado da autora [14]. Esta gramática foi criada após estudos principalmente de [3, 9]. Novas regras podem ser adicionadas à base de conhecimento assim como regras antigas podem ser removidas ou modificadas. Sabe-se que menos regras implicam geralmente um menor número de casos detectados [5]. Apesar deste trabalho diferir base de dados e base de conhecimento, é importante ressaltar que as regras estão sendo armazenadas, também, em um banco de dados relacional. A terminologia “base de conhecimento” continuará sendo adotada para identificar claramente o tipo de informação que está sendo discutida.

Monitor de eventos: é o módulo que detecta a ocorrência de um evento de interesse e avisa o módulo Servidor de Alertas. Os eventos detectados pelo Sistema de Alertas são de três tipos: mensagens, eventos temporais e eventos temporais associados com mensagens. Um exemplo de mensagem recebida pelo sistema de alertas é o aviso de *logon* no sistema pelo aluno.

Servidor de alertas: este módulo processa as regras associadas com os alertas, contidas na base de conhecimento, acessa a base de dados e, no momento que ocorre um evento crítico, emite o alerta correspondente, definido anteriormente através do editor de alertas.

Base de dados: contém todas as informações sobre a atividade dos usuários no curso e refere-se às sessões e à seqüência de páginas acessadas pelos alunos. Dentre as informações, destacam-se a data e hora de acesso, tempo despendido em cada página e sessão, seqüência de navegação do usuário, contendo informações sobre cada página acessada como título da página, URL, data e hora de acesso, entre outras. Estes dados podem ser utilizados na avaliação do aluno e para geração dos alertas, na medida em que os dados armazenados sofrerem mudanças acima dos limites estabelecidos durante a atividade do aluno. A geração destas informações começa a ser produzida a partir do momento que o usuário inicia a sua sessão de estudo. A partir daí, qualquer operação efetuada é armazenada na base de dados.

Base de mensagens: contém todas as mensagens que podem ser enviadas pelo Sistema de Alerta. As mensagens são definidas no Editor de alertas.

3 Integração do Sistema de Alertas com o projeto Tapejara

O Sistema de Alertas desenvolvido nesta dissertação de mestrado está integrado com o sistema multiagente do Projeto Tapejara. Para um melhor entendimento do escopo deste trabalho, serão discutidas a seguir características relevantes deste ambiente para o Sistema de Alertas.

3.1 Sistema Multiagente no Projeto Tapejara

A sociedade de agentes do Projeto Tapejara propõe agentes autônomos que possam se comunicar. As principais tarefas dos agentes são: analisar o comportamento do aluno a partir de um arquivo de registro de atividades (*log*), onde são registrados todas as suas interações com o ambiente de ensino, realizar o diagnóstico cognitivo do aluno, selecionar uma estratégia de ensino, realizar ajustes na estratégia selecionada [13]. As funcionalidades mencionadas acima são divididas entre os agentes de acordo com a arquitetura mostrada na Figura 2.

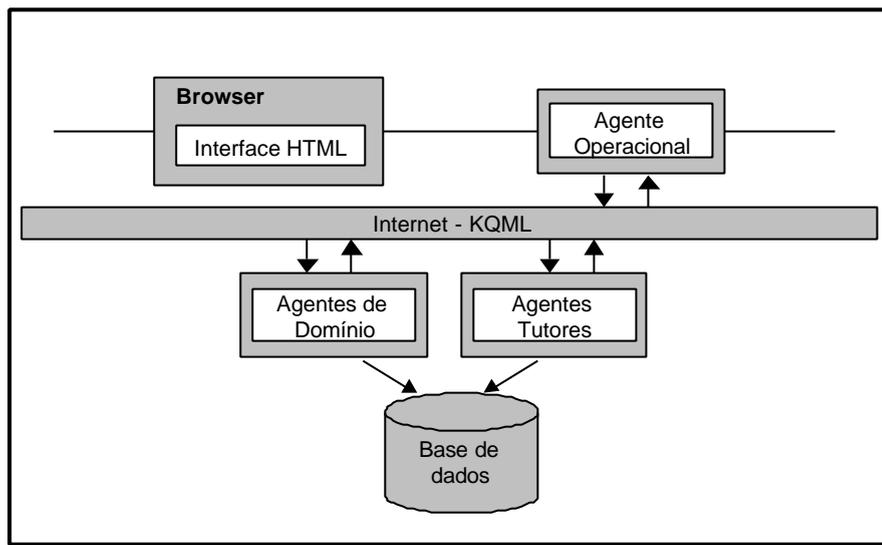


Figura 2 – Sistema Multiagente do Tapejara

O ambiente contempla uma sociedade de agentes comprometidos com o aprendizado cooperativo, envolvendo os seguintes tipos de agentes [13]:

- **Agente Operacional:** o agente operacional possui como características a reatividade, habilidade social, autonomia e continuidade temporal. Sua principal função é repassar as mensagens de requisições do aluno aos demais agentes e gravar as requisições deste no *log*;
- **Agente Domínio:** responsável por recuperar o material instrucional e enviá-lo ao aluno que o requisitou. Como características básicas, este agente é colaborador, orientado por objetivo e possui continuidade temporal. Além disso, este agente será responsável por avaliar as respostas nos exercícios e revisões avaliativas;
- **Agente Tutor:** este agente é o responsável por recuperar a trajetória do aluno (no registro de atividades) e a partir desta, gerar um plano de ensino individualizado. Este agente é responsável, por selecionar automaticamente estratégias de ensino adaptadas às características individuais do aluno, segundo um modelo de aluno pré-definido. Como características básicas, este agente é colaborador, adaptável e possui continuidade temporal.

A comunicação entre os agentes segue um padrão reconhecido mundialmente [10], sendo realizada

através de mensagens KQML (Knowledge Query and Manipulation Language) [7, 9] buscando, assim, construir uma arquitetura de agentes robusta e padronizada, que permita a reutilização de códigos para os diferentes tipos de agentes.

A KQML é uma linguagem, desenvolvida pelo projeto DARPA *Knowledge Sharing Effort* [7], para troca de informações e conhecimento entre agentes e caracteriza-se, tanto por uma formatação de mensagens, como por um protocolo, para manipulá-las, suportando, em tempo de execução, o compartilhamento de conhecimento e informações entre agentes.

3.2 Funcionamento do Sistema de alertas com Tapejara

O Sistema de Alertas serve-se de um conjunto de dados de entrada, chamados de índices de aprendizagem, para descobrir as situações que necessitam do envio das mensagens de alerta. Alguns índices são registrados no banco de dados pelo Agente Operacional Tapejara durante a interação do aluno no curso, e outros são resultados de análises sobre os dados armazenados. Estes índices foram definidos juntamente com o Grupo de Pesquisa em Processos Cognitivos da Unisinos [23], parceiro do projeto, e são mostrados na tabela 1. Alguns índices necessitam de uma avaliação humana mais complexa. Logo, do conjunto de índices apresentado, alguns não serão utilizados pelo sistema de alertas.

Categoria	Índices
Percurso	<ul style="list-style-type: none"> <input type="checkbox"/> Total de páginas acessadas no módulo. <input type="checkbox"/> Total de retornos simples – volta à página anterior ou à seqüência anterior. Ex. A → B e B → A
Movimentação da Navegação	<ul style="list-style-type: none"> <input type="checkbox"/> Paradas <ul style="list-style-type: none"> - Paradas Bruscas– Trocas rápidas de páginas durante uma seqüência. - Paradas Grandes– Pelo tempo de permanência na página. <input type="checkbox"/> Aceleração e desaceleração - significando facilidade ou dificuldade de assimilar o conteúdo, respectivamente. Pode ser verificado pelo tempo de permanência. <input type="checkbox"/> Mudanças bruscas de níveis (conteúdo, exemplos e exercícios). Ex. O aluno está lendo um conteúdo e bruscamente vai para um exercício e começa a acertar todos os exercícios. <input type="checkbox"/> <i>Chunks</i> - Blocos de resolução fechados numa seqüência de navegação com cortes. Ex. Con/Exem/Exerc Exem/Exer <input type="checkbox"/> Retroações – Quando o aluno se depara com uma dificuldade em uma página 'X' em função da interferência devido a um não entendimento das páginas anteriores que o impede de prosseguir e que acaba fazendo-o retornar às páginas anteriores. Ex. A à B à C à D parada D à C à B <input type="checkbox"/> <i>Loopings</i> - Fazer idas e voltas numa seqüência determinada. A à B à C à A à B à C3. <input type="checkbox"/> Repetição de <i>Scripts</i> – conjunto de procedimentos repetidos e 'empacados' num domínio específico de ações cognitivas. <input type="checkbox"/> Esquemas Mentais – são verificados pelos conjuntos organizados de seqüências de navegação.

Tabela 1 – Índices de aprendizagem

O Sistema de alertas é tipicamente um sistema ativo. Isto significa que o sistema detecta eventos e deve reagir quando alguma situação especial (evento de interesse) ocorre e gerar avisos. Os eventos percebidos pelo Sistema de Alertas são ações realizadas pelos alunos e estão sendo controladas e armazenadas na base de dados pelo Agente Operacional Tapejara.

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

- Mensagens: o Sistema de alertas fica ocioso à espera de algum aviso do Agente Operacional que seja interessante monitorar. Este aviso pode ser uma mensagem que informa quando o aluno loga-se no sistema, termina algum módulo, realiza algum exercício ou a percepção de transições de estado no banco de dados como a inclusão de um novo registro (inserção de nota de avaliação), ou a alteração de algum dado importante.
- Eventos Temporais: o Sistema de alertas realiza consultas no banco de dados de acordo com as especificações de tempo registradas no Editor de alertas, apresentado na seção 2. Se alguma condição de tempo for satisfeita, a regra associada é disparada e o alerta enviado.
- Eventos Temporais associado com Mensagens: neste caso, após inclusão ou alteração em algum registro de interesse, o sistema é avisado pelo Agente Operacional e, antes de disparar a regra, verifica se a condição temporal também é satisfeita, determinando se a ação estipulada para esta regra pode ser disparada.

O funcionamento do Sistema de alertas integrado no ambiente multiagente do Tapejara é apresentado na figura 3.

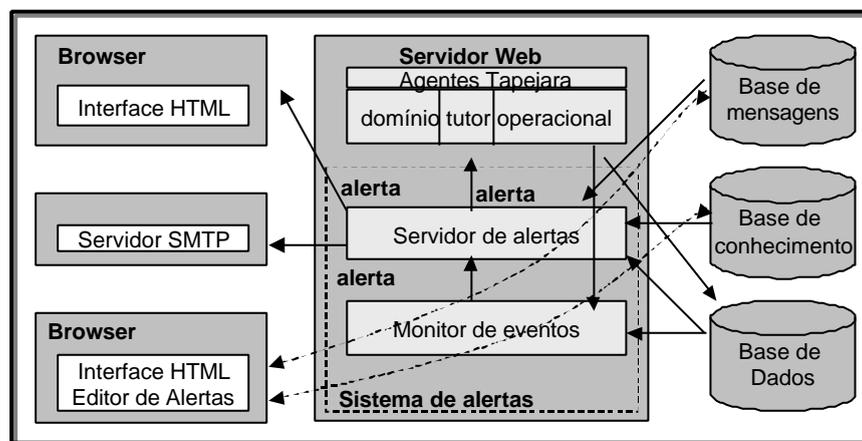


Figura 3 - Integração do Sistema de Alertas com o projeto Tapejara

Após a definição, via Editor de alertas, dos alertas e mensagens a serem enviadas, o Sistema de alertas, através do módulo Monitor de Eventos, monitora alterações na base de dados para detectar ocorrências de eventos de interesse assim como recebe aviso de ocorrências de eventos pelo Agente Operacional-Tapejara. A comunicação entre o Sistema de Alertas e os agentes-Tapejara é realizada via KQML.

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

O Monitor de eventos, através das informações obtidas da base de conhecimento, sabe quais os eventos que interessam. Quando um evento ocorre, o monitor de eventos informa o Servidor de Alertas. Este checa se existe alguma regra/condição associada para este evento na base de conhecimento. Se encontrada, ele verifica e realiza a ação indicada. Esta ação pode ser de três tipos:

- Operação de envio de mensagem no browser do aluno: o sistema envia uma mensagem para o aluno durante a sua navegação no curso, alertando-o de alguma situação. Esta mensagem pode ser dicas, perguntas, incentivo, congratulações ou avisos.
- Operação de envio de mensagem KQML para agente: uma mensagem KQML é enviada para o agente Tapejara.
- Operação de envio de e-mail para professor, aluno ou autor/responsável pelo material didático

Todas as mensagens enviadas estão armazenadas na Base de mensagens apresentada na seção 2.

4 Diagrama de Classes do Protótipo Implementado

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

Estes objetos são:

1. Alerta: guarda um conjunto de objetos do tipo *Condição*, que representam cada uma das condições que fazem a lógica do alerta ser verdadeira.
2. Invocar: guarda as condições em que um alerta será ativado.
3. Observador: apresenta uma visualização do resultado do alerta.

Cada um destes objetos será descrito a seguir:

4.1 Classes Alerta e Condição

O serviço de alertas usa um conjunto de regras como formalismo de representação de conhecimento. A classe *Alerta* contém informação de documentação, como título, autor, versão, etc. e é composta por um ou mais objetos *Condição*. O alerta é gerado sempre que uma condição representada em um objeto *Condição* se torna verdadeira. Cada condição é representada por um texto em uma linguagem criada para expressar premissas de regras. Regras como "a nota do aluno é maior do que 7" podem ser expressas usando esta linguagem. A gramática desta linguagem é apresentada em [14]. O objeto *Condição* tem um método *VerificaSintaxe*, que é usado para verificar a sintaxe da propriedade *texto_regra*, e um método *VerificaExpressão*, usado para verificar se a expressão na propriedade *texto_regra* é verdadeira. O método *Executar* do objeto *Alerta* executa cada método *VerificaExpressão* para cada objeto *Condição* e, se ao menos uma delas for verdadeira, deve-se fornecer o tempo(data e hora) em que o alerta foi gerado e algum dado (como uma frase "nota do aluno é 8", por exemplo). A figura 3 mostra as classes *Alerta* e *Condição*.

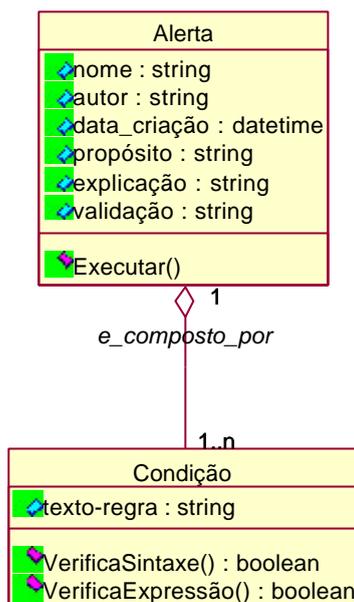


Figura 4 - As classes Alerta e Condição

4.2 Classes CondiçãoTempo, CondiçãoMensagem e CondiçãoTempoMensagem

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

- Quando uma condição temporal é verdadeira. Por exemplo, um alerta pode ser invocado a cada 5 horas, iniciando em 5/3/01, às 12:00, até 10/3/01, às 15:00. Este tipo de invocação pode ser expresso pelo objeto *CondiçãoTempo*.
- Quando uma mensagem é gerada. Por exemplo, um alerta poderia ser invocado quando um novo dado do aluno está disponível. Este tipo de invocação pode ser expressa pelo objeto *CondiçãoMensagem*.
- Quando uma mensagem é gerada e uma restrição de tempo é verdadeira. Por exemplo, um alerta poderia ser invocado 5 horas depois que o novo dado do aluno está disponível, durante 5 semanas. Este tipo de invocação pode ser expressa pelo objeto *CondiçãoTempoMensagem*.

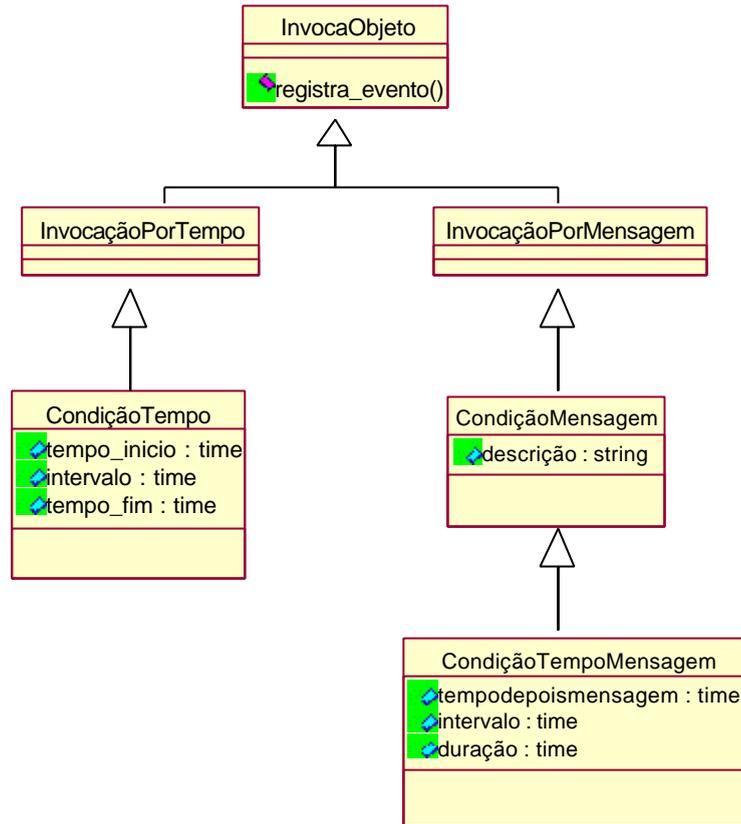


Figura 5 - As classes CondiçãoTempo, CondiçãoMensagem e CondiçãoTempoMensagem

4.3 Classe Observador

Este objeto possui um método *EnviaMensagem* que mostra uma mensagem baseada na conclusão do alerta. O *observador* é muito dependente da implementação. Se um sistema deseja enviar um *e-mail* para o professor responsável pelo curso com a conclusão do alerta, um novo *observador* deve ser criado para suportar esta característica. De fato, um *observador* pode ser usado em vários objetos deste modelo.

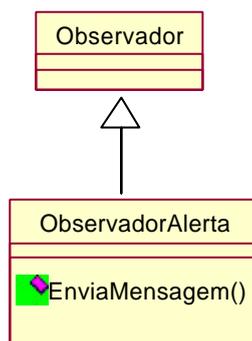


Figura 6 - A classe Observador

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

5 Ambiente de Implementação do Sistema de Alertas

O protótipo foi totalmente desenvolvido utilizando-se a linguagem Java, linguagem utilizada por todo o grupo de pesquisa dentro do qual este trabalho foi desenvolvido.

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

O Sistema de Alertas troca mensagens com os agentes seguindo um protocolo baseado na linguagem KQML. A implementação da mensagem KQML segue o trabalho de [2] e é composta por dois módulos: um formato geral que possui especificação KQML (classe KQMLMessage) e um formato específico que compõe a tarefa que o agente deve executar (classe ContentMessage).

KQML possui um conjunto de parâmetros chaves reservados que servem para estabelecer uma uniformidade dos parâmetros e suporte para que os programas entendam performativas desconhecidas mas com parâmetros conhecidos [9]. Os parâmetros foram implementados na forma de uma classe Java chamada KQMLMessage, como mostra a tabela 2.

Classe KQMLMessage	
Performative	nome da performativa
Sender	Transmissor
Receiver	Receptor
language	linguagem do conteúdo
ontology	ontologia do conteúdo
replay_to	re-enviar a mensagem para
replay_with	rótulo da resposta
force	se performativa negada
content	conteúdo – Objeto ContentMessage

Tabela 2 - Classe KQMLMessage

De acordo com o formato de mensagem KQML, é necessário especificar o parâmetro *content*, para isso, usa-se a classe *ContentMessage*. A classe *ContentMessage*, especificada por [2], foi alterada para servir o Sistema de Alertas. A tabela 3 mostra a classe *ContentMessage* utilizada no Sistema de Alertas com todos os parâmetros necessários para compor as mensagens. Tais parâmetros são utilizados conforme o tipo de mensagem usada.

Classe ContentMessage	
Type : tipo da mensagem	Time : tempo de permanência na página
Operation : operação realizada pelo usuário (exemplo: 1= logon)	Course: código do curso que está o aluno
User: codpessoa - código do aluno	Numacertos : número de acertos nos exercícios
url: codurl – endereço da página acessada pelo aluno	Session: número da sessão de estudo
Problem : tipo de problema detectado (exemplo: 1= erro exercício)	

Tabela 3 - Classe ContentMessage

O material instrucional é formado por lições, exemplos e exercícios e foi totalmente desenvolvido pelos pesquisadores do Tapejara (psicólogos, pedagogos, técnicos de desenvolvimento de páginas Web e por um especialista no domínio do assunto).

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

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

A figura 7 apresenta a interface do Editor de Alertas. Esta interface é um arquivo HTML denominado Alertas.html, que está disponível em um servidor do Instituto de Informática da UFRGS.

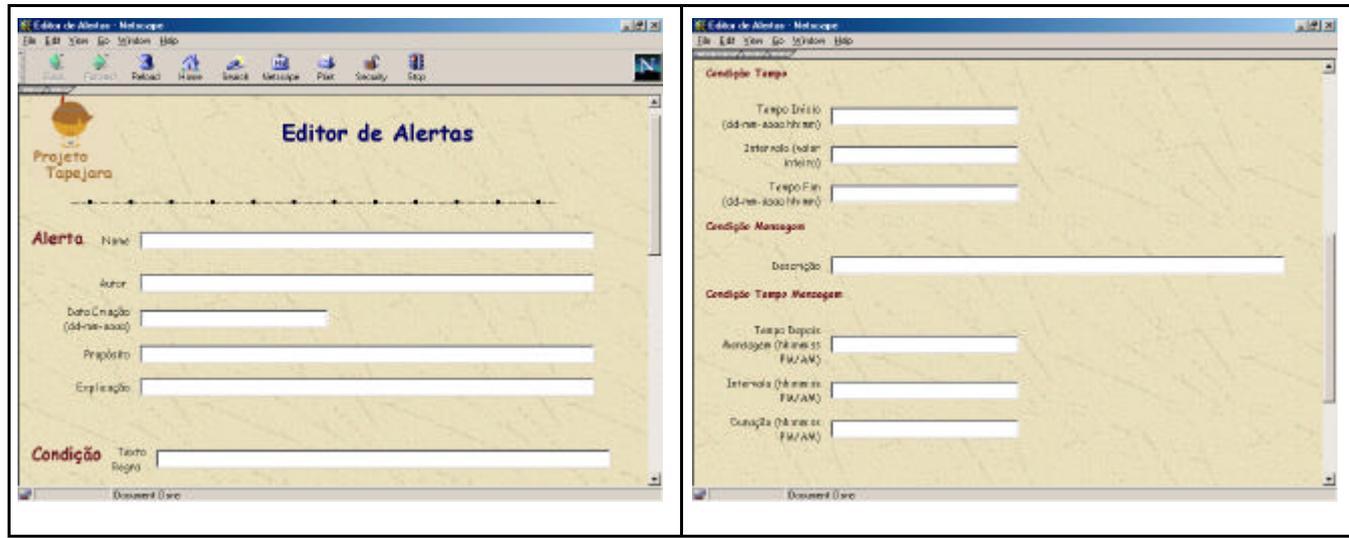


Figura 7 - Interface do Editor de Alertas

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

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

6 Estudo de Caso

Para a validação do mecanismo de alertas proposto neste trabalho, foi realizado um estudo de caso em um módulo de um curso de Telecomunicações, chamado TDMA (*Time Division Multiple Access*). Este módulo foi selecionado, pois é considerado um módulo intermediário (nem inicial e nem final) e possui complexidade suficiente para permitir que ocorra uma aprendizagem, sendo, portanto um importante meio de estudo para o sistema de alertas.

Este curso foi desenvolvido por uma equipe de integrantes do Projeto Tapejara, formada por psicólogos, pedagogos, técnicos de desenvolvimento de páginas Web e por um especialista no domínio do assunto [21]. Ele foi oferecido a funcionários da empresa de telecomunicações CRT Brasil Telecom.

Neste ambiente, todas as ações do aluno são monitoradas e gravadas numa base de dados (*log* Tapejara). O Sistema de alertas utiliza-se destas informações e ao detectar algum problema nas atividades do aluno no curso emite o alerta correspondente.

Alguns alertas gerados durante a execução do curso TDMA são apresentados a seguir.

1. **Alertas enviados ao browser do aluno :** A figura 8 apresenta um exemplo de alerta chamado “Parada Brusca”. Este alerta é enviado quando o sistema detecta uma alteração no ritmo de estudo do aluno.



Figura 8 - Exemplo de alerta “Parada brusca”

2. **Alertas enviados via e-mail :** A figura 9 apresenta dois exemplos de alertas enviados por *e-mail*. O primeiro alerta é enviado para o professor responsável pelo curso e contém o Relatório Semanal sobre os acessos dos alunos. O segundo alerta é para um dos alunos do curso quando o sistema detecta a ausência do aluno nas atividades.

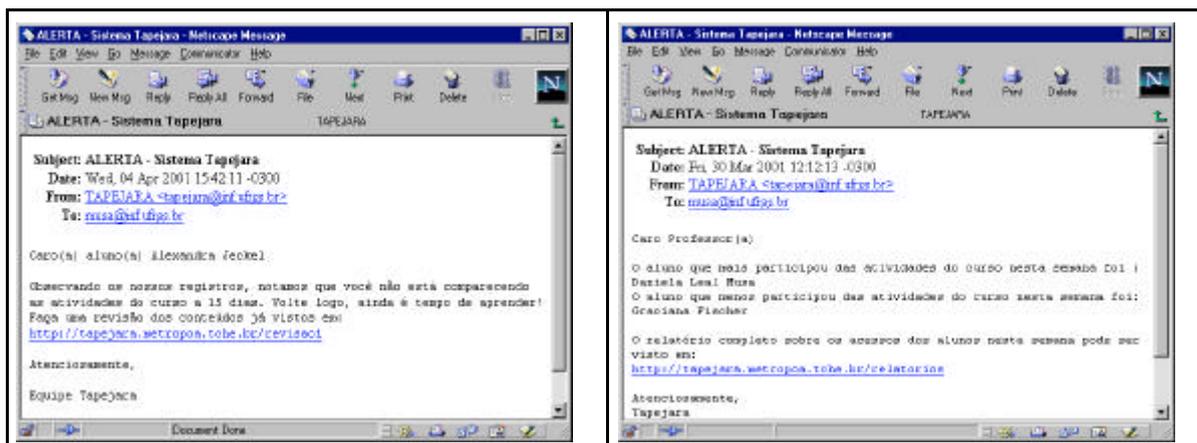


Figura 9 - Mensagem de alerta enviada via e-mail

3. **Alertas enviados ao agente:** A figura 10 ilustra um exemplo de mensagem KQML enviada para o Agente Tapejara. Este alerta é enviado ao agente, quando o sistema percebe um grande número de erros nos exercícios feitos pelo aluno. O agente encarrega-se de tomar uma decisão para esta situação.

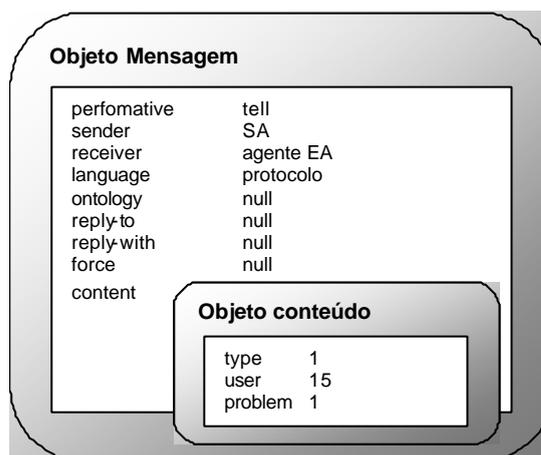


Figura 10 - Exemplo de mensagem enviada ao agente

Após a execução do curso, os usuários (alunos e professores) sentiram-se satisfeitos com as mensagens recebidas (alertas), principalmente por sentirem-se acompanhados no processo de ensino. Uma idéia sugerida pelos professores foi o envio de um conjunto de alertas, ou seja, os alertas seriam guardados em uma base temporária e após um certo critério, por exemplo, temporal, seriam enviados.

7 Conclusão

Este trabalho apresentou o sistema de alertas desenvolvido para detectar problemas nas atividades dos alunos em ambientes educacionais na Internet. O sistema monitora a interação do aluno no ambiente e na ocorrência de algum evento considerado crítico, realiza ações corretivas adequadas. Esse sistema de alertas pode ser integrado a ambientes educacionais que utilizem um mecanismo de

registro das atividades dos alunos nos cursos (*log*), pois este registro são os dados de entrada do sistema de alertas.

O Sistema de alertas apresentado neste trabalho foi proposto com o objetivo inicial de ser integrado ao ambiente Tapejara [22]. No entanto, esse sistema mostra-se adequado tanto para este ambiente como para a sua integração em outros ambientes educacionais, desde que estes possuam um mecanismo que armazene em um banco de dados ou arquivo as interações dos alunos nos cursos (*log*).

Os resultados alcançados até o momento mostram que o sistema pode auxiliar o professor a realizar um acompanhamento mais próximo das atividades dos alunos, pois além de informar sobre o comportamento destes, também identifica aqueles que necessitam de ajuda, os que estão mais adiantados, assim como uma série de outras informações de muita utilidade para a evolução do processo de aprendizagem. Esse sistema também ajuda o aluno, estimulando-o nas suas atividades e auxiliando-o na detecção de problemas. O coordenador do material instrucional também é beneficiado, pois recebe avisos sobre as páginas que não estão sendo acessadas, e outras informações.

Os conceitos aqui apresentados continuam sendo avaliados em situações reais de execução de cursos à distância [13,15]. Os novos dados coletados estão sendo analisados e posteriormente será feita uma avaliação mais completa sobre a real utilidade do Sistema de Alertas. Após este período, outros cursos na Web poderão ser ministrados, a partir deste conhecimento. Espera-se com isto contribuir para que a Educação a Distância possa dar mais um passo para ser uma alternativa viável no processo educacional.

7.1 Trabalhos Futuros

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

- Capturar os eventos gerados por *stored procedures* e *triggers*. São procedimentos que são executados quando alguma informação é modificada na base de dados. Com isso, o sistema poderá reagir às modificações que estão ocorrendo durante a interação do aluno no ambiente, tendo um controle feito em nível de banco de dados.
- Oferecer suporte a formatos de dados comuns e protocolos de mensagem, como XML.
- Disponibilizar diversos dispositivos de comunicação para escolha do usuário, sendo possível enviar os alertas via *pager*, telefone celular, página *Web* ou *fax*.

8 Referências

- [1] L.A.O, Azevedo, *Análise de Atividades de Alunos de Sistemas de Ensino via Internet*, Dissertação de Mestrado, PPGC, UFRGS, Porto Alegre, 2000.
- [2] F. Bica, *Eletrotutor III – Uma Abordagem Multiagente para o Ensino a Distância*, Dissertação de Mestrado, PPGC, UFRGS, Porto Alegre, RS, 2000.
- [3] S. Chakravarthy, “Snoop: an Expressive Event Specification Language for Active Databases”, *Knowledge and Data Engineering Journal*, vol.14, 1994, pp. 1-26.
- [4] J. Chang, “Database Alerting Techniques for Office Activities Management”, *IEEE Transactions on Communications*, New York, vol. 30, no. 1, 1982, p.74-81.

- [5] S. Chang, L. Leung, "A Knowledge-Based Message Management System", *ACM Transactions on Office Information Systems*, New York, vol.5,1987, p. 213-236.
- [6] M. T. Costa, *Uma Arquitetura Baseada em Agentes para Suporte ao Ensino a Distância*. Tese de Doutorado, Programa de Pós-Graduação em Engenharia de Produção, UFSC, Florianópolis, 1999.
- [7] The Darpa Knowledge Sharing Initiative External Interfaces Working Group, Specification of the KQML Agent-Communication Language. Relatório Técnico, University of Maryland, Baltimore, 1993.
- [8] D. Faensen, A. Hinze, H. Schweppe, "Alerting in a Digital Library Environment. Do Channels Meet the Requirements?" *Research and Advanced Tecnology for Digital Libraries*, (ECDL), Heraklion, Grecia. Lecture Notes in Computer, Springer-Verlag, p. 643-644, 1998.
- [9] R. Fritzson, "KQML as an Agent Communication Language", Computer Science Department, University of Maryland, Baltimore Country, 1996.
- [10] A. Haddadi, "Communication and Cooperation in Agent Systems", Berlin: Springer-Verlag, Lecture Notes in Artificial Intelligence, v.1056, 1996
- [11] B. F. Leão et al, "Decision Support Systems for Heathcare: a Methodology Review". *Annual Conference on Electronic Patient Records*, San Antonio, Texas, 1998.
- [12] D. B. Lenat, *Knowledge Representation*. Addison-Wesley, USA, 1988.
- [13] D. L. Musa et al, "Agente para Auxílio a Avaliação de Aprendizagem em Ambientes de Ensino na Web". *XII Simpósio Brasileiro de Informática na Educação*, (SBIE), Vitória, ES, 2001.
- [14] D.L. Musa, *Um Sistema de Alertas Inteligentes para Ambientes de Ensino na Internet*, Dissertação de Mestrado, PPGC, UFRGS, Porto Alegre, 2001.
- [15] D. L. Musa, J.P.M. Oliveira, "Um Sistema de Alertas Inteligentes para Monitoração do Aluno em Ambientes de Ensino na Web", *Taller Internacional de Software Educativo*, (TISE), Santiago, Chile. 2001.
- [16] A. Peled, "Logging for Sucess - Advancing the Use to Improve Computer Mediated Distance Learning", *Journal of Educational Computing Research*. vol. 21, no. 3, 1999.
- [17] A. P. Rodrigues, *O Processo de Avaliação de Ensino Aprendizagem em Ensino a Distância*. Trabalho Individual, PPGC, UFRGS, Porto Alegre, 2002.
- [18] H. Schweppe, A. Hinze, D. Faensen. "Event-based Notification on the Web". *International World Wide Web Conference*, n. 9, Amsterdam, 2000.
- [19] R. Silveira, *Modelagem Orientada a Agentes Aplicada a Ambientes Inteligentes Distribuídos de Ensino*, Tese de Doutorado, PPGC, UFRGS, Porto Alegre, 2000.
- [20] M. Souto et al, "Modelo de Ensino Adaptativo na Internet Baseado em Estilos Cognitivos de Aprendizagem", *Simpósio Brasileiro de Informática na Educação*, Maceió, 2000.
- M. Souto et al, "Metodologia de Construção do material instrucional em um Ambiente de Ensino Inteligente na Web", *XII Simpósio Brasileiro de Informática na Educação*, Vitória, ES, Nov. 2001.
- [21] F. Tao, F. Murtagh, "Towards Knowledge Discovery from WWW Log Data.", *International Conference on Information Technology: Coding and Computing*. (ITCC 00), IEEE, Las Vegas, Nevada, 2000.
- [22] Tapejara, "Artigo Final", <http://www.inf.ufrgs.br/~tapejara/docs/ArtigoFinal.pdf> (Junho 2002).
- [23] Tapejara, *Determinação de índices e Indicadores das trajetórias de aprendizagem de sujeitos submetidos ao ensino tutorializado assíncrono pela Internet*. Relatório Técnico. PPGC, UFRGS, Porto Alegre, RS.
- [24] R.M.Vicari, "Artificial Intelligence and Educational Systems", *Journal of School*, vol.1, no. 2 of Olivais 1998.