

PLANEJAMENTO INSTRUCIONAL AUTOMATIZADO
PARA UM SISTEMA DE ENSINO A DISTÂNCIA

por
Flávio Ferreira Borges

DISSERTAÇÃO APRESENTADA À
UNIVERSIDADE FEDERAL DE UBERLÂNDIA,
UBERLÂNDIA, MINAS GERAIS,
COMO PARTE DOS REQUISITOS EXIGIDOS
PARA OBTENÇÃO DO TÍTULO DE
MESTRE EM CIÊNCIA DA COMPUTAÇÃO.
MARÇO DE 2005

© Todos os direitos reservados à Flávio Ferreira Borges, 2005

Livros Grátis

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
FACULDADE DE COMPUTAÇÃO

Os abaixo assinados, por meio deste, certificam que leram e recomendam para a Faculdade de Computação a aceitação da dissertação intitulada “**Planejamento Instrucional Automatizado para um Sistema de Ensino a Distância**” por **Flávio Ferreira Borges** como parte dos requisitos exigidos para a obtenção do título de **Mestre em Ciência da Computação**.

02 de março de 2005

Prof. Dr. Carlos Roberto Lopes (UFU)

Prof^a. Dr^a. Márcia Aparecida Fernandes (UFU)

Prof. Dr. João Bosco da Mota Alves (UFSC)

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

02 de março de 2005

Autor: **Flávio Ferreira Borges**
Título: **Planejamento Instrucional Automatizado para um Sistema de Ensino a Distância**
Faculdade: **Computação**
Grau: **Mestrado**

Fica garantido à Universidade Federal de Uberlândia o direito de circulação e impressão de cópias deste material para fins não comercial, bem como o direito de distribuição por solicitação de qualquer pessoa ou instituição.

Flávio Ferreira Borges
Autor

O AUTOR RESERVA PARA SI QUALQUER OUTRO DIREITO DE PUBLICAÇÃO DESTA MATERIAL

A minha esposa Renata pelas esperanças depositadas.

Agradecimentos

Ao meu orientador de dissertação, Prof. Dr. Carlos Roberto Lopes, pelas oportunidades oferecidas e principalmente pela paciência e compreensão demonstradas ao longo deste trabalho.

A todos os funcionários e professores do programa de mestrado da UFU.

À Universidade de Rio Verde – Fesurv pelo suporte financeiro.

Aos meus colegas de mestrado pelo companheirismo e momentos agradáveis vividos durante nossos trabalhos.

A toda minha querida família, principalmente aos meus filhos e filha pelas inúmeras ausências.

Aos meus colegas que não conseguiram, por algum motivo, chegar ao fim dessa jornada.

E a Deus, pelo amadurecimento e crescimento proporcionado.

Resumo

Este trabalho descreve um sistema de planejamento instrucional automatizado que está presente em um ambiente para a prática do ensino a distância, denominado como SIMEDUC. Esse ambiente possui uma arquitetura multiagente e foi desenvolvida para utilizar os recursos disponíveis na *Web*, a fim de auxiliar o ensino a distância.

A arquitetura multiagente é composta por quatro tipos de agentes: o agente de avaliação, o agente especialista, o agente pedagógico e o agente assistente, sendo o último objeto de estudo deste trabalho.

O agente assistente tem como papel principal observar o comportamento dos estudantes durante a visualização de um plano de curso, e a partir de orientações internas ao agente, promover ações que possam auxiliar o estudante em seu processo de ensino/aprendizagem.

A comunicação e troca de informações entre os agentes garantem que o sistema SIMEDUC possa ser considerado uma ferramenta que utiliza a inteligência artificial para promover aos participantes um curso baseado na *Web* personalizado.

Abstract

This work describes an automated instructional planning system which is present in a distance education practice environment, known as SIMEDUC. This environment implements a multiagent architecture and was designed to make use of the resources available in the Web, aiming to support the distance education.

The multiagent architecture is composed of four types of agents: the evaluation agent, the specialist agent, the pedagogic agent and the assistant agent, where the last is the object of study in this work.

The assistant agent has the main role of observing the behavior of the students while browsing a course plan and, after the internal orientations of the agent, promote actions that can help the student in his teaching/learning process.

The communication and information exchange between the agents guarantee the system SIMEDUC to be considered a tool that uses artificial intelligence to give the participants a customized web based course.

Sumário

Agradecimentos	v
Resumo	vi
Abstract	vii
Sumário	viii
Lista de Tabelas	xi
Lista de Figuras	xii
Lista de Abreviaturas	xiv
1 Introdução	1
1.1 Objetivo.....	3
1.2 Estrutura da Dissertação.....	4
2 Técnicas de inteligência artificial no processo ensino/aprendizagem	6
2.1 Planejamento Apoiado em Inteligência Artificial.....	8
2.1.1 Planejamento Clássico	9
2.1.2 Planejamento Dinâmico	12
2.1.2.1 Sistema Procedimental de Raciocínio.....	13

2.1.2.2 RAP	16
2.1.2.3 Planos Universais	19
2.1.2.4 Regras de Controle Situadas	21
2.1.2.5 Arquiteturas de Três Camadas	23
2.2 Planejamento Instrucional.....	24
2.2.1 Sistemas Tutores Inteligentes (STI)	25
2.2.2 Técnicas de Planejamento Instrucional em STI.....	29
2.2.2.1 Planejador APE.....	30
2.2.2.2 Planejador BEETS	30
2.2.2.3 Planejador Teórico de Decisão	31
2.2.2.4 Planejador PLAIT	32
2.2.2.5 Método de aproximação de Smith	33
2.3 Planejamento Instrucional em Sistemas de EAD.....	33
2.3.1 Construção de Sistema de EAD em STI	34
2.3.2 Sistema ABITS	38
2.3.3 Sistema Tobie.....	41
2.4 Considerações Finais.....	43
3 Proposta para um Sistema de Ensino a Distância	45
3.1 Arquitetura do Sistema SIMEDUC.....	46
3.1.1 Sistema Gerenciador de Cursos	46
3.1.1.1 Ambiente Administrativo.....	47

	x
3.1.1.2 Ambiente de Curso.....	50
3.1.2 Sistema Inteligente Multiagente.....	52
3.2 Comunicação entre os Agentes	56
4 Integrando Planejamento e Reatividade	60
4.1 Planejamento	61
4.1.1 Descrição das Regras	63
4.1.2 Descrição das Regras do Planejador	67
4.1.2.1 Regras de Gerenciamento	68
4.1.2.2 Regras de Seleção de Operadores - RSO	69
4.1.2.3 Regras de Decomposição de Operadores – RDO	70
4.1.3 Algoritmo do Planejador	72
4.1.4 O Plano Gerado	74
4.2 Agente Assistente Reativo	76
4.2.1 Reatividade no Plano	78
4.3 Detalhes de Implementação	84
4.3.1 Reorganizando o Plano para a Reatividade.....	85
4.3.2 Inserindo Parâmetros para a Reatividade.....	86
4.3.3 Apresentação Reativa do Plano.....	90
5 Conclusão e Perspectivas Futuras	97
Referências Bibliográficas	100

Lista de Tabelas

Tabela 4.1 - Descrição dos operadores abstratos presentes no planejamento.....	64
Tabela 4.2 - Descrição dos operadores primitivos presentes no planejador	65
Tabela 4.3 - Conjunto de regras de gerenciamento do plano.....	68
Tabela 4.4 - Conjunto de Regras para Seleção de Operadores Abstratos.....	69
Tabela 4.5 - Conjunto de regras para decomposição de operadores Abstratos.....	72
Tabela 4.6 - Simplificação do plano de curso gerado	79
Tabela 4.7 - Parâmetros para o Operador Avalia.....	83
Tabela 4.8 - Reações previstas pelo Agente Assistente	83
Tabela 4.9 - Parâmetros de referência para a reatividade	92

Lista de Figuras

Figura 2.1 - Planejamento em IA	8
Figura 2.2 - Estrutura do Sistema PRS	14
Figura 2.3 - O Ambiente de Execução de RAPs.....	17
Figura 2.4 - Mecanismo de Execução de um RAP	18
Figura 2.5 - Sistema Projetor/Reator.....	21
Figura 2.6 - Arquitetura de um STI.....	26
Figura 2.7 - Arquitetura do Sistema ABITS	40
Figura 2.8 - Arquitetura do Sistema Tobie.....	41
Figura 2.9 - Planejamento Reativo em Tobie	42
Figura 3.1 - Tela principal do Ambiente Administrativo do SGC.....	49
Figura 3.2 - Arquitetura do SGC do SIMEDUC.....	49
Figura 3.3 - Tela Principal do Ambiente de Curso do SGC	51
Figura 3.4 - Sistema Multiagente do SIMEDUC	54
Figura 3.5 - Modelo de comunicação Blackboard do SIMEDUC	57
Figura 4.1 - Arquitetura do agente Pedagógico.	62
Figura 4.2 - Representação da decomposição do Aprender(x).	66
Figura 4.3 - Decomposição final do operador Aprender(x).....	67
Figura 4.4 - Algoritmo utilizado pelo planejador de conteúdo	73
Figura 4.5 - Seqüência de um plano de curso gerado pelo planejador.....	75

Figura 4.6 - Tela com a seqüência de conteúdos do curso a ser estudado	76
Figura 4.7 - Arquitetura do Agente Assistente Reativo	78
Figura 4.8 - Estratégia para o Agente Assistente	82
Figura 4.9 - Tela do menu dos cursos disponíveis para o estudante	86
Figura 4.10 - Parte do código fonte que possibilita a remoção das repetições	86
Figura 4.11 - Tela inicial da apresentação dos cursos para o professor/autor.....	87
Figura 4.12 - Definição inicial dos parâmetros para promover a reatividade.....	89
Figura 4.13 - Tela que define quais operadores deverão ser utilizados na reação.90	
Figura 4.14 - Apresentação parcial do curso Elementos Básicos	91
Figura 4.15 - Retorno da avaliação de baixo desempenho.....	94
Figura 4.16 - Reação no plano aplicando a revisão de objetos	94
Figura 4.17 - Retorno da avaliação satisfatória.....	95
Figura 4.18 - Visualização da continuação normal do plano.	95
Figura 4.19 - Plano apresentado após a avaliação acima do esperado.....	96

Lista de Abreviaturas

ACL – Agent Communication Language.

APE – ATLAS Planning Engine

CAI – Computer-aided Instruction.

DF – Directory Facilitator.

DT – Decision Theoretic

EAD – Educação a Distância.

EADW – Educação a Distância Apoiada na Web.

FIFO – First In First Out.

FIPA – Foundation for Intelligent Physical Agents.

ICAI – Intelligent Computer-aided Instruction.

IEEE – Institute of Electrical and Electronics Engineers.

JADE – Java Agent Development Environment.

KA – Knowledge Area (Área de Conhecimento).

LOM – Learning Object Metadata.

LTSC – Learning Technology Standards Committee.

OAA – Open Agent Architecture.

PHP – Personal Home Pages.

PLAIT – Planning agents in Intelligent Tutoring.

POO – Programação Orientada a Objetos.

POP – Planejador de Ordem Parcial.

- PRS** – Procedural Reasoning System.
- PUSH** – Plan and User Sensitive Help.
- RAP** – Reactive Action Package.
- RC** – Regras Controle do planejador.
- RCP** – Regras de Controle da Personalização.
- RDO** – Regras de Decomposição de Operadores.
- RGD** – Regras de Geração da Decomposição.
- RGP** – Regras de Geração de Planos.
- RMI** – Invocações Remotas de Métodos.
- RSD** – Regras de Seleção da Decomposição.
- RSO** – Regras de Seleção de Operadores.
- SCR** – Regras de Controle Situadas.
- SGC** – Sistema Gerenciador de Cursos.
- SIM** – Sistema Inteligente Multiagente.
- SIMEDUC** – Sistema Inteligente Multiagente para a Educação.
- SMA** – Sistemas Multagentes.
- SQL** – Structured Query Language.
- STI** – Sistema Tutores Inteligentes.
- STRIPS** – Stanford Research Institute Problem Solver.
- VHS** – Video Homeless System.
- WEB** – Rede Mundial ou Teia (World Wide Web).

Capítulo 1

Introdução

Ultimamente a Educação a Distância (EAD) no Brasil vem encontrando grande apoio por parte do governo e da sociedade. Para que a EAD possa se tornar uma realidade cada vez maior em nosso sistema educacional é indispensável que os participantes do processo, professores, pesquisadores, coordenadores e principalmente os estudantes interajam de forma independente do local onde se encontram ou do horário de disponibilidade de cada um. Várias tecnologias estão sendo utilizadas para auxiliar neste processo, tais como, programas de televisão, aulas gravadas em VHS, material impresso enviado por correios e a mais nova delas a rede mundial de computadores - Internet.

As propostas que utilizam a WEB como meio alternativo para a prática de EAD, oferecem grandes possibilidades de interação com um mundo heterogêneo de aprendizes, no qual eles possuem conhecimentos, interesses, preferências e disponibilidades distintas entre si, portanto, constituindo assim um ambiente bastante dinâmico.

Tais características fazem com que a educação apoiada na Web seja um meio adequado para aplicar técnicas que possibilitem observar e interpretar esses fatores de dinamismo, as quais poderão ser utilizadas como recursos auxiliares no

processo de ensino/aprendizagem. Essas técnicas podem ser encontradas em tutores computacionais conhecidos como Sistemas Tutores Inteligentes (STI) (MURRAY, 1999). Esses sistemas vieram para oferecer diversos recursos didáticos que auxiliam na organização dos conteúdos e proporcionam uma aprendizagem independente e flexível aos estudantes (ARETIO, 1994). Essa independência e flexibilidade, na forma de ensinar, estão diretamente relacionadas a estudos na área de Inteligência Artificial (IA) (COSTA, 2000).

Existem duas características básicas associadas a um STI (BRUSILOVSKY et al, 2003). Uma característica é o sequenciamento dos conteúdos de forma personalizada, ou seja, organização automática dos conteúdos a serem apresentados. A outra é a possibilidade de se oferecer suporte por meio de soluções interativas com os problemas que por ventura venham a surgir. Essas características diferenciam os STIs dos sistemas tradicionais de instrução assistida por computador, já que os STIs incorporam técnicas inteligentes que implementam os procedimentos de um tutor humano em um ambiente de ensino a distância.

Para se obter um ensino personalizado de forma independente e flexível, como proposto em STI, se faz necessário automatizar a forma como os conteúdos instrucionais serão seqüenciados. Além da organização coerente os conteúdos devem ser apresentados levando em consideração as pré-condições estabelecidas pelo professor/autor e as características do estudante envolvido no processo de aprendizagem. Sendo esse último fator considerado como fundamental para o sucesso da EAD conforme Brusilovsky & Vassileva (2003). A técnica de organização automática de conteúdos é conhecida como Planejamento Instrucional (QUEIROZ, 2004).

O Planejamento Instrucional para ambientes dinâmicos está em franco crescimento, por oferecer grandes possibilidades de sucesso no processo de ensino/aprendizagem. Neste trabalho exploramos uma das subáreas de planejamento, denominada planejamento reativo (VASSILEVA, 1995), para automatizar o processo instrucional. Propomos uma reatividade baseada em sistemas multiagentes.

No trabalho desenvolvido apresentamos um Agente Inteligente não humano que, a partir de um plano previamente gerado, promove alterações na execução deste, por meio do monitoramento do ambiente como, por exemplo, a navegação do estudante pelo plano com o intuito de treinar ou ensinar. Essas modificações são conhecidas como ações reativas (VASSILEVA, 1995). Na abordagem proposta, a reatividade é de responsabilidade de um dos Agentes, que é denominado Agente Assistente. O trabalho desenvolvido é dedicado a um sistema de ensino a distância baseado na Web, denominado SIMEDUC.

1.1 Objetivo

O objetivo deste trabalho é apresentar um Agente Assistente que possa promover reações a partir de observações durante a navegação do estudante em um curso planejado. Essas reações deverão oferecer ao estudante recursos que o auxiliem na compreensão dos conteúdos presentes no plano. A construção do Agente Assistente consiste em gerar um conjunto de regras condição/ação, a partir do plano de curso gerado por um Agente Pedagógico e aplicá-las em momentos oportunos. Receber novas regras do professor/autor para expandir as possibilidades de reações também é uma característica do Agente Assistente.

As ações a serem executadas pelo Agente Assistente consistem em apresentar o plano instrucional de forma dinâmica, no qual as características de navegação do estudante irão determinar quais ajustes de apresentação deverão ser feitos durante o curso. Um exemplo desse ajuste seria a reação que promove a revisão de conteúdo, acionada pela verificação da dificuldade do estudante em responder corretamente questões sobre um determinado conteúdo. Contudo, as ações do agente não poderão modificar a seqüência em que os conteúdos foram estabelecidos na geração do plano. As ativações das reações ocorrerão a partir de uma comunicação entre os Agentes presentes na arquitetura proposta.

1.2 Estrutura da Dissertação

Para a apresentação dos resultados do trabalho proposto esta dissertação foi organizada em cinco capítulos, sendo o primeiro essa introdução.

O segundo capítulo trata do estado da arte em planejamento instrucional. Os assuntos tratados por este capítulo incluem várias abordagens sobre o planejamento apoiado em Inteligência Artificial, buscando destacar as limitações e os pontos positivos. O planejamento em ambientes dinâmicos é mais detalhado por fazer parte da proposta final desta dissertação. Isto foi feito objetivando em seguida descrever técnica de planejamento instrucional utilizada nos STI que tem como base técnicas o planejamento dinâmico. Finalmente sistemas que utilizam a técnica de planejamento instrucional voltados para a Educação a Distância.

O terceiro capítulo inicia-se descrevendo o sistema multiagente SIMEDUC. Na primeira parte são descritos os módulos que compõem o sistema, destacando os componentes principais do Sistema Gerenciador de Cursos (SGC), presente na

arquitetura do SIMEDUC. Na segunda parte deste capítulo são apresentados os conceitos sobre sistema multiagente demonstrando como é realizada a interação entre o SGC e o módulo inteligente do SIMEDUC. Dessa arquitetura o Agente Assistente é o objeto de estudo.

O quarto capítulo aborda a integração entre Plano Instrucional e Regras de Controle para a reatividade. Ênfase maior será dada as regras que compõem a base da geração do plano instrucional, que será utilizado para apresentar uma seqüência lógica de conteúdos aos estudantes. No subcapítulo 4.2 encontra-se descrito a proposta central deste trabalho que é o Agente Assistente Reativo que deverá promover reações com a finalidade de auxiliar os estudantes durante o curso. No final deste capítulo trazemos alguns resultados da implementação do Agente Assistente em um plano previamente gerado pelo Agente Pedagógico.

O último capítulo da dissertação reporta algumas considerações finais e perspectivas futuras sobre o trabalho.

Capítulo 2

Técnicas de inteligência artificial no processo ensino/aprendizagem

Entre várias técnicas de Inteligência Artificial (IA) utilizadas para desenvolver sistemas, destinados ao ensino, encontramos o Planejamento Instrucional que tem finalidade gerar uma seqüência de conteúdo. Essa seqüência instrucional é composta por ações instrucionais que, uma vez executadas, satisfazem o objetivo educacional. No desenvolvimento de STI, uma subárea da IA que usa o processo educacional como domínio, faz-se necessária a existência de um mecanismo de planejamento instrucional automatizado. Em IA, existe uma outra área de pesquisa ativa que recebe também a designação de planejamento. Planejamento apoiado em IA consiste na geração automática de uma seqüência de ações (um plano) que, uma vez executada, conduzirá à satisfação de uma determinada meta. Nos primórdios da IA, o plano era formado basicamente por ações físicas a serem executadas por um robô. Portanto, percebe-se que existe uma relação entre planejamento apoiado em IA e planejamento instrucional. Em vez de ações físicas, um plano instrucional consiste de ações instrucionais.

Planejamento apoiado em IA é considerado uma das áreas mais ativas de pesquisa e é uma das mais “antigas” dentro da área de IA. Vários resultados inte-

ressantes têm sido obtidos ao longo dos anos em ambientes dinâmicos. Uma das abordagens para planejamento em ambientes dinâmicos é conhecida como *planejamento reativo*. A área de STI pode ser usada como um domínio para o desenvolvimento e testes de idéias, arquiteturas e aplicações para planejamento reativo (VASSILEVA, 1995). Isto é possível porque STI apresenta todas as características que tornam necessário o planejamento reativo.

- a) Em STI não se tem a certeza dos efeitos das ações instrucionais executadas.
- b) A hipótese de que o mundo permaneça estático enquanto as ações são executadas não pode ser considerada.
- c) Não pode assumir que se tenha conhecimento completo do mundo.

Desta forma, a inserção de componentes reativos num planejador instrucional pode trazer resultados interessantes e experiência para as áreas de planejamento apoiado em IA e STI (CHO, 2000). Assim, neste trabalho, investigaremos meios de aplicar técnicas de planejamento no domínio educacional. Apesar do estreito relacionamento, poucos trabalhos têm sido desenvolvidos no sentido da integração de técnicas de planejamento dinâmico no domínio da instrução.

Este capítulo tem o objetivo de situar o leitor em relação ao trabalho desenvolvido. Inicialmente, serão descritos os conceitos principais de planejamento apoiado em IA com ênfase em ambientes dinâmicos. Em seguida, será abordado o planejamento instrucional descrevendo alguns sistemas existentes. Finalmente, será mostrado como sistemas de educação a distância podem ser desenvolvidos com componentes de geração instrucional apoiadas em IA.

2.1 Planejamento Apoiado em Inteligência Artificial

Podemos considerar que Planejamento Apoiado em IA é um conjunto de ações que deverão ser utilizadas por um dispositivo atuador, com a finalidade de modificar o estado inicial do *mundo*. Consideramos *mundo* como sendo a representação dos objetos que compõem a definição de um problema a ser resolvido. As ações a serem executadas pelo atuador, com base no planejamento, transformam o *estado inicial*, buscando alcançar um estado final desejado ou, simplesmente, satisfazer uma meta (SMITH & FRANK & J'ONSSON, 2000).

Além da descrição do mundo, das ações a serem executadas e da meta, o planejador deverá levar em consideração se o ambiente de atuação é estático ou dinâmico. Entende-se por estáticos os ambientes onde somente o atuador é capaz de provocar mudanças. Tais ambientes são tratados por planejamentos clássicos. Os demais ambientes são tratados por planejamento dinâmico.

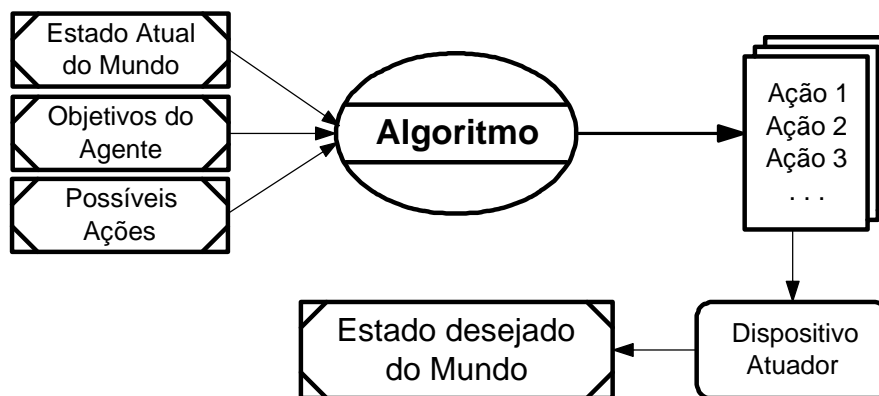


Figura 2.1 - Planejamento em IA

O modelo representado na Figura 2.1 sugere um algoritmo de Planejamento em IA que recebe três conjuntos. Fazendo uma leitura do algoritmo, temos, como informações de entrada, o **estado atual do mundo**, que são informações sobre o ambiente atual ou definição do problema a ser resolvido, aos **objetivos do agente**

que trata especificamente das técnicas de IA, na qual os agentes presentes possuem estratégias para resolver os problemas apresentados pelo estado atual do mundo, com o intuito de alcançar uma meta, e o último conjunto traz as **possíveis ações** pré-estabelecidas que devem ser aplicadas para resolver problemas ou para se alcançar a meta. Como saída do algoritmo, são apresentadas **ações** que foram criadas a partir dos dados de entrada. Essas ações serão utilizadas pelo **dispositivo atuador** que procurará interferir no ambiente atual buscando modificar o estado encontrado, alcançando, assim, um **estado desejado do mundo**, no qual a aplicação das ações propostas pelo algoritmo possa gerar modificações que venham ao encontro da satisfação da meta final estabelecida no plano.

Nos próximos subcapítulos, apresentaremos dois modelos de planejamento existentes: um dedicado ao planejamento clássico e outro a ambientes dinâmicos. Ênfase será dada ao dinâmico por ser o modelo utilizado como base para os trabalhos propostos.

2.1.1 Planejamento Clássico

Esse tipo de planejamento tem como característica trabalhar com informações corretas e completas (RUSSEL & NORVING, 1995) (LOPES, 1998). As seguintes hipóteses caracterizam o planejamento clássico:

- O mundo de atuação é composto somente de atuadores identificados na fase de especificação.
- O estado do mundo será modificado por meio de ações primitivas executadas pelo atuador, conforme suas especificações.
- Somente uma ação pode ser realizada de cada vez.

- No início do processo de planejamento, é necessário informar todos os fatos relevantes do domínio.
- Não deverá existir restrição de recursos e de tempo.

O planejador STRIPS (Stanford Research Institute Problem Solver), proposto por Fikes e Nilsson (SILVA, 2000), foi uma das primeiras propostas para resolver o problema de planejamento e utiliza uma linguagem baseada em cálculo de predicados que oferece recursos para representar ações e especificar os estados do mundo. As ações do planejador são representadas por operadores de transformação de estados, compostos por três listas (LOPES, 1998):

1. Lista de pré-condições que devem ser verdadeiras para que a ação possa ser aplicável.
2. Lista de fatos a serem subtraídos do estado do mundo atual após a aplicação da ação.
3. Lista de fatos a serem adicionados ao estado do mundo atual após a aplicação da ação.

A representação dos estados do mundo em STRIPS é feita por conjunções de literais instanciados (SILVA, 2000). Os literais não representados são considerados como falsos. Essa situação é conhecida como hipótese do mundo fechado¹.

A técnica utilizada pelo STRIPS consiste em utilizar uma pilha de dados, composta por metas e operadores. Os procedimentos são executados conforme são inseridos na pilha e no início da pilha somente pode existir a meta. Ao aplicar um conjunto de ações que satisfaça a meta ou parte dela, esse conjunto será aplicado na descrição do estado, o qual resultará em uma nova descrição do mundo e a pilha de metas é atualizada. Esse processo se repetirá até que a aplicação das ações

¹ Tradução para a frase Closed World Assumption.

venha esvaziar a pilha de metas. Após o esvaziamento da pilha, verifica-se o estado final coincide com a meta proposta, caso seja verdadeiro, a solução do problema foi encontrada, porém, se o resultado encontrado não foi o esperado, a pilha de metas recebe os objetos que não aparecem no estado final e o processo é reiniciado.

A metodologia para resolver problemas utilizadas em STRIPS é a mais difundida atualmente em sistemas de planejamento e a grande contribuição dessa proposta é o formalismo proposto, pois é possível representar as ações e estados por meio de um algoritmo, sem se preocupar com a linguagem em que ele será aplicado (SILVA, 2000).

Outro planejador que apresenta inovações em relação ao STRIPS é o planejador POP (Planejador de Ordem Parcial) que representa o plano como uma seqüência parcial de ações, ou seja, em uma seqüência não-linear de ações. O STRIPS é um planejador de ordem total que representa os planos em seqüência linear. Mesmo apresentando resultados superiores ao STRIPS, o POP demonstrou uma complexidade exponencial, o que não é desejável em sistemas computacionais (LOPES, 1998). Mesmo sendo utilizado por um grande número de sistemas de planejamento, o STRIPS apresenta algumas limitações na sua utilização (RUSSEL, 1995) (LOPES, 1998):

1ª - Incapacidade de caracterizar diferentes tipos de abstração para o novo plano a ser gerado.

2ª - Impossibilidade de descrever efeitos dependentes do contexto.

3ª - Aquisição de informações em tempo de execução é um comportamento que deverá ser planejado.

4ª - Planejamento computacional caro e, muitas vezes, os problemas de raciocínio sobre as ações são totalmente irresolutos.

5ª - Restrições de tempo impostas pelo mundo são algumas vezes severas, aumentando o grau das dificuldades enumeradas acima.

Atualmente, várias abordagens para planejamento continuam sendo propostas. É interessante observar que algumas destas propostas incorporam de alguma forma o uso de planejamento clássico. Assim sendo, muitas das pesquisas desenvolvidas na área buscam desenvolver planejadores mais eficientes adotando as hipóteses do planejamento clássico. Exemplos de avanços recentes incluem GRAPHPLAN (WELD, 1999), BLACKBOX (KAUTZ, 2003), FF (HOFFMANN & NEBEL, 2001) e HSP (BONET & GEFFNER, 1999).

2.1.2 Planejamento Dinâmico

O planejamento clássico pressupõe que os estados resultantes nos ambientes após a execução das ações sejam conhecidos. Entretanto, há ambientes que apresentam certo dinamismo dificultando assim a aplicação correta das ações e gerando resultados indesejados. Nesse tipo de ambiente, depara-se com incertezas relacionadas à descrição de estados iniciais e finais, como também os efeitos das ações.

A possibilidade de identificar e corrigir planos em domínios nos quais existem incertezas é uma característica de sistemas de planejamento para ambientes dinâmicos. Os próximos subcapítulos descreverão alguns sistemas de planejamen-

to propostos para domínios dinâmicos e com incertezas. Neles, o agente executor das ações deve ser capaz de reagir adequadamente a uma nova informação obtida.

2.1.2.1 Sistema Procedimental de Raciocínio

Inspirado em conceitos de crenças, desejos e intenções, Georgeff et al (1985, 1987; 1987) desenvolveu um sistema para controle de robôs denominado *Procedural Reasoning System* (PRS). Esse sistema considera ações que dependem dos seus desejos atuais, das crenças e dos desejos anteriores, sendo um sistema capaz de raciocinar sobre o seu estado interno e modificá-lo em caso de necessidade.

PRS, cuja estrutura é apresentada na Figura 2.2, provê reatividade, essencial aos sistemas autônomos, presentes em ambientes dinâmicos. Sua base de dados é formada pelas crenças atuais do sistema. Tais crenças podem ser fornecidas pelo usuário ou obtidas no momento de sua execução.

Os comportamentos ou desejos do sistema são representados por metas, as quais aparecem tanto na pilha de metas quanto na representação das áreas de conhecimentos (KA's). Isto implica que metas não somente representam estados de um mundo estático a ser obtido.

As KA's consistem de um corpo que descreve os passos do procedimento e uma condição de invocação que especifica suas condições de utilização. As metas são satisfeitas pela execução de KA's.

Além de KA's o sistema possui também meta-KA's que contém informações sobre a manipulação de crenças, desejos e intenções do próprio PRS. Por exemplo, meta-KA's típicas codificam vários métodos para escolher entre múltiplas KA's relevantes e determinam como obter uma conjunção de metas ou uma

disjunção de metas e computar a quantidade de raciocínio adicional dado às restrições de tempo real do domínio do problema. Para cada aplicação, existirão meta-KA's definidas independentes de domínio.

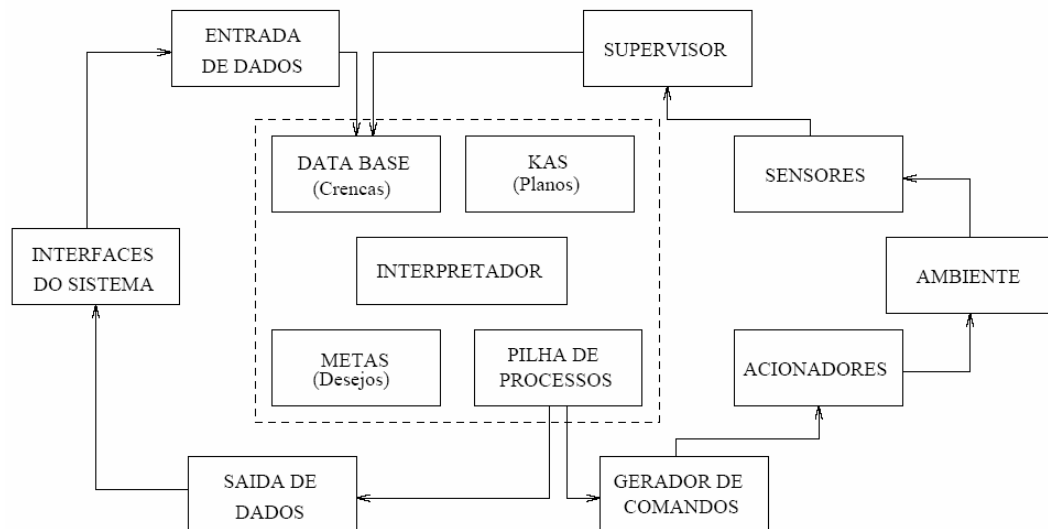


Figura 2.2 - Estrutura do Sistema PRS

O funcionamento do sistema é de responsabilidade de um interpretador, o qual, em qualquer instante, é capaz de perceber os objetivos que estão ativos e as crenças existentes na base de dados. Desta forma, uma KA será escolhida a partir de um subconjunto e inserida na pilha de processos que representa uma intenção de satisfazer determinada meta.

Os ambientes dinâmicos, ou seja, aqueles ambientes que podem sofrer modificações a qualquer momento conduzem alterações nas crenças do sistema que, por sua vez, podem resultar na consideração de novos planos que não são meios para obter qualquer fim desejado anteriormente. Isto significa dizer que nem sempre PRS utiliza de um raciocínio de meios-fins.

O uso de meta-KA's pode resolver vários tipos de problemas de planejamento reativo, inclusive em relação ao tempo gasto para a satisfação de uma meta.

Como exemplo, Georgeff utiliza-se de um robô engajado numa estação espacial, preenchendo o papel de um assistente de astronauta. Neste cenário, ocorre frequentemente emergências de vários níveis de prioridade durante o processo de gerenciar outras tarefas menos críticas. O robô pode decidir que, dada a situação atual, não existe mais tempo para raciocinar e desta forma agir imediatamente.

PRS opera com uma medida bem definida de reatividade. PRS é capaz de observar a aplicabilidade de uma área de conhecimento após toda ação primitiva que ele executa. Dado que tal ação primitiva tem um tempo de execução de, no máximo t , e dado que o interpretador PRS leva no máximo um tempo s para unificação, então PRS tem um atraso de reatividade de no máximo $t + s$. Acredita-se que a sua utilização é de grande utilidade para um sistema de planejamento reativo (THIEBAUX, 1993).

Como já foi mencionado, o sistema PRS raciocina, planeja e também é capaz de reagir conforme ao estado atual (NILSSON, 1994). A sua estrutura semântica faz com que o projetista considere em detalhes os aspectos dinâmicos do domínio (HASEMANN, 1995).

Apesar de ser usado como componente em várias arquiteturas que integram planejamento e reatividade, o PRS tem sido considerado limitado por alguns pesquisadores.

Segundo Blythe (1995), PRS não representa um algoritmo de planejamento geral. De fato, a idéia presente neste sistema é que muitos dos problemas podem ser resolvidos pelo uso de procedimentos previamente codificados (Drummond, 1989).

Musliner et al (1993, 1995) afirma que o sistema não pode concentrar-se numa determinada atividade e ignorar informação sensorial desnecessária. O modelo do mundo é constantemente modificado. A resposta do sistema a um evento pode ser arbitrariamente interrompida pela chegada de outros que pode retardar o processamento anteriormente iniciado.

Como este sistema não possui a habilidade de geração de planos em face de novos objetivos, vários sistemas foram propostos com uma arquitetura constituída de duas camadas: uma superior e uma inferior. Esta possui uma estrutura similar a PRS enquanto aquela é responsável pela criação de novos planos que satisfazem novas metas e por repassá-los à camada inferior.

2.1.2.2 RAP

Uma entidade independente perseguindo algum objetivo em competição com muitos outros, em tempo de execução, é denominada por Firby (1987) como *Reactive Action Package* (RAP), ou seja, pacote de ação reativa.

Os RAP's são criados conforme são definidos os objetivos a serem alcançados; para cada objetivo há um RAP. O algoritmo proposto em RAP ataca os problemas de monitoramento de execução e replanejamento em domínios dinâmicos, usando representação e estrutura de controle simples e uniformes, conforme a Figura 2.3, na qual o modelo do mundo corresponde à descrição do mundo atual, a interface de hardware controla a comunicação com o mundo real, o interpretador RAP e a fila de execução de RAPs fornecem um mecanismo para coordenar competição entre RAPs.

Visando reduzir a busca necessária para a satisfação de uma dada meta, um RAP possui um conjunto predefinido de métodos, os quais consistem de redes parcialmente ordenadas de tarefas. Cada tarefa na rede é um comando primitivo ou uma submeta que chamará um outro RAP, conforme descreve a Figura 2.4.

Inicialmente, um RAP é selecionado pelo interpretador da fila de execução. A seleção é baseada em critérios de tempo (*deadlines*) e restrições de ordenação impostas pela rede de tarefas. Se o RAP escolhido corresponde a um comando primitivo, ele é passado diretamente para o hardware. Caso contrário, o interpretador o executa.

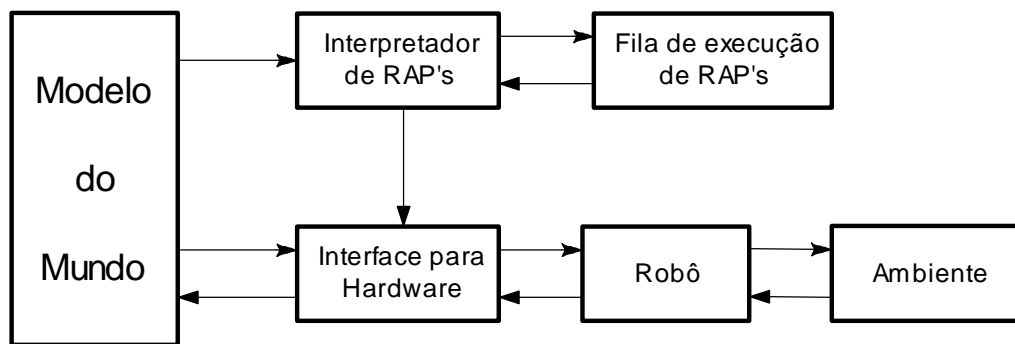


Figura 2.3 - O Ambiente de Execução de RAPs.

A constituição de um RAP é dividida em duas partes, uma que verifica os objetivos e outra que seleciona a rede de tarefas. A execução começa sempre com o verificador checando se o objetivo a ser atingido já está satisfeito no estado atual. Se estiver, o RAP termina em sucesso; caso contrário, um método é selecionado e colocado na fila de execução. Neste ponto, o RAP selecionou um plano para obter seu objetivo e deve esperar para ver como o mundo evolui. Esta espera é obtida por colocar o RAP selecionado no fim da fila de execução após o método. Quando o RAP é retirado da fila, ele é executado como antes. Portanto, um RAP continua selecionando métodos até que seu objetivo seja alcançado. Exauridas to-

das as possibilidades, o que ocorre quando não existem métodos que possam ser aplicados ao estado atual do mundo, ele sinaliza fracasso.

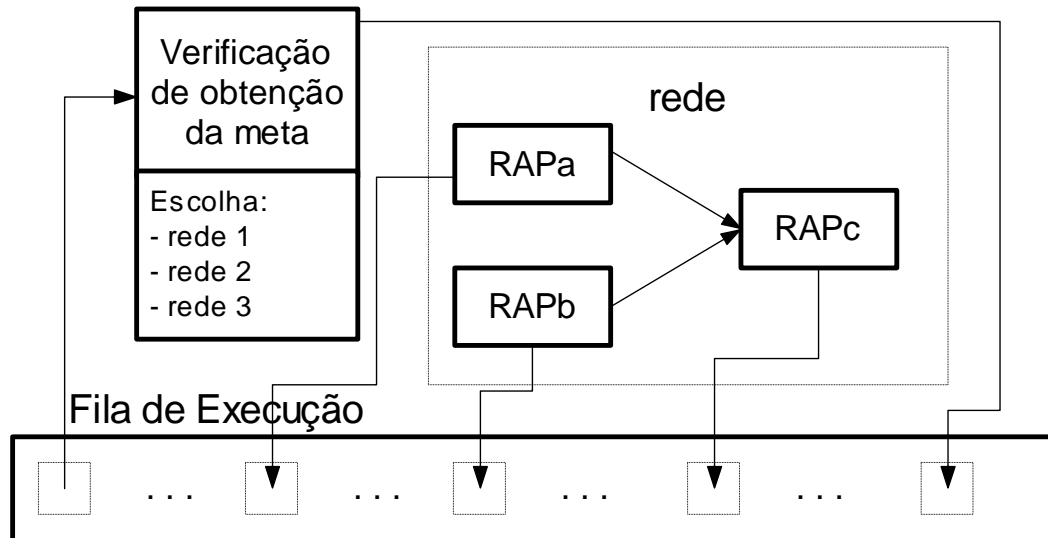


Figura 2.4 - Mecanismo de Execução de um RAP

Os passos descritos acima levantam dois tipos de problemas: o fracasso de um elemento da rede e a não alteração do mundo pelo fracasso de um método associado a um RAP. A solução para o primeiro problema consiste em manter uma relação de dependência entre os elementos da rede. Fracassando um elemento, todos os demais elementos que compõem a rede são removidos da fila de execução. O problema levantado pelo fracasso de um método diz respeito à geração de laços de execução. Se o mundo não é alterado pelo fracasso de um método, o mesmo pode ser escolhido novamente pelo mesmo RAP que o havia selecionado anteriormente. Uma possível solução para esse problema é detectar o RAP que gera um laço de execução e fazer com que ele receba uma baixa prioridade para fins de remoção da fila. O objetivo deste processo está baseado na esperança de que o mundo se altere e ele possa finalmente ser executado com sucesso. Se isto não o-

correr, o RAP então deve sinalizar fracasso para que seu pai selecione um método diferente.

Em Rap's não se considera incertezas. Logo que um comando primitivo é executado e um fracasso é identificado o hardware e a sua interface são encarregados de analisá-lo e corrigir o estado do mundo, tornando-o automaticamente consistente, por meio da realimentação corretiva do domínio e replanejamento devido às inconsistências.

Os sistemas híbridos que integram planejamento e reatividade têm utilizado RAP (HASEMANN, 1995). Entretanto, os sistemas apresentam limitações, não enfatizando critérios de tempo (MUSLINER & DURFEE & SHIN, 1993, 1995), planejamento estratégico e subsistemas de execução, visto que os dois últimos compartilham um modelo de mundo global que acarretam em conflitos, tornando o sistema lento.

Assim como PRS, RAPS também faz parte de novas arquiteturas aparecendo como uma camada inferior de uma arquitetura de múltiplas camadas.

2.1.2.3 Planos Universais

Baseado na idéia dos chamados planos universais, Schoppers (1987) propôs uma abordagem para tratar com ambientes imprevisíveis, com o objetivo de obter metas. Um plano universal especifica reações apropriadas para toda possível situação dentro de um domínio, as quais podem ser geradas automaticamente, diferenciando-se dos sistemas reativos apresentados anteriormente.

Durante a execução do plano, o interpretador obtém a ação a ser executada pela utilização do plano universal, o qual é gerado com o auxílio de um planejador

não linear (ALLEN, 1990), similar a uma árvore de decisão, a qual é percorrida em busca das pré-condições que tornam verdadeiras e executáveis as ações no mundo. Sempre que os valores das pré-condições forem modificados, a árvore é novamente percorrida, permitindo assim ao sistema tirar vantagem da ocorrência de determinados eventos externos. Suponha, por exemplo, que uma seqüência de ações se faz necessária para satisfazer determinados objetivos. A ocorrência de um evento externo pode gerar uma situação em que um número menor de ações seja necessário para satisfazer as metas. Portanto, situações benéficas resultantes da ocorrência de eventos podem fazer com que o esforço necessário para a satisfação das metas seja reduzido, uma vez que tais situações normalmente teriam que ser atingidas por meio de execução de ações apropriadas. Da mesma forma, é possível tratar com situações indesejadas surgidas intencionalmente (sabotagem). O surgimento de situações desejadas e indesejadas durante a realização de tarefas são características de um mundo dinâmico e real.

Tendo em vista que planos universais consideram todas as ações possíveis, segundo alguns autores “a idéia de um plano universal é quase universalmente equivocada” (GINSBERG, 1989). Em geral, nas aplicações envolvendo situações do mundo real, não há como prever todas as situações possíveis. Musliner et al (1993, 1995) apresenta mais algumas limitações desta abordagem:

- Planos universais não enfatizam critérios de tempo.
- Planos universais são gerados não considerando estados de mundos possíveis.
- Planos universais não raciocinam para verificar ou alterar metas.

Mesmo apresentando certas limitações, a idéia de planos universais influenciou positivamente várias pesquisas, , como por exemplo as regras de controle situadas desenvolvidas por Drummond (1989) e a adaptação incremental de reatividade proposta por Lyons em (LYONS & HENDRIKS, 1992).

2.1.2.4 Regras de Controle Situadas

Segundo Drummond o sucesso do planejamento em IA tem sido limitado devido à visão de um plano como um programa ou uma seqüência de ações, “um plano deve informar a ação ao invés de defini-la”.

Para planejar o tempo é um fator que influencia nas ações do agente. Quando se tempo de planejar a meta direciona o resultado do plano, enquanto que a falta de tempo obriga o agente a agir conforme suas habilidades comportamentais básicas. Seguindo esse raciocínio Drummond e seus colegas desenvolveram a arquitetura para agentes demonstrada na figura 2.5.

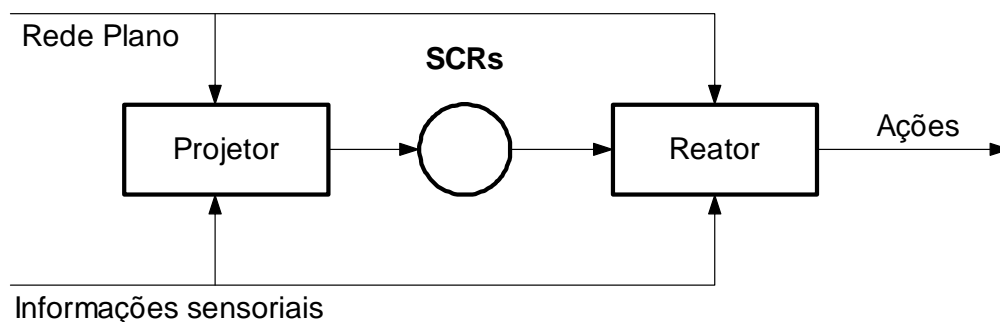


Figura 2.5 - Sistema Projetor/Reator

O componente reator é responsável pela produção de comportamento reativo. Para isto, ele recebe como dados de entrada uma “rede plano” e informações sensoriais. As habilidades comportamentais básicas do agente são descritas por meio de “redes planos” que são redes de Petri do tipo condição/ação. O reator interpreta essa rede como um programa não determinístico, escolhendo e executan-

do ações na medida em que as condições necessárias para sua execução são satisfeitas. Assim sendo, os resultados deste procedimento são imprevisíveis. Uma ordem no processo de execução pode ser definida por regras de controle situadas (SCR), as quais definem comportamentos por meio de ações e situações. A ação a ser executada é definida pela regra que teve seu antecedente verificado e validado, porém, antes dessa ação, o reator checa qual regra será aplicada à situação.

As metas são satisfeitas a partir de uma seqüência de operadores que formam um plano, o qual é de responsabilidade do componente projetor que, por sua vez, utiliza uma linguagem baseada em STRIPS (FIKES & NILSSON, 1971) para descrever estados e operadores. Sendo assim, um conjunto de ações pode ser associado a situações específicas que identificam possíveis fracassos, os quais podem ser evitados a partir da indexação das ações aos estados.

O raciocínio apresentado em algoritmos restringidos pelo tempo (*anytime algorithms*) é utilizado no processo de geração de regras SCR, que mesmo não sendo suficientes para informar todas as ações escolhidas, aumentam a probabilidade de obtenção das metas (DRUMMOND & SWANSON & BRESINA et al, 1993).

O algoritmo proposto para geração de regras SCR apresenta garantia quanto ao seu término. Entretanto, ele é exponencialmente complexo. O uso de heurísticas é sugerido para controlar o espaço de busca. Embora uma seqüência de operadores possa ser gerada por este raciocínio, isto não implica em entender o mecanismo de execução como realizando cada uma das ações sequencialmente. Isto ocorre porque ações são amarradas a estados do mundo e é o estado atual do mundo que se torna responsável pela execução de uma determinada ação.

Um das vantagens em se indexar ações a situações é o gerenciamento do fracasso de ações. Por exemplo, a ocorrência de um evento externo pode fazer com que o agente seja colocado numa situação diferente daquela imaginada ao se executar uma determinada ação. Se isto ocorre e existe uma SCR indexada ao estado resultante, o agente a executará recolocando-o novamente no caminho para satisfação das metas.

2.1.2.5 Arquiteturas de Três Camadas

Há projetos de sistemas que utilizam a técnica de desenvolvimento em camadas, nesta abordagem, iremos considerar a de três camadas: a camada deliberativa, a camada de sequenciamento e a camada reativa.

A camada deliberativa usa técnicas de raciocínio e representações clássicas em IA. Atividades nesta camada correspondem a tarefas que consomem tempo, como o planejamento estratégico de longo prazo. Sistemas de planejamento do tipo STRIPS, por exemplo, podem ser utilizados em sua implementação.

A camada de sequenciamento corresponde a um planejador que seleciona *táticas* apropriadas usando regras dependentes de contexto. Uma *tática* (também denominada tarefa) é um conjunto pré-ordenado de ações (também chamadas de comportamentos ou operadores). A camada de sequenciamento seleciona tarefas apropriadas e as executa de acordo com o relacionamento de precedência dentro da tarefa. Execução de tarefas conduz à ativação e término (o que envolve monitoramento, reconhecimento de condições de término e fracasso) de comportamentos da camada reativa. O PRS (GEORGEFF, 1987) e o RAP (FIRBY, 1987) são dois sistemas que podem ser utilizados na implementação desta camada.

A camada reativa está diretamente em contato com o ambiente. Ela é responsável pela execução das ações e também pela recepção de sinais provindos do meio externo. As abordagens descritas anteriormente na seção de sistemas reativos se encaixam bem na implementação da camada reativa.

Um exemplo de arquitetura de três camadas é a arquitetura ATLANTIS desenvolvida por Gat (1992). O componente reativo ou o controlador, termo utilizado por Gat, é responsável pelo controle de atividades ou ações implementadas por módulos computacionais e executadas sem a necessidade de deliberação. A implementação deste componente é facilitada pela utilização de uma linguagem de programação denominada Alpha (GAT, 1992). A camada de sequenciamento inicia e termina atividades por ativar e desativar os módulos do controlador. Este componente é baseado em RAP. O componente deliberativo é composto por um conjunto de programas tradicionais de IA.

2.2 Planejamento Instrucional

O procedimento adotado para mapear uma seqüência global de ações, objetivando uma meta que permita alcançar determinado grau de instrução é conhecido como Planejamento Instrucional. Esse planejamento ou plano tem como objetivo prover aprendizagem por meio de apresentação de conteúdos instrucionais de forma contínua e coerente.

O planejamento instrucional é destinado a oferecer uma interação com o estudante a partir de oportunidades instrucionais que vão surgindo durante a apresentação de um plano como, por exemplo, conceitos errados que são identificados, questões formuladas pelos estudantes, tempo gasto em determinada sessão, entre

outras. Para Wasson (1996), “*planejamento instrucional é usado para suportar interação instrucional, não para defini-la*”, expondo que as metas e planos são criadas para indicar ao sistema qual comportamento é requerido, dado um determinado estado de um ambiente de aprendizagem. Com a possibilidade de se definir qualquer comportamento por meio do planejamento instrucional, várias estratégias instrucionais podem ser utilizadas, podemos citar, como exemplo, as encontradas nos tutores tradicionais presentes em (CASAS, 1999). Esses comportamentos estão presentes na construção de sistemas para ao auxílio ao ensino, seja ele presencial ou auxiliado pelos recursos da Web.

O planejamento instrucional está presente em STI. Nos próximos subcapítulos, iremos descrever um STI e apresentar algumas técnicas de planejamento instrucional associadas a alguns STI.

2.2.1 Sistemas Tutores Inteligentes (STI)

O surgimento de sistemas tutores, destinados à prática do ensino, inicia-se com projetos conhecidos como CAI (Computer-aided Instruction). Uma ferramenta CAI tem como objetivo manipular automaticamente o conteúdo de certo domínio de conhecimento que se deseja ensinar. A forma pela qual os conteúdos eram apresentados assemelhava-se a um livro eletrônico, e a vantagem oferecida ao estudante era a liberdade de explorar os conteúdos de forma dinâmica. Uma grande restrição desse sistema é o fato de tais escolhas terem que ser pré-programadas pelo desenvolvedor do sistema, gerando assim um grande esforço na sua criação, e, mesmo assim, não garantia uma abrangência sobre os estudantes (EBERSPÄCHER & KAESTNER, 1997).

Novos esforços se multiplicaram no desenvolvimento de sistemas, gerando assim os ICAI (Intelligent CAI), também chamados STI (WENGER, 1987), que incorporam técnicas de IA.

As aplicações de IA a Educação são típicas de STI e proporcionam uma tentativa de levar o ensino tradicional para um meio computacional de forma dinâmica e individualizada, não no sentido de transformar o indivíduo em um ser individualista, e sim, oferecer acompanhamento ou tutoramento personalizado através de um processo interativo.

Os STI's oferecem flexibilidade na apresentação do conteúdo e habilidades para interagir com o estudante, buscando atender às necessidades particulares de cada um. Eles buscam informações relevantes sobre o estudante, proporcionando-lhe um aprendizado individualizado (COSTA , 1999).

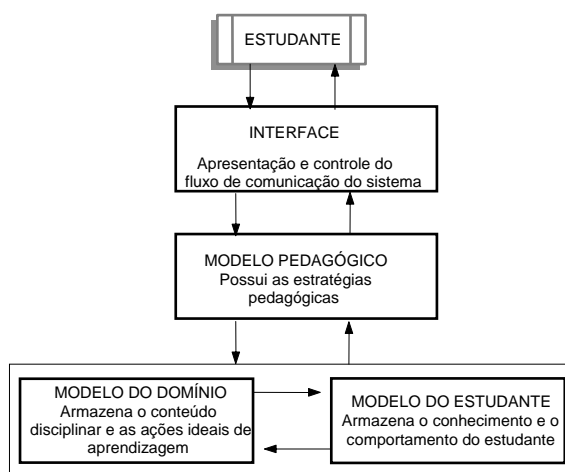


Figura 2.6 - Arquitetura de um STI

As funções operacionais básicas de um STI, como descrito em Souza (2000), são determinadas por quatro componentes principais, que são apresentados na Figura 2.6.

Modelo do Domínio: Responsável pela base de conhecimento. Ele armazena o conteúdo disciplinar organizado de alguma maneira para representar o conhecimento de um especialista ou de um professor. Uma representação do conhecimento bem organizada é fundamental para que o sistema possa realizar de forma satisfatória o ensino individualizado.

Modelo do Estudante: Armazena o comportamento, desempenho e aprendizado do estudante. As informações a serem armazenadas não são triviais; as restrições computacionais para perceber o estado atual do estudante, como por exemplo, o humor e expressões corporais interferem na construção do seu modelo ou estado atual de aprendizagem.

Modelo Pedagógico: Contém o conhecimento para tomar decisões sobre quais estratégias de ensino devem ser utilizadas dentre aquelas presentes no sistema. Essas estratégias são utilizadas levando em consideração as informações presentes no modelo de domínio e no modelo do estudante. Desta forma, as decisões sobre as informações a serem apresentadas ao estudante caracterizam o tutoramento personalizado.

Interface: É a forma como os conteúdos serão apresentados ao estudante. Chamado também de módulo de comunicação ou ambiente de ensino, é como o STI se comunica com o meio externo ao sistema.

A construção de um STI requer uma integração de todos esses quatro componentes, fazendo com que a ação de ensinar seja realmente agradável e individualizada. Os benefícios a serem alcançados são perceptíveis quando se utiliza um STI no processo de ensino e aprendizagem.

São muitas as dificuldades para a idealização de um STI. Com o objetivo de minimizar os custos em sua construção, vários esforços estão sendo conduzidos para a criação de ferramentas de autoria para STI. Os benefícios a serem alcançados com este tipo de ferramenta são (MURRAY, 1999):

- A diminuição de tempo de desenvolvimento e economia de recursos financeiro ou pessoal;
- Organização, relacionamento e estruturação automática dos conteúdos inseridos no curso;
- Deixar de forma transparente para o professor/autor a complexidade de um STI;
- Prototipação rápida e eficiente.

Murray (1999) coloca também que, conforme as características, uma ferramenta de autoria pode ser classificada em:

- Currículo: Organização e Planejamento;
- Ensinando Estratégias;
- Simulação de dispositivo e Treinamento de Equipamento;
- Domínio de Sistema Especialista;
- Tipos de Conhecimento múltiplos;
- Propósito especial;
- Hipermídia Inteligente / Adaptativa.

Conforme conclui Murray, as fronteiras entre as classificações não são bem definidas. Existem algumas ferramentas de autoria que possuem características combinadas. Se uma ferramenta de autoria, possuir em sua composição, características diversas, ela poderá oferecer um nível de interação maior.

2.2.2 Técnicas de Planejamento Instrucional em STI

O planejamento instrucional associado aos recursos presentes em STI possibilita a geração de um plano com ações o qual oferece ao sistema gerado uma consistência, coerência e continuidade que são utilizadas para apoiar o processo ensino/aprendizagem (WASSON, 1990). Esse conjunto de técnicas, planejamento e STI são conhecidos como planejador pedagógico (QUEIROZ, 2002), que, além de oferecer um plano que possa orientar o estudante na aquisição do conhecimento, possibilita a modificação do plano caso alguma dificuldade ou problema seja detectado; esse processo é conhecido como reatividade (VASSILEVA, 1995). É com foco em reatividade que estamos propondo o agente Assistente descrito no capítulo 4.

Os planos gerados pelo planejador pedagógico possuem metas que deverão ser satisfeitas a partir de informações presentes no ambiente de aprendizagem, o qual inclui dados do conteúdo a ser ensinado e dados dos estudantes. Várias estratégias instrucionais tais como Tutoria Tradicional, Aprendizes Cognitivos, Treinamento ou Coaching, Diálogo Socrático e Tutoria Negociada são utilizadas pelos planejadores (CASAS, 1999) (WASSON, 1992). Nos próximos subcapítulos, iremos descrever alguns sistemas que utilizam o planejamento instrucional em sua arquitetura.

2.2.2.1 Planejador APE

APE (ATLAS Planning Engine) é um planejador reativo baseado em diálogos por meio de linguagem natural e faz uso da arquitetura PRS, descrita na seção 2.1.2.1. A meta do Atlas é construir um sistema com base na “conversação”, na qual textos e situações vividas são inseridos no sistema por meio da “conversação”. De acordo com o autor Mills (2001), o APE pode ser usado para gerar diálogos que envolvem um discurso construído arbitrariamente. Porém, planejadores que utilizam a técnica de diálogos construídos com base em máquinas de estado-finito são mais poderosos. APE apresenta um modelo com conteúdo hierárquico e planos com múltiplos retornos. Mesmo sendo um planejador reativo, o APE não é capaz de construir um raciocínio deliberativo, como acontece em outros planejadores de diálogo.

Além do ATLAS, proposto por Mills (2001), o sistema CircSim-Tutor, aplicado no domínio de fisiologia cardiovascular, é um exemplo prático da utilização do planejador APE (CHO et al, 1999, 2000).

2.2.2.2 Planejador BEETS

O Beets é uma arquitetura genérica e modular para administração de sistemas que utilizam o diálogo para a aquisição do conhecimento. Sua arquitetura é composta por três módulos principais: interpretação, atualização e geração de resposta. Cada um dos módulos pode ter acesso a informações por meio dos diálogos globais e por interfaces com fontes de conhecimento externas como, por exemplo, modelo de estudante, bases de conhecimento, currículo e estratégias de autoria.

As tarefas realizadas pelos módulos presentes no BEETS são:

- Módulo de interpretação: permite a interação com o estudante por meio de texto e gráficos.
- Módulo de atualização: utilizado para manter o contexto atualizado.
- Módulo de geração de resposta: responsável por calcular os movimentos apropriados na forma de seminário e sintetizar realimentação desses seminários, por meio de textos ou outras modalidades. Para executar esse planejamento, ele utiliza uma arquitetura de três filas, sendo:
 - Na fila superior, está o planejador deliberativo que sintetiza os planos (uma sucessão estruturada de tarefas) em um nível de abstração alto que é passado para a fila do meio.
 - A fila do meio executa o conteúdo de uma agenda, que é alimentada pela fila superior, usando um intérprete do modelo RAP, apresentado na seção 2.1.2.2. O intérprete executa uma tarefa selecionando um método chamado de pacote de ação reativa (RAP) que deverá concluir uma tarefa. O resultado dessa chamada é o refinamento da tarefa ou uma ação primitiva. Se não foi possível concluir a tarefa, um novo refinamento deverá ser feito, isso irá gerar subtarefas que serão refinadas até que a tarefa seja concluída. São delegadas para a fila inferior as ações de diálogo primitivas.
 - A fila inferior recebe uma seqüência de atos elementares de fala e microplanos para que possa gerar a realimentação multimodal (expressões vocais de idioma naturais e ações de GUI).

BEETs foi construída usando-se duas tecnologias: o sistema de diálogo TRINDKIT e a Arquitetura Aberta de Agente (OAA).

2.2.2.3 Planejador Teórico de Decisão

Há três principais problemas a serem superados no planejamento instrucional aplicados em modelos educacionais: a imperfeição do modelo do estudante, insegurança sobre ações não esperadas e respostas inesperadas dos estudantes. Tentando

resolver isso, o planejador teórico de decisão modela essas ações como transições de estado em um modelo probabilístico, utilizando o modelo de suposição de Markov (LOVEJOY, 1991).

O método utilizado por esse planejador é representado por uma seqüência de estados, que assume as regras de instrução como sendo o conjunto finito de estados pedagógicos, um conjunto de políticas das ações, um modelo de transição de estados e as preferências dos professores em uma função de recompensa, a qual determina quais ações deverão ser ativadas conforme as transições dos estados. Um protótipo STI é implementado para avaliar o planejador teórico de decisão. O resultado é comparado com planejador hierárquico de tarefas em rede, implementado para a mesma tarefa no mesmo domínio.

2.2.2.4 Planejador PLAIT

PLAIT (PLanning Agents in Intelligent Tutoring) é um sistema desenvolvido na Universidade de Rostock, na Alemanha. Dois agentes compõem este sistema: o agente de aprendizagem e o agente tutor.

O agente de aprendizagem guia e controla o processo de ensino/aprendizagem com base nas informações fornecidas pelo agente tutor, que se baseia no comportamento do estudante por meio de um planejamento hierárquico. O agente tutor avalia as ações do estudante com o intuito de promover reações que possam auxiliá-lo dentro do contexto em que se encontra. O planejamento hierárquico é sugerido para facilitar o processo de autoria de um plano de ensino.

Há três níveis de hierarquia dentro um plano gerado por PLAIT. No nível mais alto de hierarquia, há um plano abstrato composto por tarefas que são inde-

pendentes do domínio de atuação. No nível intermediário, há um plano composto parcialmente por tarefas que dependem do domínio. No menor nível da hierarquia, existe um plano de ações ordenadas parcialmente que estão no seu mais baixo nível de especificação. De acordo com os autores, o planejamento é usado para antecipar a meta do estudante e suas intenções. Como resultado, permite decidir se o estudante comporta-se de uma maneira coerente.

2.2.2.5 Método de aproximação de Smith

Smith descreve um método de aproximação do domínio e da representação de tarefas em um ambiente de aprendizagem interativo. A aproximação entre eles está baseada nas regras de controle de situação (SCR) propostas por Drummond. O autor discute que a aproximação entre o domínio e a aprendizagem fornece ao estudante a possibilidade de avaliar o seu aprendizado efetivo. Sistemas utilizados como descobrimento de ensino, por parte do estudante, são efetivamente insuficientes, sem uma realimentação apropriada e podem levar o estudante a adquirir conhecimentos equivocados. Um tutor de VCR foi desenvolvido para ilustrar a aproximação do conhecimento com o domínio.

2.3 Planejamento Instrucional em Sistemas de EAD

Como alternativa ao ensino presencial, caracterizado pela interação pessoal entre alunos e professores em salas de aula, surgiu a Educação a Distância (EAD), composto por um conjunto de recursos tecnológicos de comunicação bidirecional o qual tem como principal finalidade promover uma aprendizagem independente e flexível. A prática da EAD possibilita que grandes contingentes de estudantes

possam ser atendidos, independente do local onde se encontram ou do tempo que dispõem (NUNES, 1994).

Dentro do conjunto tecnológico utilizado por EAD, podemos citar o uso da Internet, que se encontra em grande expansão por apresentar baixos custos de utilização e oferecer aos participantes do processo uma grande flexibilidade em relação ao local e ao horário disponíveis para a realização do curso. Os próximos subcapítulos apresentarão as metodologias utilizadas para se construir um sistema de EAD baseado em STI com planejamento.

2.3.1 Construção de Sistema de EAD em STI

Dentro do conjunto tecnológico citado no item anterior, podemos destacar o uso da Internet. A utilização da Web como instrumento de apoio a EAD está sendo altamente explorada, por meio da criação de sítios especializados e ferramentas de apoio que demonstram certo grau de inteligência (CAPUANO & MARSELLA & SALERNO et al, 2000). Essa utilização está classificada em três tipos de metodologias, que são:

Sistemas Estáticos: é a prática de disponibilizar os materiais de ensino por meio de páginas estáticas, criadas pelos autores dos cursos, na qual eles disponibilizam o material em uma seqüência que deverá ser seguida por todos os estudantes que navegarem na página.

Sistemas Personalizados: possuem as mesmas características de sistemas estáticos no que diz respeito a disponibilizar o material do curso ao aluno, a diferença ocorre na possibilidade do autor do curso interferir com auxílio de um software, por meio de inserções de materiais de ensino auxiliar, com o intuito de aju-

dar o estudante na evolução do curso. Essas interferências ocorrem tendo como base a observação do estado cognitivo do estudante.

Sistemas Adaptativos: basicamente é um sistema personalizado, com a diferença que todas as ações de interferência no curso são executadas automaticamente por um agente. Essa característica é alcançada por meio da utilização de técnicas de IA. A maioria dos sistemas desenvolvidos baseia-se em STI's adaptados para WEB. Esta característica está presente no trabalho que foi desenvolvido.

Entre os três tipos de sistemas citados, o *estático* é o que mais agrupa sistemas educacionais existentes, pois a grande maioria dos cursos disponíveis na Web não utiliza técnicas de IA ou não considera o estado cognitivo do estudante (BRUSILOVSKY, 1999).

Considerando que os participantes de um curso a distância são heterogêneos e que a apresentação de um curso, capaz de atender às expectativas de cada um, não é uma tarefa trivial, a utilização da metodologia de sistemas adaptativos passa a ser de fundamental importância para que o aprendizado personalizado possa ocorrer (BRUSILOVSKY, 1999).

Os sistemas adaptativos sugeridos por Brusilovsky devem possuir a capacidade de interagir com diferentes níveis de conhecimento e com os anseios de aprendizagem bastante variados. A principal característica, apontada pelo autor, é a capacidade de um sistema se adaptar ao perfil do estudante, independentemente do nível de conhecimento deste sobre o domínio ensinado. Para realizar esta tarefa, é necessário construir um modelo que possa armazenar informações essenciais a serem utilizadas durante a participação do estudante no curso (BRUSILOVSKY,

1999) (QUEIROZ & LOPES & FERNANDES, 2002). Entre as principais informações presentes em um modelo de ensino, estão o modelo do domínio e o modelo do estudante, descritos a seguir:

O modelo do domínio presente em STI tem como papel principal identificar que tipo de conhecimento deverá ser aprendido; isso é possível por meio da representação do conteúdo do domínio em componentes, tais como fatos, princípios, conceitos e procedimentos. A relação existente entre os componentes também é levada em consideração, podendo ser dependências lógicas e hierárquicas. Para a pré-especificação do domínio, leva-se em consideração duas suposições básicas da prática de instrução (QUEIROZ, 2003):

- Compreensão do domínio: é o pressuposto de uma correspondência entre o estudante e o planejador instrucional, de como eles compreendem o domínio.
- Compreensão dos conceitos: é aceito que conceitos amplos podem ser aprendidos por meio da sua representação em partes, ou seja, a compreensão do conjunto de subconceitos auxilia no aprendizado de um conceito maior.

No capítulo quatro dessa dissertação, demonstramos como um planejador instrucional representa os objetos de ensino e suas relações para que possam ser utilizados durante a geração e utilização de um plano de ensino.

O modelo do estudante é responsável por identificar as características do estudante como, por exemplo, suas preferências de navegação, conhecimentos pré-adquiridos e seu histórico de navegação. Todas essas informações ocorrem de forma dinâmica por parte do estudante, por isso é necessário que o sistema proposto venha atualizar continuamente o modelo do estudante (PALAZZO, 2000)

(QUEIROZ, 2003) (DORÇA, 2004). As principais características a serem observadas são:

- **Conhecimento do estudante:** é a representação do conhecimento prévio do estudante sobre o domínio o qual ele está atuando. De acordo com Brusilovsky (1996), este conhecimento é fundamental para se obter uma personalização adequada.
- **Histórico do estudante:** é o armazenamento sobre os passos do estudante durante a visualização do curso como, por exemplo, quais recursos são mais utilizados por ele, que tipo de material instrucional é mais visitado por ele ou como ele se comporta durante a visualização dos conteúdos.
- **Preferências do estudante:** As preferências diferem das duas características anteriores por não ser possível ou viável deduzir automaticamente quais são as preferências do estudante. Cabe ao participante do curso informar ao sistema, por meio direto ou por *feedback*, qual é o seu perfil de usuário para que o sistema possa oferecer uma personalização próxima do ideal (BRUSILOVSKY, 1999).

A utilização dessas informações para a personalização ou adaptatividade em um sistema de EAD tem como intuito orientar o estudante dentro do processo de ensino/aprendizagem. Para Brusilovsky, a adaptatividade de um sistema é alcançada pelo uso das técnicas de IA como, por exemplo, planejamento instrucional e STI. São propostas duas formas de adaptatividade (BRUSILOVSKY, 1996):

Adaptatividade por apresentação: o conteúdo a ser apresentado ao estudante é gerado a partir de informações presentes no modelo do estudante que são aplicadas ao modelo do domínio, gerando assim um plano contendo somente os objetos de ensino relevantes para o contexto atual de aprendizagem. Além da apresentação dos conteúdos, é possível também a apresentação de outros recursos

conforme as preferências do estudante. Além do SIMEDUC, apresentado no capítulo 3, o sistema PUSH, proposto em (ESPINOZA & HÖÖK, 1996), são alguns exemplos de sistemas que utilizam esse modelo de adaptatividade.

Adaptatividade por navegação: esse tipo adaptatividade consiste em orientar o estudante no processo de visualização dos conteúdos, podendo limitar o seu espaço de atuação com o intuito de intensificar o aprendizado em uma região ou incentivá-lo a avançar no curso por meio de observações de sucessos na conclusão de etapas de aprendizado. Ações presentes na arquitetura desse modelo são basicamente a de modificar as páginas de conteúdos conforme o estudante vai evoluindo no curso.

A proposta dessa dissertação é apoiada nesses dois princípios de adaptatividade. Utilizamos os conceitos presentes em Adaptatividade por Navegação para promover o processo de reatividade apresentado no capítulo quatro. Nos dois próximos subcapítulos, apresentaremos, em detalhes, os sistemas ABITS e Tobie que também fazem uso da adaptatividade para se promover o aprendizado.

2.3.2 Sistema ABITS

A proposta apresentada no sistema ABITS sugere uma arquitetura inteligente de ensino altamente reutilizável (BRUSILOVSKY & VASSILEVA, 2003), podendo ser aplicada em diversos domínios de conhecimento. É proposto ainda a utilização de funções “inteligentes” responsáveis por identificar o modelo do estudante e gerar, assim, um currículo personalizado, podendo utilizar a Web como veículo de execução.

Podemos encontrar também, na arquitetura do ABITS, técnicas de IA por meio da utilização de agentes inteligentes, estando presente três tipos de agentes: de avaliação, pedagógico e afetivo. A utilização desses agentes oferece ao sistema a capacidade de se adaptar ao ambiente de atuação. A Figura 2.7 traz o modelo proposto do ABITS, no qual o núcleo central do ABITS se comunica com um Sistema de Gerenciamento de Cursos – SGC e essa comunicação é realizada por Invocações Remotas de Métodos (RMI). A integração entre os dois módulos possibilita o compartilhamento das mesmas bases de dados.

O conhecimento no ABTIS é representado por meio de objetos de aprendizagem identificados como entidades, que podem ser utilizadas de diversas formas no processo de aprendizagem. Para que a utilização dos objetos seja eficiente, é necessário um modelo de indexação, favorecendo assim a identificação e a utilização dos objetos. O modelo utilização foi o LTSC LOM (Learning Object Metadata) da IEEE o qual é capaz de armazenar informações essenciais para a reutilização dos objetos (CAPUANO & MARSELLA & SALERNO et al, 2000). Os estados cognitivos e as preferências dos estudantes são identificados por meio de modelagem, utilizando-se os conjuntos *Fuzzy*.

Os estudantes dos cursos oferecidos pelo ABITS podem participar de vários outros cursos e todos esses são compostos por um conjunto de metas de aprendizagem e por um conteúdo. As metas definem qual será o estado final de aprendizagem do estudante no final do curso e o currículo é utilizado para orientar a aplicação dos objetos de aprendizagem que irão satisfazer a meta estabelecida no mesmo. Há também recursos que verificam o estado cognitivo do estudante antes de gerar o currículo do curso, isto garante que cada estudante possa ser atendido

individualmente. Esse processo é executado pelo agente pedagógico em três passos:

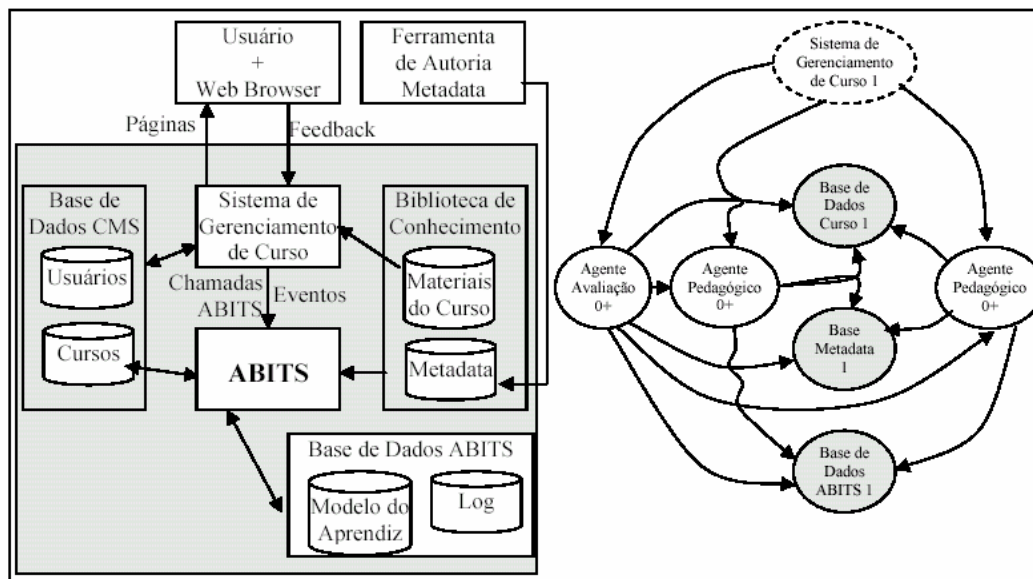


Figura 2.7 - Arquitetura do Sistema ABITS

1. **Identificação de conceitos:** com base em um conjunto de conceitos de um determinado conteúdo, o sistema faz uma intersecção entre esse conjunto, a meta de aprendizagem informada e entre o estado cognitivo do estudante. O resultante dessa intersecção será um conjunto de conceitos que deverão ser aplicados na geração do currículo do curso.
2. **Obtenção dos objetos de aprendizagem:** com o conjunto de conceitos já definido, o sistema irá construir uma seqüência de objetos de aprendizagem, seguindo como referência as preferências dos estudantes e os pré-requisitos de cada objeto, ou seja, se para apresentar um objeto, há a necessidade de se apresentar outro objeto que é requisito, o sistema irá inseri-lo no plano.
3. **Geração do currículo:** depois da organização dos objetos de aprendizagem em uma ordem que atende ao estudante e satisfaz a meta estabelecida, o sistema irá gerar o currículo a ser apresentado ao estudante, inserindo também os objetos de avaliação que irão avaliar e atualizar o estado cognitivo do estudante, por meio da atualização da base de dados modelo do estudante.

Uma característica positiva do sistema ABITS é a possibilidade de integrá-lo a outros SGC, formando outros sistemas.

2.3.3 Sistema Tobie

O sistema Tobie (VASSILEVA, 1995) possui características de um STI e utiliza a técnica de planejamento de conteúdos para interagir com os usuários. Em sua arquitetura, demonstrada na Figura 2.8, encontramos uma base de conhecimento, um componente pedagógico e o modelo do estudante.

A base de conhecimento do Tobie lembra muito um grafo AND/OR, que são grafos nos quais um único arco pode apontar para um conjunto de nós sucessores (RICH, 1988), oferecendo, assim, uma linguagem de representação na qual conceitos e metas podem ser representados (VASSILEVA, 1995).

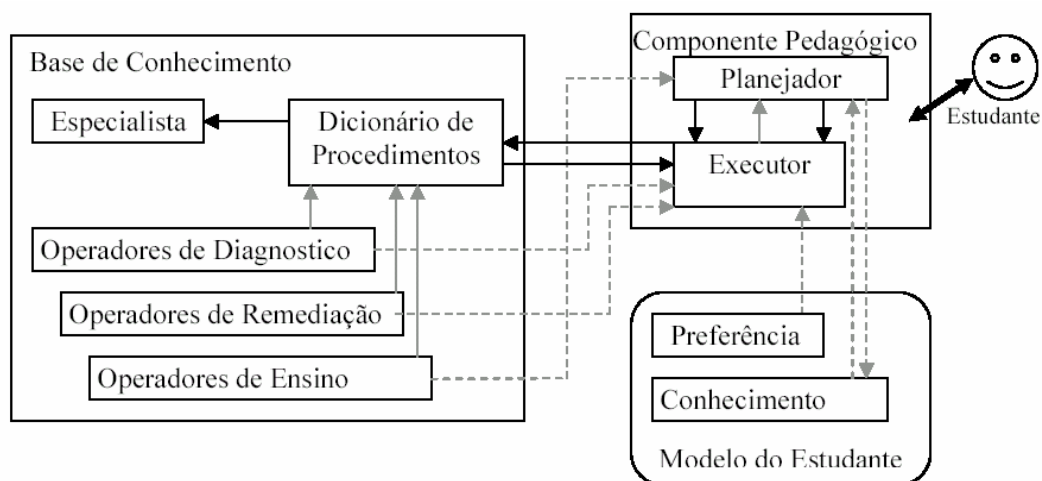


Figura 2.8 - Arquitetura do Sistema Tobie

São utilizados em Tobie Operadores de Ensino – TO's para representar a base de conhecimento, esses operadores são do tipo STRIPS, divididos em seis partes:

- um nome;
- uma lista de pré-condições;
- uma lista de efeitos;
- uma ação de ensino, no qual há a biblioteca de procedimentos de ensino;
- um campo de diagnóstico, utilizado para verificar o nível de conhecimento dos estudantes;
- parâmetros que descrevem as ações instrucionais, armazenados nos campos **tipo**.

O módulo que interage com o estudante é conhecido como componente pedagógico composto por dois subcomponentes que juntos têm a função de gerar e executar a apresentação de um plano, podendo também reagir conforme as ações dos estudantes. Essa reação torna o Tobie um sistema reativo, característica que pode ser visualizada na Figura 2.9.

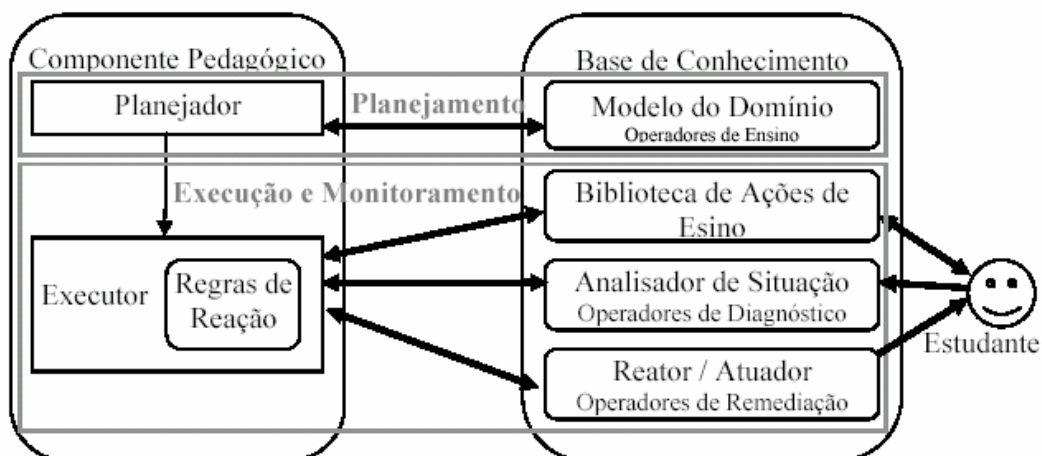


Figura 2.9 - Planejamento Reativo em Tobie

Com a capacidade de realizar buscas heurísticas no grafo AND/OR, por meio do algoritmo AO*, o subcomponente planejador presente no componente pedagógico é capaz de construir o plano inicial a ser apresentado ao estudante e

fica sob a responsabilidade do subcomponente executor de apresentá-lo por meio da aplicação dos operadores TO's.

Durante a execução do plano, o estudante pode apresentar dificuldades ou provocar eventos que não estavam sendo esperados no plano como, por exemplo, solicitar auxílio sobre um assunto que não conseguiu assimilar. Caso isso ocorra, o sistema tem a capacidade de promover reações, visando atender ao evento que provocou a chamada (VASSILEVA, 1995).

As reações presentes em Tobie são classificadas em quatro tipos: ignorar a situação; escolher uma reação oportunista sem modificar o plano; fazer um reparo no plano ou fazer um replanejamento. A aplicação correta das reações é orientada pelo conjunto de regras de reação.

2.4 Considerações Finais

Conforme descrito anteriormente, a maioria dos sistemas de planejamento instrucional, considerando o aspecto dinâmico do processo educacional, foram desenvolvidos para domínios envolvendo diálogo. Esses sistemas apresentam arquiteturas geralmente compostas por duas ou mais camadas, sendo que a camada de mais baixo nível é baseada em PRS ou RAPS. Tais camadas apresentam reatividade sem, contudo, apresentar a capacidade de geração de novos planos.

Entendemos que deva haver uma integração entre planejamento e reatividade. Para isso, trabalhamos em uma arquitetura multiagente onde aparecem o agente Pedagógico e o agente Assistente. O primeiro agente é o responsável pela geração de planos enquanto o segundo pela interação com os usuários. O agente Assistente, que será descrito nos próximos capítulos, baseia-se

em regras de controle situadas. Neste sentido, a proposta apresentada por Smith possui certa similaridade com a nossa. Contudo, o nosso trabalho aplica-se a educação a distancia e baseia-se numa arquitetura multiagente.

Uma arquitetura multiagente parece adequar-se bem ao processo de ensino/aprendizagem. Dessa forma, o trabalho a ser descrito foi inspirado nessa arquitetura, sofrendo influências de sistemas como o ABITS.

Capítulo 3

Proposta para um Sistema de Ensino a Distância

Com o intuito de colaborar com a educação a distância apoiada na Web está sendo proposto o Sistema Multiagente para o Ensino a Distância – SIMEDUC. Propostas como esta diferem das existentes no mercado pela sua capacidade de interagir com o estudante por meio do uso de Agentes Inteligentes (AROYO & KOMMERS, 1999). Criado a partir dos conceitos de STI (MURRAY, 1999) e sua comunicação interna apoiada em Sistemas Multiagentes – SMA (COSTA, 2000), ele possui um conjunto de características consideradas inovadoras para sistema de ensino apoiados na Web.

Podemos citar como sendo fortes características do SIMEDUC a possibilidade de interagir com o estudante a partir da detecção do seu perfil de estudo, a geração automática de planos de cursos, por ser uma sociedade multiagente e ser reativo conforme a navegação em seus cursos. Há também características similares às ferramentas existentes tais como navegação por meio de um navegador Web comum, sem a necessidade de instalar qualquer programa auxiliar, ferramentas de interação comuns aos sistemas como, por exemplo, chat, correio eletrônico, agendas, entre outras.

A presença desses recursos facilita o uso do SIMEDUC, possibilitando que os participantes dos cursos possam estar visualizando-os de qualquer máquina e a qualquer momento. Nos próximos subcapítulos iremos apresentar a arquitetura do sistema SIMEDUC e, logo após, sua característica multiagente.

3.1 Arquitetura do Sistema SIMEDUC

A estrutura do SIMEDUC é baseada em STI e sistemas multiagentes, cujo objetivo é gerar uma aplicação avançada para educação a distância baseada em Web que ofereça algum grau de inteligência e adaptatividade¹. O seu desenvolvimento foi dividido em dois módulos principais. Inicialmente, foi desenvolvido o Sistema Gerenciador de Curso (SGC) que é similar à maioria dos sistemas existentes e encontra-se em desenvolvimento a segunda parte, o SIM, que é responsável por fornecer adaptatividade e inteligência por meio da introdução de agentes (DORÇA, 2003).

A integração do SIM com o SGC resulta no SIMEDUC. Nos próximos subcapítulos descreveremos com maiores detalhes as funcionalidades e interações entre os dois ambientes.

3.1.1 Sistema Gerenciador de Cursos

O módulo do SIMEDUC conhecido como SGC tem como funcionalidade gerenciar os conteúdos dos cursos, esse controle é feito por dois ambientes, o de gerenciamento de conteúdos conhecido como Ambiente Administrativo e o de apresentação de conteúdo, conhecido como Ambiente de Curso.

¹ Capacidade que o sistema oferece de se adaptar às necessidades particulares do estudante. Estas adaptações estão aqui relacionadas ao seqüenciamento do conteúdo, preferências e apoio personalizado ao estudante.

3.1.1.1 Ambiente Administrativo

Manipular e introduzir os conteúdos para os cursos destinados a Web, utilizando um STI, não é uma tarefa fácil (MURRAY, 1999), para minimizar as dificuldades o SGC oferece um ambiente administrativo que recebe todas as informações necessárias para a montagem de um curso, composto por: gerenciador de testes, gerenciador de cursos, gerenciador de usuários, gerenciador de estudantes e ferramenta de autoria. As funcionalidades desses gerenciadores são:

- **Gerenciador de testes:** oferece recursos para que os usuários do sistema, preferencialmente para aqueles cadastrados como professores/autores, possam montar e gerenciar as avaliações dos cursos.
- **Gerenciador de Cursos:** oferece recursos para que os usuários possam incluir os conteúdos dos cursos, com a possibilidade de utilizar e compartilhar materiais com outros cursos já cadastrados.
- **Gerenciador de Usuários:** responsável pelo cadastramento e gerenciamento das pessoas envolvidas no processo de criação dos cursos, classificados como administradores, professores/autores e tutores. Cada um com seus níveis de acesso e permissões.
- **Gerenciador de Estudantes:** oferece recursos que auxiliam no gerenciamento das informações relacionadas aos estudantes como, por exemplo, relatório de evolução, alteração de cadastro, matrícula em cursos, entre outras. Por meio dessas informações, é possível saber quais cursos estão sendo mais visitados, tempo de navegação dos estudantes e quais ferramentas são mais utilizadas.

- **Ferramenta de autoria:** destinada a auxiliar o professor/autor na inclusão de conteúdos dos cursos. Essa ferramenta possibilita que os conteúdos dos cursos sejam inseridos nas bases de dados do sistema. O armazenamento é organizado de tal maneira que vários cursos podem utilizar um único tópico de determinado conteúdo.

Entre as vantagens de utilizar os gerenciadores está a possibilidade de economizar tempo no desenvolvimento de páginas para os cursos, oferecer transparência de uso ao professor/autor e estabelecer uma organização dos conteúdos de forma que dependências ou pré-condições dos conteúdos sejam informadas (MOREIRA, 1982). A Figura 3.1 traz a tela principal do ambiente de administração, na qual podemos verificar também a presença de ferramentas de comunicação, tais como: fórum, chat, mensagens e quadro de avisos, essas funcionalidades favorece a comunicação entre os participantes do processo de criação e manutenção dos cursos e os estudantes.

O SGC possui três bases de dados que são utilizadas para armazenar as informações geradas, representadas na Figura 3.2. A base de dados *administrativa* é responsável pelos cadastros dos estudantes, professores e monitores, pelas mensagens enviadas por meio das ferramentas de comunicação bate-papo, fórum, correio interno, bloco de notas e mural de avisos. Outra base de dados é a de *conhecimento* que possui diversas informações sobre os cursos a serem ministrados, seus conteúdos e as avaliações a serem aplicadas aos estudantes. A terceira e última base de dados é a *histórico do estudante* que traz informações sobre a navegação dos estudantes, como por exemplo, o tempo que ele ficou conectado a um cur-

so, quais e quantas mensagens foram enviadas, em qual lição dos cursos ele parou, entre outras.



Figura 3.1 - Tela principal do Ambiente Administrativo do SGC

As principais funcionalidades do Ambiente Administrativo são de organizar os cursos a serem visitados pelos alunos, inserindo os conteúdos e estabelecendo os pré-requisitos dos conteúdos, definir o período do curso, habilitar os tutores que irão auxiliar nas orientações dos estudantes e definir as avaliações que deverão ser aplicadas no final de cada lição ou curso, que servirão como referência para identificar o estado cognitivo do estudante.

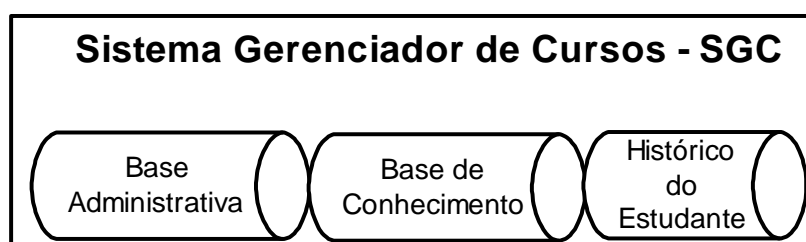


Figura 3.2 - Arquitetura do SGC do SIMEDUC

3.1.1.2 Ambiente de Curso

O Ambiente de Curso tem o papel de apresentar aos estudantes os conteúdos do curso, que é apresentado de forma seqüencial e estática. Para executar esta tarefa ele cruza as informações presentes nas bases de dados conhecimentos e histórico do estudante possibilitando assim que os estudantes possam visualizar os seus cursos sempre do ponto onde pararam. Além desse recurso, o Ambiente de Curso oferece outras funcionalidades que têm por objetivo auxiliar os estudantes no processo de ensino/aprendizagem, são elas:

- **Ferramentas de comunicação:** possui várias ferramentas de comunicação que são comuns em ambientes Web, facilitando assim interação entre os demais participantes do curso como a troca de experiências com os professores/autores e tutores, pois são de fácil utilização e conhecidas pela maioria dos internauta, podemos citar como exemplo o chat e o correio eletrônico.
- **Material de apoio:** são sugestões dos professores/autores sobre conteúdos que venham auxiliar na assimilação dos conteúdos do curso, estes materiais podem ser um arquivo com informações ou links para outras páginas que podem ser internas ou externas ao ambiente SIME-DUC.
- **Quadro de avisos:** utilizado para que os participantes possam postar informações relevantes sobre os cursos podendo ser, por exemplo, o agendamento de um encontro no chat.



Figura 3.3 - Tela Principal do Ambiente de Curso do SGC

A Figura 3.3 Apresenta a tela na qual os estudantes acessam o curso e têm a sua disposição as ferramentas de apoio.

Além de visualizar os cursos podemos citar algumas ações que os estudantes podem fazer durante a participação em um curso. Segue um breve relato dos recursos disponíveis para os estudantes no SGC: Ao se matricular em um curso, o estudante passa ser observado pelo gerenciador, que irá apresentar o curso sempre do ponto onde o estudante parou, ou seja, cada lição completada é memorizada e quando o estudante sair do curso e voltar ele sempre voltará à lição em que ele estava. As ferramentas de comunicação são sempre direcionadas para o curso que está sendo visualizado, as mensagens postadas no fórum, por exemplo, serão vistas somente pelos estudantes que estiverem matriculados no mesmo curso. O professor/autor ao disponibilizar algum material de apoio para determinado curso, todos os participantes do curso visualizarão o material. Só poderão participar dos chats os estudantes do mesmo curso, a cada curso um novo grupo de participantes nos chats. A caixa de mensagem funciona com os mesmos critérios, somente apa-

recerção na lista de destinatários os participantes, tutores e professores do curso que está sendo apresentado no momento.

Apesar de todas as funcionalidades do SGC, ele não é capaz, na apresentação de um curso, de promover ações que possam modificar a forma em que o curso é mostrado, pois ele não leva em consideração as experiências anteriores dos estudantes, por esse motivo todos os participantes dos cursos terão a mesma sequência de apresentação. Visando superar essas dificuldades foi adicionado ao SGC recursos de adaptação inteligente orientados por agentes inteligentes, identificados como Sistema Inteligente Multiagente – SIM, que será descrito em maiores detalhes no próximo subcapítulo.

3.1.2 Sistema Inteligente Multiagente

O SIM é responsável por fornecer adaptatividade e inteligência por meio da introdução de agentes. Este componente é responsável por todo o processamento pedagógico como: sequenciamento de conteúdo, avaliação do conhecimento adquirido pelo estudante, avaliação do comportamento do estudante durante o curso e fornecimento de ajuda ao estudante, quando necessário. A integração do SIM com o SGC resulta no SIMEDUC, um ambiente de educação a distância baseado em Web com características de inteligência e adaptatividade (DORÇA, 2003). São quatro os agentes que compõem a arquitetura do SIM, são eles: o agente de Avaliação, o agente Especialista, o agente Pedagógico e o agente Assistente, sendo esse último agente o objetivo desta proposta. A Figura 3.4 representa graficamente a ligação desses agentes dentro do SIMEDUC.

O agente de Avaliação tem como funcionalidade gerenciar as atividades relacionadas com a avaliação formal dos participantes do curso, entre elas podemos citar a aplicação dos testes elaborados pelo professor/autor e as atualizações no modelo do estudante que é utilizado pelos agentes Pedagógicos e Assistentes para definir o grau de conhecimento do estudante sobre determinados conteúdos.

Além de apresentar o curso aos estudantes o agente Especialista oferece suporte ao agente de Avaliação. Ao receber do aluno as respostas sobre os questionamentos o agente Especialista envia-as ao agente de Avaliação que irá compará-las com os valores internos, gerando assim um resultado que poderá ser utilizado pelo agente Assistente para identificar o nível de instrução do estudante. O agente Assistente, juntamente com o Pedagógico serão apresentados com maiores detalhes no capítulo 4.

O modelo apresentado na Figura 3.4 demonstra como a comunicação entre os agentes é realizada, que podemos descrever da seguinte maneira:

- Comunicação entre o agente Assistente e agente Pedagógico (1): A troca de informação entre estes dois agentes tem o intuito de apresentar o plano de curso que foi gerado para o estudante.
- Comunicação entre o agente Assistente e o SGC (2): é responsável por observar as ações dos estudantes e, com base nas informações de outros agentes, promover a adaptatividade na apresentação dos cursos.

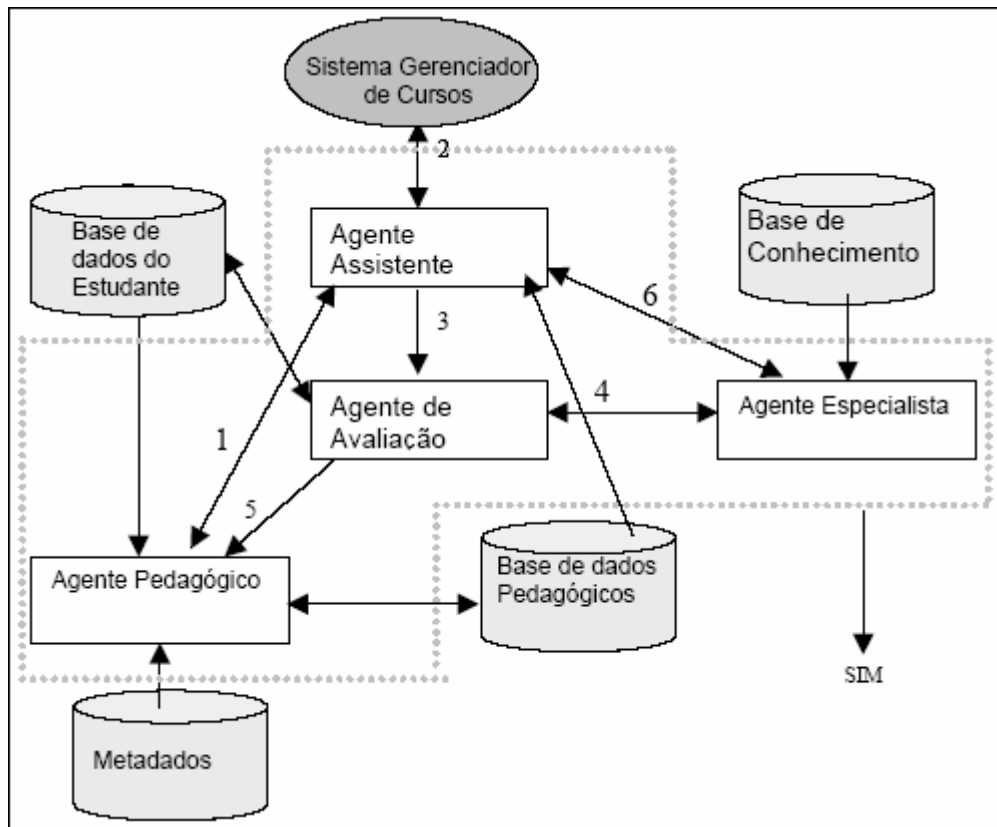


Figura 3.4 - Sistema Multiagente do SIMEDUC

- Comunicação entre o agente Assistente e o agente de Avaliação (3): Quando os objetos de ensino que tratam das avaliações dos estudantes são identificados pelo agente Assistente o mesmo comunica-se com o agente de Avaliação passando as respostas dos estudantes que serão verificadas e classificadas conforme os níveis de acerto.
- Comunicação entre o agente de Avaliação e o agente Especialista (4): estes dois agentes têm como finalidade principal corrigir e classificar os níveis de conhecimento dos estudantes. O primeiro envia os dados da avaliação e o segundo verifica quais alternativas estão corretas e devolve ao primeiro.

- Comunicação entre o agente de Avaliação e o agente Pedagógico (5): a responsabilidade do agente Pedagógico é de gerar planos personalizados com base no histórico do estudante e pelo seu estado cognitivo, para completar corretamente este processo ele recebe do agente de Avaliação o nível de conhecimento do estudante em determinados objetos de ensino.
- Comunicação entre o agente Assistente e o Agente Especialista (6): Durante a comunicação do agente Assistente com o SGC é possível que o estudante queira esclarecer algumas dúvidas, isto ocorre a partir de situações no plano que possibilita ao estudante formular perguntas, que deverão ser respondidas pelo agente Especialista.

As comunicações entre os agentes ocorre por meio de envio de mensagens gerenciadas por um programa facilitador, o formato das mensagens é simples e bem definido e tem a característica fundamental de ser interpretada por todos os agentes do sistema. O agente, ao receber a mensagem, verifica se há conteúdo na estrutura da mensagem que lhe pertence, caso encontre, ela é lida e retirada da estrutura.

O sistema SIMEDUC foi desenvolvido para ser executado em um servidor Web que possua suporte para uma plataforma multiagentes em que as comunicações entre os agentes são gerenciadas pela plataforma distribuída JADE (DORÇA, 2004).

3.2 Comunicação entre os Agentes

Os agentes inteligentes desenvolvidos para o SIMEDUC foram implementados por meio da plataforma multiagente JADE (BELLIFEMINE et al, 2000) (BELLIFEMINE et al, 2002), esses agentes podem ser executados em locais diferentes, pois a JADE é uma plataforma distribuída permitindo o compartilhamento de informações mesmo estando em locais diferentes. Para que isso ocorra a comunicação é feita por meio de RMI, fornecidas pelo JAVA.

Entre as várias classes que compõem a JADE, a responsável pela comunicação entre os agentes é a *classe Agent*, que utiliza as especificações FIPA (FIPA, 2000) nas trocas de mensagens assíncronas. Os endereços de localização dos agentes são relativos e para resolver e identificar o endereço físico de cada agente é utilizado o agente facilitador que tem a capacidade de verificar os agentes registrados nos endereços do sistema, permitindo assim que os agentes possam estar em locais diferente da rede. O facilitador é uma rotina interna da JADE identificada como Directory Facilitator – DF (DORÇA, 2004).

Na criação dos agentes é respeitada a unidade entre eles, ou seja, somente poderá existir um agente Assistente, um agente Pedagógico e assim por diante. Eles também podem ser iniciados em um local e se moverem para outro, esse processo é realizado por meio da serialização do seu código e estado. Contudo, os agentes do SIMEDUC não possuem características de mobilidade, podendo isso ser necessário caso o sistema venha a se tornar mais complexo.

Entre várias informações que o agente pode transportar têm aquelas que garantem o tratamento individualizado dos estudantes como, por exemplo: a sua identificação e qual curso estão realizando, garantindo assim que um agente ativo

possa atender a diferentes estudantes em diferentes cursos simultaneamente. Caso o tempo de processamento torna-se desfavorável em decorrência do aumento de participantes e de cursos, se faz necessário iniciar cópias dos agentes, essa tarefa poderá ser realizada por meio da técnica de balanceamento de carga, uma das propostas futura a ser implementadas no SIMEDUC.

Como mencionamos no início desse capítulo, o SIMEDUC é composto por dois módulos, o SIM e o SGC, e como eles são independentes um do outro e o SGC não é uma arquitetura de agente, a comunicação entre eles difere da comunicação utilizada entre agentes. Para que haja troca de informações entre o SIM e o SGC foi estabelecida uma comunicação baseada em *Blackboard* (quadro negro). Esse recurso foi implementado utilizando duas tabelas para o armazenamento das mensagens, uma delas é utilizada para armazenar as informações sobre os usuários, alimentada pelo SGC, e a outra é alimentada pelos agentes trazendo dados que foram coletados por meio das ações dos estudantes. Os dados nas tabelas são escritos e lidos por meio de uma fila FIFO, que após serem lidos os mesmos são removidos da fila. O modelo de comunicação Blackboard é mostrado na Figura 3.5.

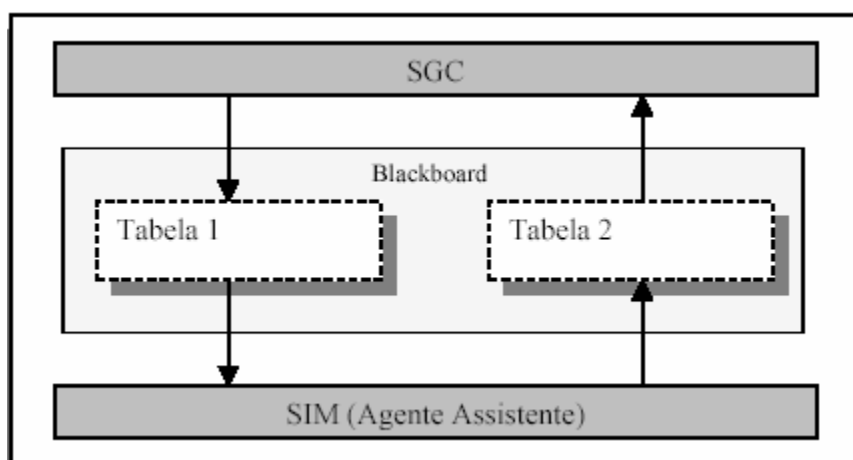


Figura 3.5 - Modelo de comunicação Blackboard do SIMEDUC

As mensagens presentes nas tabelas 1 e 2 da Figura 3.5 possui informações sobre o modelo do estudante e do curso, tais como: identificador do estudante, identificador do curso, identificador do objeto de ensino, tempo gasto pelo estudante, frequência de utilização das ferramentas presentes no SIMEDUC, respostas das avaliações, entre outras. Os valores presentes nas mensagens variam conforme o comportamento dos estudantes e em qual curso ele está. Na Figura 3.6, podemos ver um exemplo dessa troca de mensagens por de um histórico de comunicação entre o SIM e o SGC.

```

C:\WINDOWS\System32\command.com
AG.ESPECIALISTA - Nao existem mensagens pendentes na fila. Aguardando.
AG.AVALIACAO - Nenhuma mensagem pendente na fila. Aguardando...
AG.ASSISTENTE - Enviar mensagem FIMUNIDADESIMPLES para AGENTE AVALIACA
Procurando registro do Ag.Avaliacao no DF...
Agente Ag.Avaliacao encontrado. Enviando mensagem...
AG.AVALIACAO - Mensagem recebida FIMUNIDADESIMPLES<FROM: AG.ASSISTENTE
codigoObjeto: 31
codigoCurso: 27
codigoUsuario: 93
tempoGasto: 9
IndiceForum: 1
IndiceChat: 2
IndiceCorreioAluno: 3
IndiceCorreioProfessor: 4
NumeroConsulta: 5
AgAssistente enviou mensagem para Ag.Avaliacao
AG. AVALIACAO - Parametros para avaliacao nao foram encontrados.
AG.AVALIACAO - Enviando mensagem RETORNOAVALINFORMAL (TO: AG.ASSISTENT
Procurando registro do Ag.Assistente no DF...
AG.ASSISTENTE - Nao existem mensagens pendentes na fila. Aguardando...
AG.ASSISTENTE - Nenhuma mensagem pendente no Blackboard do CMS. Aguard
Agente Ag.Assistente encontrado. Enviando mensagem...
AgAvaliacao enviou mensagem para Ag.Assistente
AG.ASSISTENTE - Mensagem recebida - Resposta do Ag.Avaliacao - RETORNO
AL
codigoObjeto: 31
codigoCurso: 27
codigoUsuario: 93
conceito: 2
feedback: 2
insert into mensagens_sim(codigo,evento,mensagem,codigousuario,codigoc
es(null,'RETORNOAVALINFORMAL','31|27|93|2|2','93','27');
AG.ASSISTENTE - Mensagem enviada ao CMS com sucesso.
AG.AVALIACAO - Nenhuma mensagem pendente na fila. Aguardando...
AG.ASSISTENTE - Nenhuma mensagem pendente no Blackboard do CMS. Aguard
AG.ASSISTENTE - Nao existem mensagens pendentes na fila. Aguardando...
AG.AVALIACAO - Nenhuma mensagem pendente na fila. Aguardando...
AG.ASSISTENTE - Nenhuma mensagem pendente no Blackboard do CMS. Aguard
AG.ESPECIALISTA - Nao existem mensagens pendentes na fila. Aguardando.
AG.ASSISTENTE - Nao existem mensagens pendentes na fila. Aguardando...

```

Figura 3.6 - Histórico da comunicação entre agentes no SIMEDUC

No histórico apresentado na Figura 3.6, verificamos que o agente Assistente envia uma mensagem ao agente de Avaliação contendo dados sobre o comportamento do estudante após a avaliação de um objeto de ensino. O agente de Avaliação acessa a mensagem e interpreta os dados, devolvendo ao agente Assistente o resultado obtido pelo estudante. Após o recebimento do valor da avaliação, o agente Assistente que, por sua vez, irá analisar o resultado buscando aplicar alguma reatividade a ser apresentada ao estudante, processo explicado com maiores detalhes no capítulo 4.

Todas as ações dos agentes, que implicam em enviar e receber informações, são identificadas no sistema como comportamento do agente. Os principais comportamentos dos agentes são: monitorar a sua fila de mensagem, retirar e verificar o tipo de mensagem que chegou a fila, executar as tarefas identificadas na mensagem e enviar mensagens para outros agentes. Os comportamentos dos agentes são executados continuamente, enquanto estiverem ativos no sistema.

Os comportamentos dos agentes são simples de serem atualizados, podendo ser acrescentados novos parâmetros, possibilitando uma maior interação entre os agentes, objetivando que respostas mais personalizadas possam ser geradas para os estudantes. A facilidade de manutenção também é um fator predominante na implementação dos agentes, pois as interações com outros sistemas podem ocorrer por meio de classes comuns de Java. Atualmente no SIMEDUC as mensagens entre os agentes já estão implementadas, porém as ações executadas pelos agentes Assistente e Pedagógico não utilizam a plataforma Jade, contudo, as estruturas de mensagens já foram propostas por Dorça (2004).

Capítulo 4

Integrando Planejamento e Reatividade

Como foi apresentado no capítulo anterior, o sistema SIMEDUC é composto de vários agentes, entre eles o agente Pedagógico e o agente Assistente. O primeiro tem como função a geração automática de um plano de curso e o segundo de apresentar o plano ao estudante (QUEIROZ, 2003).

A apresentação deste plano de forma linear e estática ao estudante, não acrescenta grandes novidades aos modelos de cursos baseados na Web. Uma possível alternativa nesse processo é a possibilidade de modificar a forma como o plano é apresentado, porém essa modificação deve ser feita somente se houver a necessidade de interferir no aprendizado do estudante.

Observações feitas durante a navegação do estudante no plano de curso é fator determinante para se promover alterações na apresentação do plano, identificadas como Reatividade. Para que a Reatividade ocorra, as ações do estudante são observadas pelo agente Assistente, que a partir de regras de condição/ação promoverá ajuste na apresentação do plano ao estudante, sem modificar as seqüências definidas pelo agente Pedagógico. Os dois próximos subcapítulos irão demonstrar a construção do plano e como as regras de reação podem ser utilizadas pelo agente reativo e apresenta também exemplos da implementação do agente.

4.1 Planejamento

Para um bom desempenho do agente Pedagógico, é necessário uma boa representação do conhecimento do domínio. Esta representação deve conter informações pedagógicas de cada um dos conteúdos existentes no domínio. Estas informações são utilizadas na adaptação e individualização do conteúdo do curso (QUEIROZ, 2003). A base de conhecimento desse agente é composta por duas partes:

- Base de Material Didático: Contém todo tipo de material que pode ser apresentado na Web, essa base pode estar centralizada em um servidor ou espalhada em vários servidores pela Internet.
- Base Metadata: Contém as informações necessárias para indexar os materiais didáticos, atribuindo-lhes significados e relacionando as dependências entre eles.

Para a definição da base de conhecimento foi utilizado como padrão o IEEE LTSC *Learning Object Metadata* (HODGINS, 2001). Nesse padrão, um objeto de ensino é definido como alguma entidade, digital ou não, que pode ser usada para o aprendizado, educação ou treinamento. Um objeto de ensino Metadata define o conjunto mínimo de propriedades necessárias para permitir o gerenciamento, a localização e a avaliação destes objetos (QUEIROZ et al, 2003).

O plano gerado por este agente deve oferecer ao estudante uma seqüência que viabilize o aprendizado, objetivando determinada meta instrucional, que é o conhecimento final que deverá ser assimilado. Podemos exemplificar a meta na seguinte afirmação: “o estudante tem conhecimento suficiente em Java”. Além dos conteúdos necessários, o agente deve levar em consideração o estado cogniti-

vo do estudante. A arquitetura do planejador de conteúdo e apresentada na Figura 4.1.

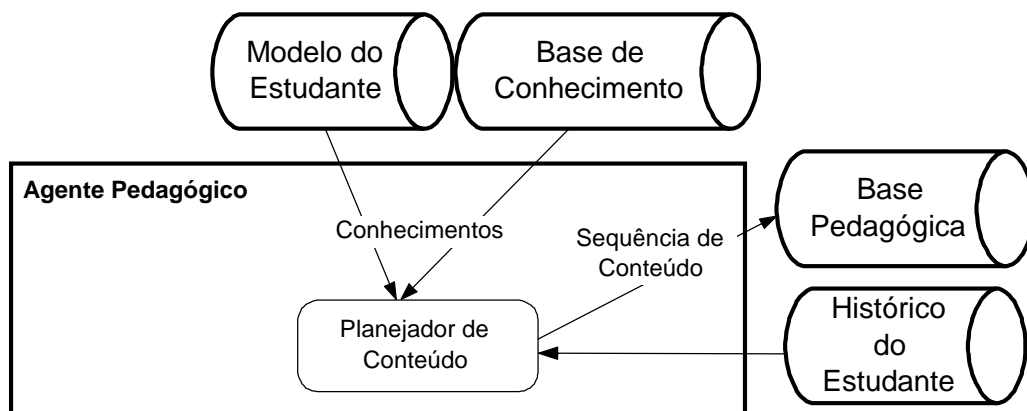


Figura 4.1 - Arquitetura do agente Pedagógico.

As bases de dados presentes nesta arquitetura formam o conjunto de informações necessárias para que o plano possa ser gerado. As informações presentes na base de conhecimento são inseridas pelos autores dos cursos, identificados como objetos instrucionais que têm como objetivo final uma meta de aprendizagem, esses objetos são inseridos conforme as especificações LOM. A base de dados modelo do estudante tem como finalidade armazenar o estado cognitivo do estudante, composto por informações que possibilitam identificar suas preferências de aprendizagem como, por exemplo, tipos de materiais mais utilizados, o quanto ele já sabe sobre certos objetos instrucionais, principais dificuldades, resultado das últimas avaliações, entre outras. No próximo capítulo, será explicado como o literal $Sabe(x,y)$ verifica o estado cognitivo do estudante.

A base de dados denominada histórico do estudante é utilizada para armazenar as ações do estudante como, por exemplo, os objetos de ensino visualizados, atividades exercidas e recursos computacionais mais utilizados durante a visualização dos cursos. As informações contidas no histórico são as primeiras a serem

consultadas pelo agente Pedagógico para que possa iniciar a geração de um plano personalizado. Após a geração, o plano é armazenado na base de dados pedagógico, utilizado também pelo agente assistente que irá apresentá-lo ao estudante.

Para gerar o plano, o agente Pedagógico se orienta por meio da execução das regras de decomposição que, por sua vez, faz uso de operadores que possuem características de identificar os conteúdos a serem selecionados para o curso, os subcapítulos subsequentes descrevem em maiores detalhes essas regras.

4.1.1 Descrição das Regras

As regras têm como finalidade selecionar os materiais instrucionais a serem utilizados e que sejam capazes de satisfazer determinadas metas instrucionais. Esses materiais, identificados como objetos de ensino, são capazes de representar o conhecimento sobre determinado tema. A seleção de um objeto é feita por dois tipos de operadores: os abstratos e os primitivos.

Os operadores abstratos buscam identificar os objetos de ensino que possam satisfazer a meta instrucional presente na pilha de metas. Estes operadores são: ensinar, ensinar novamente ou rever.

Podemos dizer que a aplicação dos operadores abstratos tem como finalidade ensinar o estudante utilizando objetos de ensino cujos conteúdos têm os mais baixos níveis de abstração do conhecimento.

Um exemplo de como o agente Pedagógico identifica o operador abstrato que irá fazer parte da pilha de operadores, pode ser dado pela seguinte seqüência: o primeiro passo é a verificação no histórico do estudante se há existência de objetos de ensino que satisfaça a meta instrucional informada, depois, identificar

qual operador será aplicado pela regra, descritos no item 4.1.2. O nível de conhecimento do objeto é identificado por meio do estado cognitivo do estudante. Após estes procedimentos o operador correspondente é aplicado, conforme é apresentado na tabela 4.1.

Tabela 4.1 - Descrição dos operadores abstratos presentes no planejamento

Operadores Abstratos	Suas funções
Aprender (ensinar)	Operador responsável pelos conceitos que são vistos pela primeira vez. É representado por $Aprender(X)$, onde X é o identificador do objeto de ensino a ser apresentado.
ReAprender (ensinar novamente)	Operador utilizado quando o estudante apresentar dificuldades em compreender o conceito apresentado. Sua aplicação é similar ao $Aprender$, ou seja, $ReAprender(X)$.
Rever (rever)	Operador que tem como finalidade oferecer uma revisão sobre o objeto de ensino, representado como $Rever(X)$

Após a inclusão dos operadores abstratos na pilha de operadores, é feita uma ligação entre esses operadores e as metas instrucionais presentes na pilha de metas, que são retiradas da pilha, pois elas foram satisfeitas pelos operadores escolhidos.

Ao verificar que as metas foram satisfeitas, o planejador monta uma seqüência inicial dos conteúdos do plano, essa primeira seqüência é composta somente de operadores abstratos, presentes na pilha de operadores, porém esse tipo de operador ainda não oferece possibilidade de se apresentar o plano, por isso eles são decompostos em operadores primitivos.

Os operadores primitivos têm como função identificar os materiais instrucionais que serão utilizados para ensinar determinado conceito, em seu menor nível de especificação. O agente Pedagógico faz uso de nove tipos de operadores primitivos, demonstrados na tabela 4.2.

Tabela 4.2 - Descrição dos operadores primitivos presentes no planejador

Operadores Primitivos	Função
Definir	Responsável por apresentar materiais instrucionais que definem um conceito.
Explicar	Responsável por apresentar materiais instrucionais que explicam um conceito.
Exemplificar	Responsável por apresentar materiais instrucionais que exemplificam um conceito.
Interpretar	Responsável por apresentar materiais instrucionais que são capazes de auxiliar o estudante à interpretar o conhecimento, já apresentado ao estudante, sobre um determinado conceito.
Concluir	Responsável por apresentar materiais instrucionais que contém um conteúdo que conclui os conhecimentos apresentados sobre um conceito.
Demonstrar	Responsável por apresentar materiais instrucionais que são capazes de fazer uma demonstração do uso dos conhecimentos.
Avaliar	Responsável por apresentar materiais instrucionais que são capazes de avaliar os conhecimentos apresentados sobre um determinado conceito.
Complementar	Responsável por apresentar materiais instrucionais que são capazes de complementar os conhecimentos apresentados sobre um determinado conceito.
MostrarLinks	Responsável por apresentar <i>links</i> de materiais instrucionais.

A decomposição dos operadores é realizada com o auxílio das informações presentes no modelo do estudante, possibilitando assim a geração de um plano personalizado conforme os conhecimentos do estudante. A recuperação desse conhecimento é realizada por meio da utilização de literais como, por exemplo, o $Sabe(X,Y)$, onde X é o identificador do objeto de ensino e Y um valor representativo do nível de conhecimento do estudante sobre o objeto, este valor varia de 0 a 1 (QUEIROZ, 2004), esses literais são representados conforme especificações da lógica de predicados de primeira ordem.

Ao verificar a existência do objeto de ensino, o agente Pedagógico busca identificar qual operador abstrato pode satisfazer o literal $Sabe(x,y)$. Os possíveis operadores que podem ser utilizados nesse processo são o $Aprender(x)$, $Reaprender(x)$ ou $Rever(x)$. Os dois últimos são utilizados somente em casos em que o

estudante já tenha visualizado o objeto em questão e que seu nível de conhecimento seja maior que zero. Para objeto de ensino ainda não visto ou com conhecimento igual a zero, o operador abstrato utilizado para tratá-lo é o $Aprender(x)$.

Para um entendimento melhor sobre o processo, será demonstrado como uma meta X é satisfeita, levando em consideração que o objeto de ensino nunca foi visto pelo estudante. O primeiro passo é identificar qual operador abstrato irá satisfazer a meta; nesse caso, o operador escolhido será o $Aprender(X)$, onde X é o objeto de ensino a ser assimilado pelo estudante.

O segundo passo é a decomposição do operador $Aprender(x)$ em operadores primitivos, como é apresentado na Figura 4.2, que possam satisfazer a meta. Se a busca for verdadeira, o operador abstrato é substituído pelos operadores primitivos, juntamente com seus objetos de ensino.

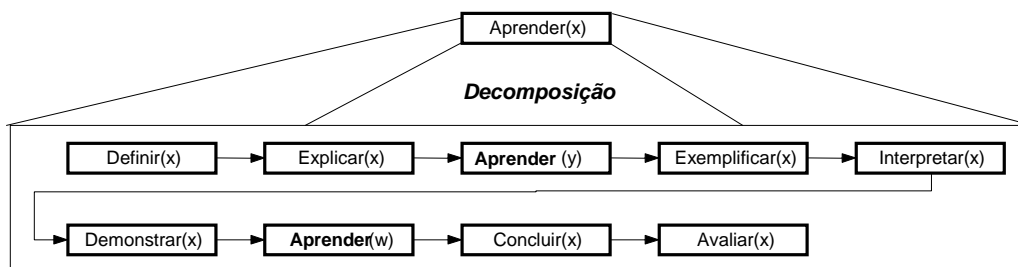


Figura 4.2 - Representação da decomposição do $Aprender(x)$.

Pode ocorrer, durante a decomposição, o surgimento de subconceitos ou objetos de ensino que ainda não foram contemplados no planejamento ou não assimilados pelo estudante. Quando isto ocorre um novo operador abstrato é inserido na pilha de operadores, juntamente com seu objeto de ensino. Podemos observar que isto ocorreu na Figura 4.2, dois objetos de ensino (y e w) ou submetas ainda não foram assimilados pelo estudante, por isso são tratados pelo operador abstrato $Aprender()$.

Para retirar os operadores abstratos que ainda persistem na pilha de operadores, o planejador volta na pilha e faz a decomposição, em separado, do novo operador abstrato e acrescenta esta nova decomposição ao plano, em substituição ao operador. Este processo se repete até que todos os operadores abstratos presentes na pilha de dados tenham sido tratados. A Figura 4.3 ilustra a seqüência dessa execução.

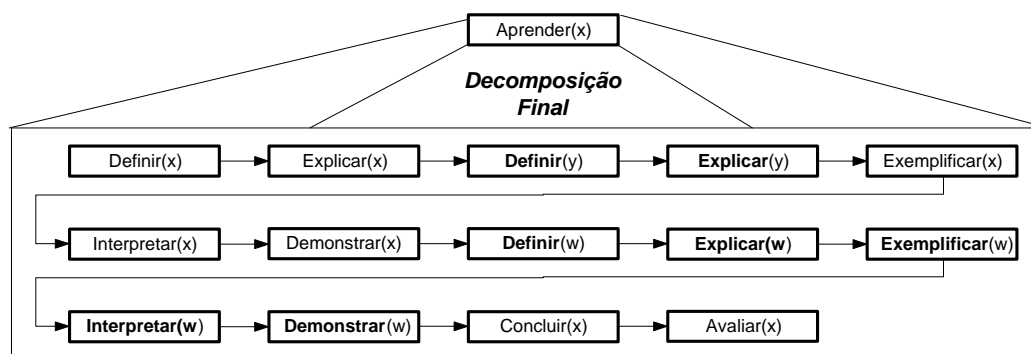


Figura 4.3 - Decomposição final do operador Aprender(x).

A decomposição final dos operadores abstratos apresenta agora um plano composto somente por operadores primitivos e, portanto, está pronto para ser apresentado ao estudante.

4.1.2 Descrição das Regras do Planejador

Como foi citado anteriormente, a decomposição é realizada por meio da aplicação de regras que satisfizeram a meta instrucional, no final desta decomposição é criada uma pilha de operadores primitivos, que é o resultado da decomposição dos operadores abstratos e tem como finalidade estabelecer a seqüência de conteúdos e ações do plano.

As regras de decomposição são divididas em três conjuntos, que são: o primeiro conjunto é composto de regras de gerenciamento responsáveis em controlar todo o processo de geração do plano. O segundo conjunto é responsável pelas re-

gras de seleção de operadores abstratos e o último conjunto de regras é responsável pela decomposição dos operadores.

4.1.2.1 Regras de Gerenciamento

O conjunto de regras de gerenciamento ou regras de geração de plano têm como finalidade coordenar todo o processo de geração das seqüências de conteúdos do plano. Esse conjunto de regras é apresentado na tabela 4.3.

Tabela 4.3 - Conjunto de regras de gerenciamento do plano

Regras de Geração de Plano – RGP	
Gerencia a montagem do plano, utilizando para isto as regras de <i>seleção de operadores</i> e de <i>decomposição de operadores</i> .	
RGP01	Se a pilha de metas e a pilha de operadores estão vazias, encerra a geração do plano.
RGP02	Se a meta ainda não foi contemplada, execute as regras de seleção de operadores (RSO) para obter o operador adequado para a meta.
RGP03	Se houver operadores abstratos no plano e a pilha de metas estiver vazia, execute as regras de decomposição de operadores (RDO).

A primeira regra (RGP01) tem como objetivo finalizar o processo de geração do plano, isso ocorre quando as pilhas de metas instrucionais e operadores abstratos estiverem vazias. A segunda regra (RGP02) tem como objetivo esvaziar a pilha de metas, para isso ela faz uso das Regras de Seleção de Operadores (RSO), que identifica os operadores abstratos que satisfazem as metas instrucionais, removendo-as assim da pilha de metas.

A terceira e última regra (RGP03) é acionada quando a pilha de metas estiver vazia e se na pilha de operadores houver operadores abstratos a serem decompostos. Para executar esse procedimento as Regras de Decomposição de Operadores (RDO) são acionadas para selecionar os operadores primitivos que serão utilizados pelo plano.

4.1.2.2 Regras de Seleção de Operadores - RSO

O conjunto de Regras de Seleção de Operadores (RSO) é responsável por procurar operadores abstratos que possam atender a meta instrucional. A tabela 4.4 mostra todas as regras que são utilizadas para esta identificação.

Tabela 4.4 - Conjunto de Regras para Seleção de Operadores Abstratos

Regra	Definição
RSO01	Tem como finalidade verificar se o operador atual atende a meta do curso, caso positivo, remove a meta e faz uma reordenação dos operadores.
RSO02a	Verifica se o conceito X não está presente na base de conhecimento do estudante e se ele nunca foi visto, por meio do operador $Sabe(X;0)$. Se o retorno for verdadeiro o conceito será tratado pelo operador abstrato $Aprender(X)$.
RSO02b	Verifica se o conceito X não está presente na base de conhecimento do estudante e se ele já visto, por meio do operador $Sabe(X;0)$. Se o retorno for verdadeiro o conceito será tratado pelo operador abstrato $ReAprender2(X)$.
RSO03a	Verifica se o valor do operador $Sabe(X,Y)$ é menor que 0.7. Se o retorno for verdadeiro o conceito será tratado pelo operador abstrato $ReAprender1(X)$.
RSO03b	Verifica se o valor do operador $Sabe(X,Y)$ é menor que 0.4. Se o retorno for verdadeiro o conceito será tratado pelo operador abstrato $ReAprender2(X)$.
RSO04a	Verifica se o valor do operador $Sabe(X,Y)$ é maior que 0.7 e se X foi visto nos últimos seis meses. Se o retorno for verdadeiro o conceito será tratado pelo operador abstrato $Resumir1(X)$.
RSO04b	Verifica se o valor do operador $Sabe(X,Y)$ é maior que 0.7 e se X foi visto nos últimos dezoito meses. Se o retorno for verdadeiro o conceito será tratado pelo operador abstrato $Resumir2(X)$.
RSO04c	Verifica se o valor do operador $Sabe(X,Y)$ é maior que 0.7 e se X foi visto a mais de dezoito meses. Se o retorno for verdadeiro o conceito será tratado pelo operador abstrato $Resumir3(X)$.

A primeira regra (RSO01) é acionada quando for identificado o operador que atenda a meta do curso. A partir daí o operador é inserido na lista de operadores e uma ligação é feita com a meta que foi atendida. Se, no final do processo, nenhum operador for encontrado, um operador abstrato adequado será inserido na pilha. A escolha desse operador leva em consideração o estado cognitivo do estudante e o seu histórico de atividades.

Para concluir a escolha do operador abstrato, o planejador compara o resultado do literal $Sabe(x,y)$ com os valores de referência que varia de 0 a 1 e o período (meses) que o conceito foi visto. Os períodos utilizados como referência foram considerados com o objetivo de definir a necessidade de uma revisão sobre o conteúdo já visto pelo estudante, essas regras vão de RSO02 a RSO04. Somente uma regra será executada durante um único o ciclo de execução.

4.1.2.3 Regras de Decomposição de Operadores – RDO

Após a criação da pilha de operadores, composta somente por operadores abstratos que satisfizeram as metas instrucionais, é necessário que os mesmos sejam decompostos em operadores primitivos. Sabemos que os operadores primitivos são aqueles que representam o conhecimento no seu menor nível de entendimento.

As regras de decomposição de operadores (RDO) são divididas em dois subconjunto, o primeiro é composto de regras de seleção de decomposição (RSD) que tem como finalidade identificar quais regras do segundo sub-conjunto serão utilizadas. Este subconjunto é composto por regras de geração da decomposição (RGD), que são os operadores primitivos a serem inseridos na pilha de operadores. Na tabela 4.5 estão representados os dois subconjuntos da RDO.

A execução das regras presentes em RSD é simples, verifica-se qual operador abstrato está no topo da pilha (Aprender, ReAprender ou Resumir), ativa a única regra válida que irá fazer a chama de RGD. Diferente das aplicações anteriores, onde somente uma regra é ativada por vez, o sub-conjunto RGD irá executar quantas regras forem necessárias para atender à decomposição.

Em uma definição mais detalhada de RGD, podemos dizer que as regras RGD01a e RGD01b são utilizadas para apresentar o material instrucional que define o objeto de ensino. A diferença entre elas é a densidade semântica utilizada na definição do objeto. Entenda-se por densidade semântica ao grau de complexidade do material instrucional, e quanto maior a densidade, maior é o nível de detalhamento e de informações do material. O mesmo ocorre para as regras RGD02a, RGD02b (explicar), RGD06a, RGD06b (interpretar), RGD07a, RGD07b (concluir), RGD08a, RGD08b (demonstrar) e RGD10a, RGD10b (complementar), mudando somente os tipos de operador presentes nelas.

A regra RGD03 tem a função de identificar se o objeto de ensino possui um sub-conceito, no caso em que isto ocorre a regra faz uma chamada RSO para que o operador abstrato correspondente possa ser identificado. O mesmo ocorre para a regra RGD05, porém esta verifica se o objeto de ensino possui alguma especialização a ser apresentada.

As regras RGD04a, RGD04b e RGD04c são responsáveis pela identificação da quantidade de objetos de ensino que podem exemplificar o conceito atual, sendo possível encontrar até três exemplos por objeto. Além de possuir um operador que pode exemplificar, há também a regra RGD11 que tem como finalidade buscar um objeto que possa apresentar resumo do conteúdo atual. E, por fim, a regra RGD09 que tem como característica apresentar o objeto de ensino e como objetivo avaliar o conteúdo já apresentado.

Tabela 4.5 - Conjunto de regras para decomposição de operadores Abstratos

Regras de Decomposição de Operadores – RDO			
Sub-regras p/ selecionar a decomposição		Sub-regras p/ gerar a decomposição	
Regras	Operador Abstrato	Regras	Operador Primitivo
RSD01	Para os operadores abstratos do tipo <i>Aprender</i> , as sub-regras a serem utilizadas são: RGD01a, RGD02a, RGD03a, RGD04a, RGD05, RGD06a, RGD07a, RGD08a e RGD09.	RGD01a	Definir(objeto) com densidade semântica normal.
		RGD01b	Definir(objeto) com densidade semântica elevada.
		RGD02a	Explicar(objeto) com densidade semântica normal.
		RGD02b	Explicar(objeto) com densidade semântica elevada.
RSD02a	Para os operadores abstratos do tipo <i>ReAprender1</i> as sub-regras a serem utilizadas são: RGD01a, RGD02b, RGD03, RGD04b, RGD05, RGD06a, RGD07a, RGD08a, RGD09 e RGD10a.	RGD03	Se for sub-conceito do objeto, executar as regras RSO.
		RGD04a	Exemplificar(objeto) uma única vez.
		RGD04b	Exemplificar(objeto) duas vezes, de forma diferente.
RSD02b	Para os operadores abstratos do tipo <i>ReAprender2</i> , as sub-regras a serem utilizadas são: RGD01b, RGD02b, RGD03, RGD04c, RGD05, RGD06b, RGD07b, RGD08b, RGD09 e RGD10b.	RGD04c	Exemplificar(objeto) três vezes, de forma diferente.
		RGD05	Se houver especificação do objeto, executar as regras RSO.
		RGD06a	Interpretar(objeto) com densidade semântica normal.
RSD03a	Para os operadores abstratos do tipo <i>Resumir1</i> , utilize o operador primitivo <i>MostrarLink()</i> .	RGD06b	Interpretar(objeto) com densidade semântica elevada.
		RGD07a	Concluir(objeto) com densidade semântica normal.
		RGD07b	Concluir(objeto) com densidade semântica elevada.
RSD03b	Para os operadores abstratos do tipo <i>Resumir2</i> , utilize a sub-regra RGD11.	RGD08a	Demonstrar(objeto) com densidade semântica normal.
		RGD08b	Demonstrar(objeto) com densidade semântica elevada.
		RGD09	Avaliar(objeto)
RSD03c	Para os operadores abstratos do tipo <i>Resumir3</i> , as sub-regras a serem utilizadas são: RGD11, RGD02a e RGD04a.	RGD10a	Complementar(objeto) com densidade semântica normal.
		RGD10b	Complementar(objeto) com densidade semântica elevada.
		RGD11	Resumir(objeto)

4.1.3 Algoritmo do Planejador

Todos os processos descritos sobre a execução do agente Pedagógico são representados no algoritmo da Figura 4.4. Para que o algoritmo seja executado corretamente é necessário informar dois valores de entrada: a meta instrucional e o estudante. O valor de entrada correspondente à meta instrucional será utilizada pelo

algoritmo para estabelecer qual é a meta a ser alcançada pelo plano a ser gerado e a entrada referente ao estudante será utilizada para buscar informações sobre o estado cognitivo e o histórico do estudante.

O primeiro procedimento a ser executado pelo algoritmo é a geração de um plano inicial, linha 1, composto pelo estado cognitivo do estudante e as metas instrucionais a serem atendidas. Logo após, na linha 2, uma estrutura de repetição é iniciada, tendo como valores de referência as pilhas de operadores e de metas. Enquanto a pilha de metas não estiver vazia e a pilha de operadores não estiver livre de operadores abstratos a geração do plano não é terminada. A regra que verifica as pilhas é a RGP01.

```

Planejador(metaInstrucional, estudante)
{
1 plano = montaPlanoInicial(metaInstrucional, estudante);
2 Enquanto(pilhaOperadores!=vazio e pilhaMetas!=vazio) {

3   se (pilhaMetas != vazio) {
4     subMeta = selecionaMetaNTratada(pilhaMetas)
5     operador = selecionaOperador(subMeta);
6     insereOperador(operador);
7   }else {
8     opAbstrato=selecionaOperadorAbstrato(pilhaOperadores);
9     planoDecomposicao = Decompor(opAbstrato);
10    concatena(plano, planoDecomposicao);
11  }
12 }
13 finaliza();
}

```

Figura 4.4 - Algoritmo utilizado pelo planejador de conteúdo

Os códigos compreendidos entre as linhas 3 e 6 são controlados pela regra RGP02 que tem como finalidade identificar e selecionar quais operadores abstratos satisfazem as metas instrucionais presentes na pilha de metas. Esta regra só é executada caso a pilha de metas não estiver vazia. Se a pilha de metas estiver vazia, o algoritmo irá decompor os operadores abstratos que satisfizeram as metas, linhas 8, 9 e 10, por meio da aplicação da regra RGP03. A cada operador decom-

posto, o algoritmo reorganiza o plano gerado, realimentando a pilha de operadores com os operadores primitivos encontrados.

Com a ativação das regras RGP01, RGP02 e RGP03 todas as demais regras apresentadas podem ser acionadas, isso dependerá dos valores encontrados pelas regras de geração do plano.

4.1.4 O Plano Gerado

Assim que o planejador encerrar a geração do plano, já é possível apresentá-lo ao estudante. Na Figura 4.5, apresentamos o exemplo de um plano gerado, utilizando como meta instrucional o ensino do conceito Java. Foi considerado que o estado cognitivo e o histórico do estudante era vazio, ou seja, não havia informações que pudessem modificar a forma como o plano fosse gerado.

Ao final da geração do plano, todos os operadores primitivos necessários para apresentar o curso estão armazenados em uma seqüência lógica. Para entender melhor a leitura do plano, observe o texto abaixo que descreve o plano de ensino do conceito Java, de forma resumida.

“Os objetos de ensino html, poo, servlet, *herança* e *polimorfismo* são apresentados por quase todos os operadores presentes em RDO, em uma seqüência normal, já o objeto Java recebe uma pausa em sua apresentação. A partir do operador complementa(Java) até o exemplifica(Java) são apresentados objetos em nível de especificação do Java, depois desta especificação o plano retoma a apresentação do objeto Java já entrando nas fases de e-

xemplificação, interpretação, conclusão, demonstração e avaliação.”

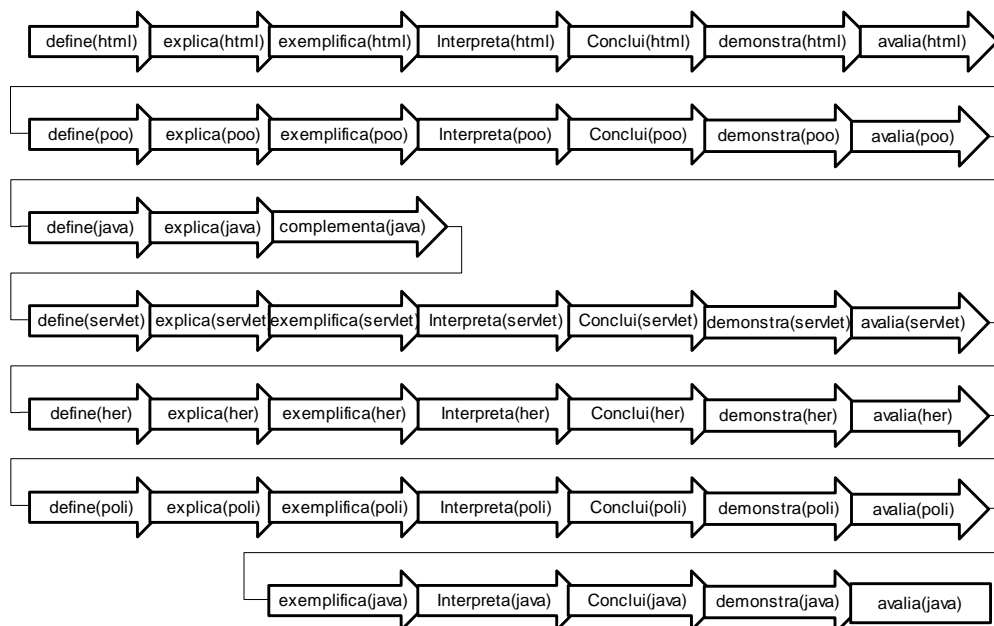


Figura 4.5 - Sequência de um plano de curso gerado pelo planejador

Mesmo disponibilizando o curso seqüencialmente organizado, o sistema SIMEDUC apresenta uma grande evolução no que diz respeito ao atendimento personalizado do estudante, pois a geração dos planos é individualizada para cada estudante, ou seja, conforme a meta do curso e os conhecimentos prévios do estudante um plano diferente é gerado. A Figura 4.6 apresenta uma tela do protótipo, onde é listado parte do plano gerado pelo planejador, contendo o operador primitivo e o conteúdo do objeto de ensino. No protótipo ainda não foram incluídos os conteúdos descritivos dos objetos de ensino, somente o título do objeto, os relacionamentos essenciais e a meta do curso.

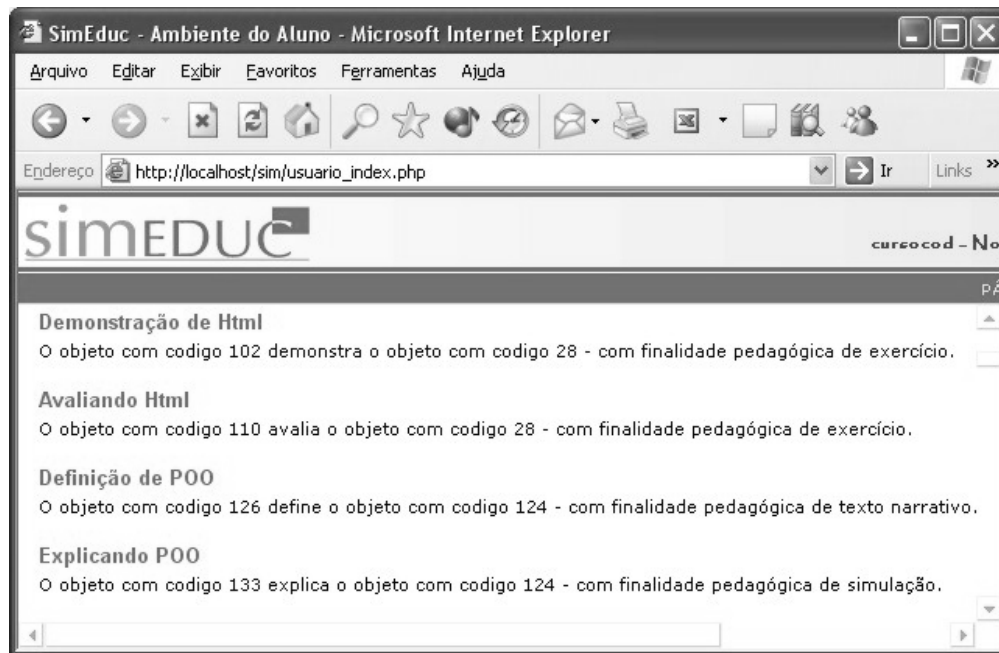


Figura 4.6 - Tela com a seqüência de conteúdos do curso a ser estudado

4.2 Agente Assistente Reativo

Após a geração do plano de curso, o mesmo estará disponível para ser apresentado ao estudante, esta tarefa é realizada pelo agente Assistente, foco deste trabalho. Além de apresentar o plano, o agente deverá ser capaz de identificar dificuldades e promover alterações de como o curso será mostrado, com o intuito de estimular a continuidade do estudante, auxiliar na superação de suas limitações e, para aqueles estudantes que apresentarem um desempenho maior do que esperado, avançar no curso.

Verificar como o estudante está evoluindo no processo de ensino/aprendizagem via Web não é uma tarefa fácil, pois envolve vários fatores como, por exemplo, limitação nos recursos computacionais para identificar com precisão as dificuldades de interpretação, o baixo desempenho nas avaliações, entre outros. Por outro lado, há aqueles estudantes que apresentam alto desempenho e o processo normal do curso pode deixá-los desmotivados a continuar no curso. Um

dos recursos mais utilizados para obter informações sobre o desempenho do estudante é a observação do seu desenvolvimento no decorrer do curso. Esse processo oferece informações que são analisadas e tem como intuito identificar suas possíveis dificuldades e/ou sucessos durante a aprendizagem (FERREIRA & LOPES, 2004).

Para Vassileva (1995) as informações observadas durante a navegação do estudante pode identificar fatores que são *externos ao processo* de ensino e possíveis de serem identificados. Para monitorar estes fatores, Vassileva (1995) propõe a criação de operadores que possam interagir com o estudante, promovendo assim ajustes no plano em tempo de execução, possibilitando a personalização do plano. Segundo Drummond (1989) a hipótese de que planos são conjuntos totais ou parcialmente ordenados de operadores é muito limitador: os planos devem informar ações e não serem responsáveis por defini-las. Drummond entende que um plano não deve ser visto como uma seqüência e sim como uma especificação de reações. É com esta visão que o agente Assistente foi projetado e utiliza o plano gerado para dirigir as reações no sentido de fazer com que a meta do curso seja satisfeita. Cabe ao agente Pedagógico a geração deste plano, conforme mencionado no subcapítulo 4.1, que será utilizado pelo agente Assistente.

Podemos notar que a arquitetura do agente Assistente apresentada na Figura 4.7, compartilha as bases de dados presentes no agente Pedagógico, são eles: modelo do estudante, histórico do estudante e base pedagógica, sendo que as bases de dados do estudante são atualizadas pelo agente Assistente. Este compartilhamento tem como objetivo centralizar as informações obtidas por meio da navegação do estudante, possibilitando assim um processo de personalização mais eficiente. A

base de dados “regras de reação” tem o papel de armazenar as regras de reação que podem ser definidas pelo professor/autor ou instanciadas na criação do agente. Outra funcionalidade apresentada na arquitetura é a interação direta com o estudante, na qual há troca de informações, o agente apresenta o plano e o usuário fornece dados por meio da navegação. Os dois algoritmos presentes na arquitetura são explicados com maiores detalhes nos próximos subcapítulos.

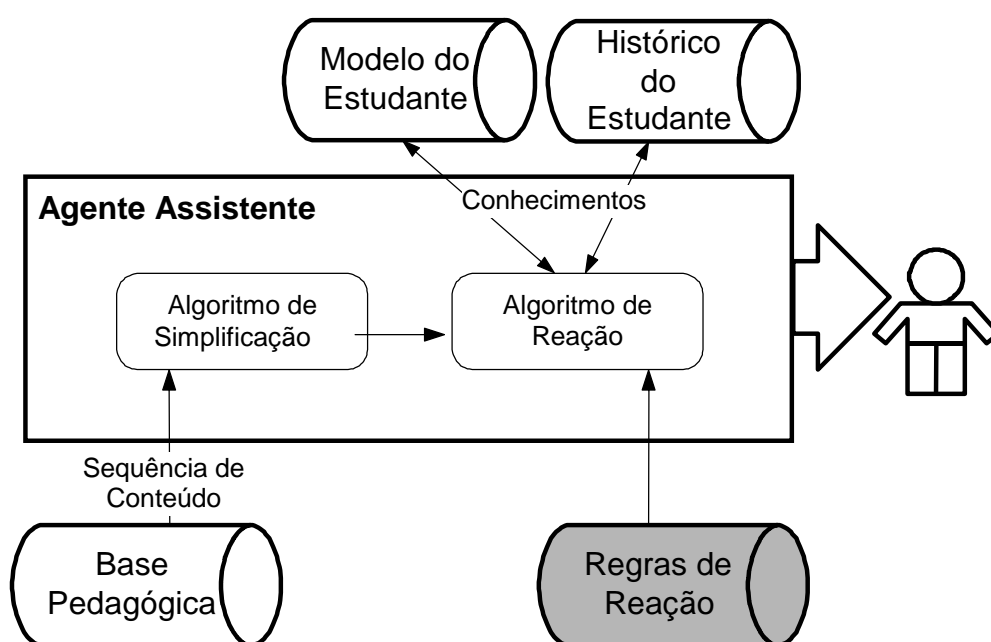


Figura 4.7 - Arquitetura do Agente Assistente Reativo

4.2.1 Reatividade no Plano

O plano de curso gerado pelo planejador apresenta repetições sistemáticas de operadores e objetos de ensino, como é apresentado na Figura 4.5, por exemplo: o operador define() aparece seis vezes e o conceito Java oito vezes cada. Esta repetição ocorre de forma natural durante a geração do plano, mas não é aconselhável ou até mesmo possível trabalhar com reatividade em estruturas que apresentam

várias opções de ação, por oferecerem varias alternativas de ações, dificultando assim o gerenciamento das ações aplicadas.

Para que o agente Assistente possa promover a reação é necessário que em sua estrutura não ocorram as repetições de objetos e operadores apresentadas no plano gerado pelo agente Pedagógico. Esse problema foi resolvido por meio da reestruturação do plano, criando duas novas estruturas: a pilha de objetos de ensino, na qual os objetos de ensino são representados sem suas repetições e a pilha de operadores que armazena os operadores utilizados na geração do plano. Durante a reorganização a ordem em que os objetos e os operadores aparecem é respeitada. A tabela 4.6 apresenta estas duas novas estruturas, tendo como origem o plano apresentado na Figura 4.5.

Tabela 4.6 - Simplificação do plano de curso gerado

Plano gerado na Figura 1	Pilha de Objetos de ensino	Pilha de Operadores
Define(Var/op)	(Var/op)	Define()
Explica(Var/op)	(poo)	Explica()
Exemplifica(Var/op)	(java)	Exemplifica()
Interpreta(Var/op)	(servlet)	Interpreta()
Conclui(Var/op)	(herança)	Conclui()
Demonstra(Var/op)	(polimorfismo)	Demonstra()
Avalia(Var/op)		Complementa()
Define(poo)		Avalia()
...		
Conclui(java)		
Demonstra(java)		
Avalia(java)		

O agente Assistente tem como estratégia aplicar todos os operadores, presentes na pilha de operadores, sobre a pilha de objetos, porém somente um operador será ativado por vez. O controle da aplicação ou não do operador sobre um determinado objeto se dá pela presença de uma marcação no objeto, isso se faz necessário para memorizar quais operadores já trataram o objeto. A marcação é

adicionada no momento em que o objeto é apresentado ao estudante. Para controlar as marcações dos objetos foi criada uma lista auxiliar simples, que não exige nenhum controle de organização. A Figura 4.8 apresenta a estratégia que possibilita esta reatividade que consiste em retirar o objeto a ser apresentado ao estudante, linha 3 da Figura 4.8, e a partir deste ponto testar qual regra será ativada, lembrando que somente uma única regra poderá ser satisfeita a cada ciclo de execução.

O primeiro teste, linha 4, verifica se o estado do estudante é vazio, ou seja, o curso ainda não foi iniciado sendo a primeira vez que informações são recebidas do estudante, por isso não será verificado a existência do objeto e nem a presença do operador que o trata, as ações **aplica** e **atualiza** serão imediatamente aplicadas. A ação **aplica** tem como função apresentar o objeto de ensino por meio do operador que está livre de marcação, após a apresentação o objeto recebe a marcação de que já foi tratado por esse operador, possibilitando que outros operadores que da aplicação prévia do operador sejam liberados. Outra ação presente é a **atualiza** que modifica o modelo do estudante memorizando quais objetos já foram visto por ele. Estes dados são armazenados na base de dados histórico do estudante, apresentado na arquitetura do agente.

Os testes realizados da linha 7 até a linha 22 têm como função aplicar os operadores livres do objeto de ensino, entenda por operador livre aquele operador que tiver seus pré-requisitos satisfeitos, ou seja, todos os operadores que os antecederem já foram apresentados, se isso for verdade a ação **aplica** será executada com o operador atual e o estado do estudante será atualizado. Ao executarmos todos os passos da estratégia sem aplicar qualquer reatividade e obtermos o esvazi-

amento da pilha de objetos, linha 30 da Figura 4.8, o curso será apresentado normalmente, conforme foi gerado.

O que promove a diferença entre as duas formas de apresentar o plano é a capacidade do agente Assistente de re-apresentar ou omitir a apresentação de certos objetos, com a finalidade de promover o reforço na aprendizagem ou avançar no curso para os estudantes com nível de conhecimento elevado. Estas reações ocorrem simplesmente modificando a marcação dos operadores nos objetos, como é mostrado nas linhas 27 e 29 da Figura 4.8 por meio da função **modifica** **Marcações** que consulta na base de dados “regras de reação” os parâmetros a serem utilizados no objeto de ensino atual.

O exemplo de reatividade utilizado na estratégia verifica se o aluno foi avaliado, se todos os operadores foram aplicados, linha 25, e qual foi o resultado da avaliação, linhas 26 e 28. Nesse caso, as avaliações que apresentarem baixo desempenho irão ativar as regras responsáveis por rever o objeto, linha 27, e para as avaliações que apresentarem resultados acima do esperado irão promover um avanço no plano, linha 29.

O professor/autor poderá informar os parâmetros a serem utilizados pelo agente reativo, que passa a desconsiderar os valores internos de referência, entre as possíveis ações de reatividade aplicadas a um plano de ensino, que podem ser orientadas pelo professor, há aquelas que têm como finalidade a revisão do objeto, sugestões de materiais de apoio ou de avançar no plano. A tabela 4.7 apresenta a inserção de parâmetros que deverão ser aplicados em avaliações de baixo desempenho, neste caso somente os operadores definir, interpretar e concluir deverão ser reapresentados. É possível utilizar qualquer um dos operadores no processo, desde

que eles estejam presentes no plano de ensino gerado e armazenados no banco de dados *regras de reação*, apresentado na arquitetura da Figura 4.7.

```

0 estadoEstudante = vazio;
1 assistente(objetos,operadores)
2 {
3 objetoAtual = retiraObjeto(objetos)
4 se estadoEstudante == vazio
5   aplica(objetoAtual,'operador');
6   estadoEstudante = atualiza(objetoAtual)

7 se objetoAtual presente no estadoEstudante e operador == ' '
8   aplica(objetoAtual,'def');
9   estadoEstudante = atualiza(objetoAtual)

10 se objetoAtual presente no estadoEstudante e operador == 'def'
11   aplica(objetoAtual,'exp');
12   estadoEstudante = atualiza(objetoAtual)

13 se objetoAtual presente no estadoEstudante e operador == 'def;exp'
14   aplica(objetoAtual,'exe');
15   estadoEstudante = atualiza(objetoAtual)

16 se objetoAtual presente no estadoEstudante e operador == 'def;exp;exe'
17   aplica(objetoAtual,'int');
18   estadoEstudante = atualiza(objetoAtual)

19 se objetoAtual presente no estadoEstudante e operador == 'def;exp;exe;int'
20   aplica(objetoAtual,'ava');
21   estadoEstudante = atualiza(objetoAtual)

22 se objetoAtual presente no estadoEstudante e operador == 'todos'
23   objetoAtual=retiraObjeto(objetos);
24   estadoEstudante = atualiza(objetoAtual)

25 se objetoAtual == avaliaçãoEstudante e operador == 'todos'
26   se resultadoAvaliação == 'abaixo do esperado'
27     estadoEstudante = modificaMarcações(revisar);
28   se resultadoAvaliação == 'acima do esperado'
29     estadoEstudante = modificaMarcações(avançar);

30 se objetos = vazio
31   fim execução;

```

Figura 4.8 - Estratégia para o Agente Assistente

A partir do resultado da avaliação, o professor/autor poderá indicar uma série de ações prioritárias do agente Assistente. Na proposta atual as ações disponíveis são: reaplicar operadores para promover o reforço de aprendizagem ou antecipar a aplicação de operadores para promover um avanço na apresentação do plano. Estas regras juntamente com mensagens de alertas propostas por Musa (2001), podem ser combinadas formando uma rede de plano (DRUMMOND, 1989), que se assemelha com uma rede de Petri, a ser executada.

Tabela 4.7 - Parâmetros para o Operador Avalia

Resultado da Avaliação	Operadores de reação
Baixo Desempenho	Definição Interpretação Conclusão
Tipo de reação	Reação
Revisão do objeto	Reapresentar o objeto de ensino utilizando somente os operadores indicados

Nos testes atuais, consideramos a pontuação da avaliação o fator principal da reação, levando em conta o modelo Fuzzy de avaliação. Os valores das faixas de reação foram definidos a partir de dados empíricos, sendo que posteriormente poderão ser ajustados. A tabela 4.8 descreve de forma resumida as possíveis formas de reações utilizando a avaliação como referência.

Tabela 4.8 - Reações previstas pelo Agente Assistente

Reações sobre o operador Avalia(objeto)		
Condição para a ação	Ação	Descrição
Se <i>avaliação</i> menor que 0.3	Remover as marcações exp, exe, int e ava do objeto	O objeto de ensino passa a estar livre para novamente ser explicado, exemplificado, interpretado e avaliado, podendo utilizar os mesmos conteúdos ou buscar novos conteúdos apontados pelo professor/autor.
Se <i>avaliação</i> maior ou igual 0.3 e menor que 0.5	Remover as marcações exe, int e ava do objeto	A reação será executada pelos operadores de exemplificação, interpretação e avaliação.
Se <i>avaliação</i> maior ou igual 0.5 e menor que 0.7	Remover as marcações exe e ava do objeto	Os operadores exemplificar e avaliar estarão disponíveis para a reação.
Se <i>avaliação</i> maior ou igual 0.7 e menor que 0.9	Não execute nada	Segue o plano normalmente.
Se <i>avaliação</i> maior ou igual 0.9	Atualiza as marcações def, exe e int no próximo objeto.	Após buscar o próximo objeto ele é atualizado com as marcações dos operadores definir, exemplificar e interpretar. Com estas marcações presentes no objeto estes operadores não serão executados.

As alterações na apresentação do plano de curso utilizando o agente Assistente, no sistema SIMEDUC, simulando manualmente os valores das avaliações, apresentaram resultados satisfatórios, demonstrados no subcapítulo 4.3 que trata da implementação. Os objetos que apresentaram baixos índices na avaliação fo-

ram re-apresentados ao estudante. Para avaliações que apresentaram índices considerados satisfatórios continuou-se a apresentação do plano em sua seqüência normal e para objetos que apresentaram avaliações acima do limite estabelecido a reação proporcionou ao estudante um avanço no curso, buscando evitar a utilização de operadores que tratam de conceitos considerados básicos para o nível do estudante, identificado pela avaliação.

4.3 Detalhes de Implementação

O SIMEDUC é composto por várias aplicações que podem ser executadas como aplicações Web, e portanto, suas implementações devem utilizar linguagens destinadas para este fim. O agente Assistente foi implementado em PHP (CONVERSE, 2003) e Java (BIGUS & BIGUS; 2002), linguagens que possibilitam criar interações dinâmicas via Web. As bases de dados utilizadas para o armazenamento do plano de ensino, como também suas reações, foram implementadas em MySQL (SUEHRING; 2002).

A implementação do agente foi dividida em três módulos, o primeiro consiste em reorganizar o plano de curso gerado pelo agente Pedagógico, tarefa que exige um processamento maior com banco de dados, implementado em Java, especificamente com aplicações Servlets/MySQL. O segundo módulo, considerado o mais simples, tem como função receber do professor/autor os parâmetros a serem aplicados ao objeto de ensino e armazená-los. O terceiro e último módulo consiste em apresentar o plano de curso ao estudante, esperando promover alguma reação, os dois últimos módulos foram implementados em PHP/MySQL. Essas implementações são mais detalhadas nos próximos subcapítulos.

4.3.1 Reorganizando o Plano para a Reatividade

O plano de curso gerado pelo agente Pedagógico apresenta repetições sistemáticas de objetos de ensino e operadores, como foi mencionado no subcapítulo 4.2. Para remover estas repetições e, ao mesmo tempo, garantir que estas modificações não alterarem a seqüência estabelecida pelo agente, o agente Assistente cria uma nova estrutura de dados que será utilizada para armazenar o plano, removendo os objetos de ensino que se repetem. Essa nova estrutura é formada por duas bases de dados, sendo a primeira destinada a armazenar os objetos de ensino e suas marcações e a segunda para os operadores.

O processo de remoção das repetições tem início quando o agente Pedagógico tiver encerrado a geração do plano e o estudante inicia-lo pela primeira vez. A Figura 4.9 apresenta a tela do SIMEDUC na qual são listados os cursos gerados para o estudante. Caso o estudante escolha um curso que ele ainda não tenha iniciado, o agente Assistente irá gerar um plano que corresponda ao plano original. Na implementação foi utilizada duas estruturas de repetições, apresentadas na Figura 4.10, a primeira repetição tem como objetivo garantir que todos os objetos do plano original sejam tratados. Ao receber o objeto de ensino o agente armazena-o na tabela *objetosreativos* e abre a segunda estrutura de repetição com o objetivo de remover as repetições dos objetos. Dentro do laço de repetição é armazenado na tabela *passosreativos* os operadores selecionados para apresentar o objeto de ensino atual.

A tabela *passosreativos* é utilizada para armazenar os operadores primitivos do plano e suas marcações, utilizadas para verificar se o operador já foi ativado ou não, conforme apresentado no subcapítulo 4.2.1.

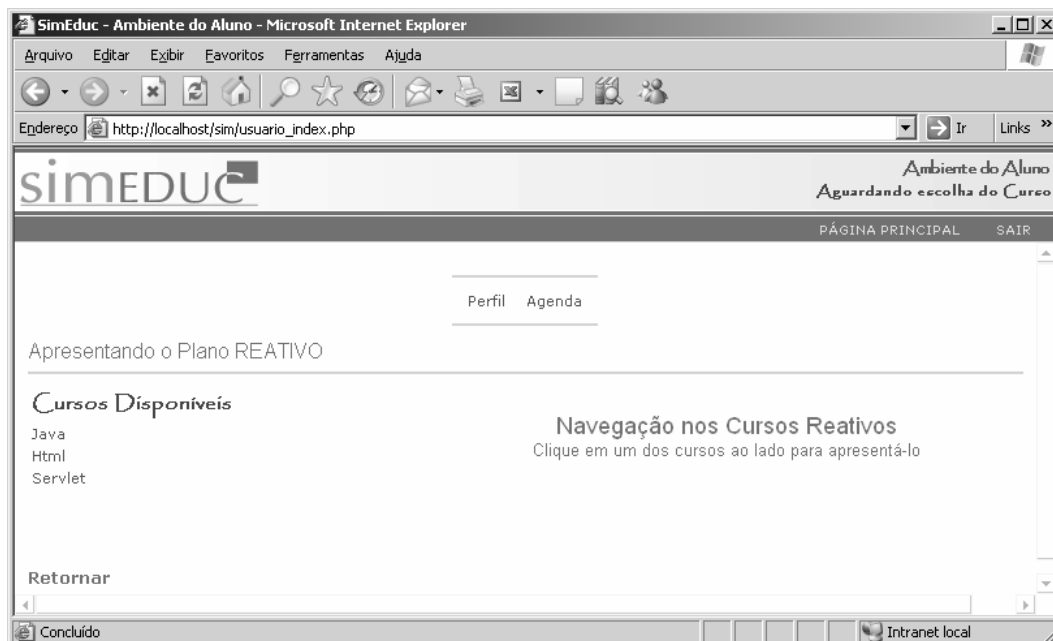


Figura 4.9 - Tela do menu dos cursos disponíveis para o estudante

```

while (plano_encontrado == plano_atual){
    objeto_encontrado = RSPassos.getInt("codigoObjetoConceito");
    stmtloop.executeUpdate("INSERT INTO objetosreativos SET
        codigoPlano="+plano_encontrado+",
        codigoObjetoConceito="+objeto_encontrado);
    . . .
    while (objeto_encontrado == objeto_atual){
        stmt.executeUpdate("INSERT INTO passosreativos SET
            codigoPlano="+plano_encontrado+",
            codigoObjetoConceito="+objeto_encontrado+",
            codigoObjeto="+objeto_operador+",
            operador = "+operador_encontrado+", status=0");
        . . .
    }
}

```

Figura 4.10 - Parte do código fonte que possibilita a remoção das repetições

Assim que o agente verificar que o plano resumido foi totalmente gerado, o curso passa a estar disponível e aguardando a navegação do estudante com a finalidade de receber informações que possam promover alguma reatividade.

4.3.2 Inserindo Parâmetros para a Reatividade

Com o objetivo de oferecer um sistema que possa promover reações de várias formas possíveis, foi implementada uma rotina para receber do professor/autor os

parâmetros de reação dos objetos. Como citado no subcapítulo 4.2.1, as reações serão ativadas conforme o retorno do operador avalia().

A Figura 4.11 apresenta a tela do SIMEDUC destinada ao professor/autor, que mostra quais cursos estão disponíveis e a opção para inserir os parâmetros de reatividade, em destaque na figura.

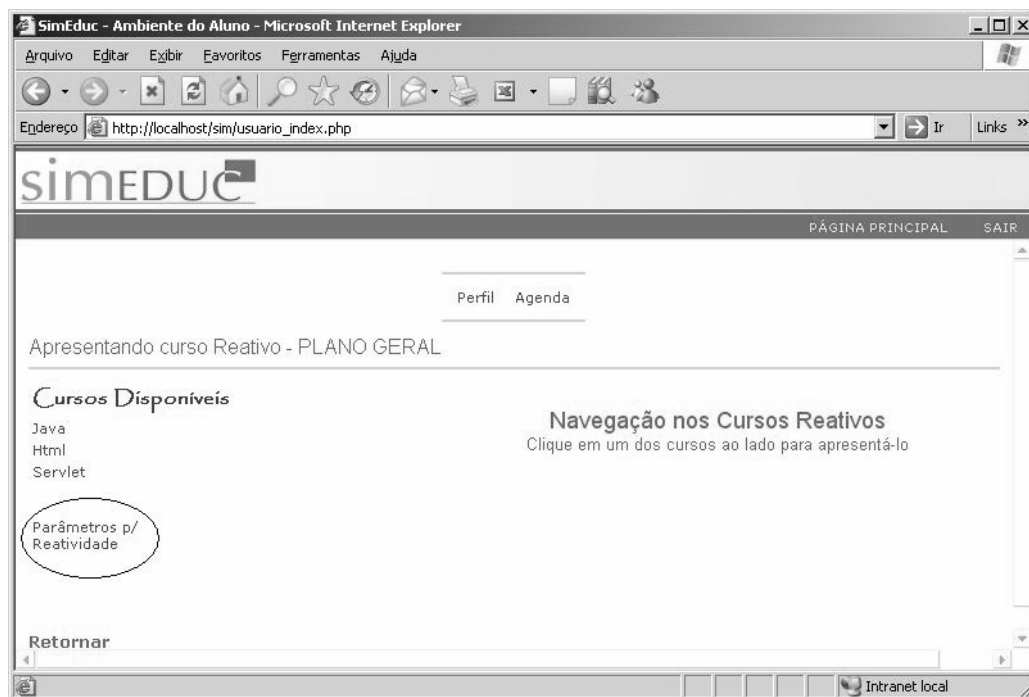


Figura 4.11 - Tela inicial da apresentação dos cursos para o professor/autor

Assim que o professor/autor escolher a opção para definir os parâmetros, será apresentado a ele uma segunda janela, Figura 4.12, na qual são solicitadas as seguintes informações:

- **Plano de curso:** ele deverá informar qual curso deseja estabelecer as regras de reação. Todos os objetos desse curso serão tratados pelas regras aqui estabelecidas.
- **Tipo de avaliação:** para qual nível do resultado da avaliação as regras deverão ser aplicadas. Os valores que definem os níveis das

avaliações foram definidos de modo intuitivo sendo que posteriormente poderão ser ajustados. Nessa implementação são considerados três níveis: baixo desempenho, satisfatória e acima do esperado.

- **Abrangência da reação:** aqui deverá ser informado quais objetos de ensino serão tratados pela reatividade. Um plano instrucional é composto por vários objetos de ensino, quando ocorre a avaliação ela retorna o resultado somente para o objeto avaliado, se o professor/autor considerar necessário ele poderá sugerir que a reação abranja um número maior de objetos. Abrangência pode ocorrer em um objeto ou em todos os objetos do plano.
- **Número de operadores:** esta opção define quais operadores deverão ser aplicados ou desabilitados durante a reação. É por meio dessa informação que as marcações dos operadores primitivos são modificadas.

No exemplo apresentado na Figura 4.12 o professor/autor está configurando o curso Java, que será revisado caso a avaliação de baixo desempenho seja identificada e esta revisão ocorrerá nos dois próximos objetos, nesse caso os dois anteriores, e por meio da alteração das marcações de três operadores, que não serão mais utilizados pelos objetos.

Após definir os parâmetros iniciais o professor/autor deverá escolher quais serão os operadores responsáveis pela reatividade, ao clicar no botão correspondente, uma nova janela será apresentada trazendo as definições estabelecidas anteriormente e solicitando a marcação dos respectivos operadores, Figura 4.13. Nesse

exemplo os operadores a serem utilizados são: define, interpreta e demonstra. A qualquer momento pode-se alterar esses parâmetros ou removê-los.

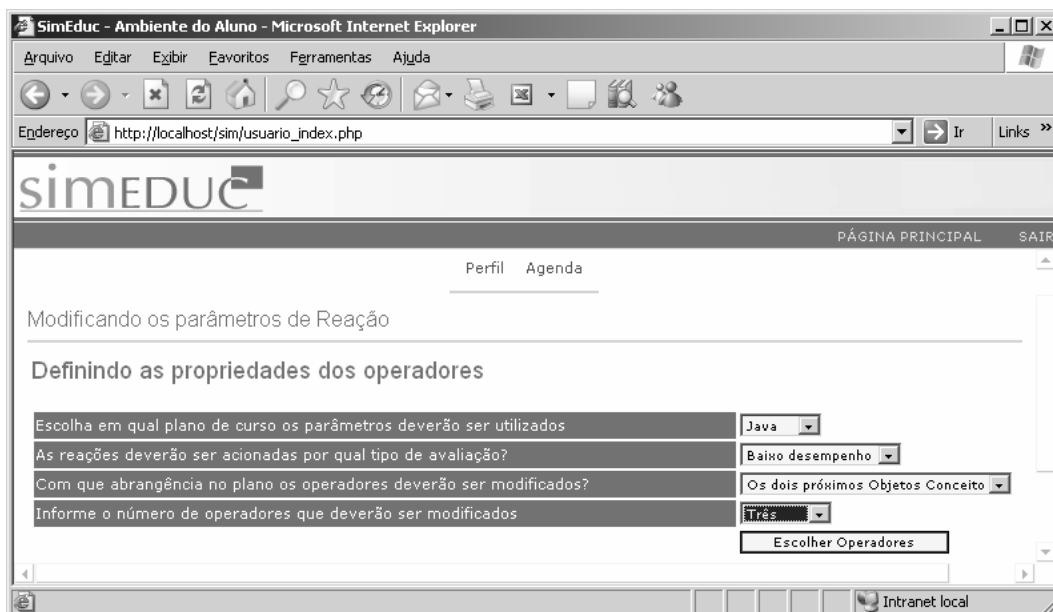


Figura 4.12 - Definição inicial dos parâmetros para promover a reatividade

Os parâmetros de reação são armazenados na base de dados *regras de reação* apresentada na arquitetura do agente. Na implementação foram utilizadas duas tabelas para realizar este processo.



Figura 4.13 - Tela que define quais operadores deverão ser utilizados na reação.

4.3.3 Apresentação Reativa do Plano

Com o plano reorganizado e os parâmetros de reação armazenados, o agente Assistente já possui todas as informações necessárias para apresentar o plano de curso reativo. Todo esse processo é transparente ao estudante, ou seja, não é perceptível para quem está navegando que há um controle ou ajustes sendo feitos em tempo de visualização do curso.

A estratégia de personalização do curso tem início a partir do momento em que o estudante entra no sistema SIMEDUC. Após a confirmação do login o agente verifica quais os planos de cursos foram gerados para esse estudante, Figura 4.9, disponibilizando-os para navegação.

Conforme foi apresentado na estratégia de personalização no subcapítulo 4.2.1, o início da apresentação do plano consiste em verificar o modelo do estu-

dante, caso seja a primeira vez que o curso esteja sendo visualizado, modelo vazio, o sistema apresenta o curso a partir de seu início. Se o agente verificar os objetos de ensino que já foram apresentados ele deverá continuar a apresentação a partir do último objeto apresentado.

Para exemplificar a aplicação da estratégia de personalização proposta, utilizamos como exemplo a visualização do curso Elementos Básicos da programação Orientada a Objetos - POO. A Figura 4.14 traz uma seqüência parcial desse plano de curso, na qual os objetos de ensino estão sendo apresentados pelos operadores definir, explicar, interpretar, concluir, demonstrar e avaliar. A apresentação desses operadores consiste em aplicar as regras da estratégia, na qual os operadores somente serão apresentados se as suas condições prévias forem satisfeitas e, portanto, disponíveis para serem apresentados, linhas de 5 à 24 da Figura 4.8.

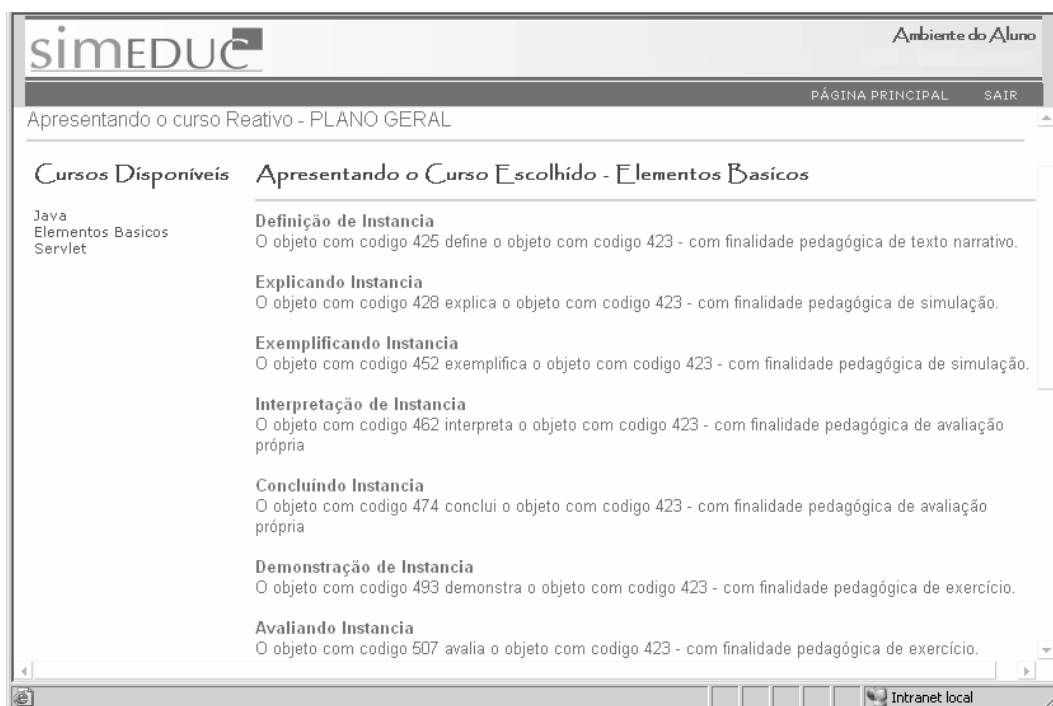


Figura 4.14 - Apresentação parcial do curso Elementos Básicos

Ao continuar a apresentação do curso, o operador **avalia()**, em algum momento, será aplicado e o seu resultado será verificado pelo agente Assistente com o intuito de aplicar alguma regra de reação que seja satisfeita pelo valor retornado, estratégia de personalização nas linhas 25, 26 e 28 da figura 4.8. Na implementação utilizamos três níveis de avaliação como referência para promover a reação. Cada nível identificado utiliza um parâmetro diferente de reação, o professor/autor poderá modificar alguns desses parâmetros, especificados no subcapítulo 4.3.2. Segue abaixo a definição de cada nível da avaliação e seus valores iniciais de reatividade.

Tabela 4.9 - Parâmetros de referência para a reatividade

Curso: Elementos Básicos	
Avaliação de Baixo Desempenho	Abrangência da reação: Todos os objetos Nº de operadores envolvidos: três Operadores desabilitados: define, interpreta e demonstra.
Avaliação Satisfatória	Não há parâmetros definidos pelo professor/autor, utilizando valor padrão.
Avaliação Acima do Esperado	Abrangência da reação: Próximo objeto Nº de operadores envolvidos: um Operador desabilitado: interpreta

- Avaliação de baixo desempenho: será feita uma revisão dos objetos já apresentados, como padrão os objetos serão todos reapresentados.
- Avaliação satisfatória: todos os objetos de ensino apresentados serão considerados já satisfeitos e marcados como já apresentados, liberando, com isso, a possibilidade de apresentação dos próximos objetos de ensino.

- Avaliação acima do esperado: será oferecido ao estudante a possibilidade de avançar no plano omitindo a aplicação de alguns operadores. Como padrão está sendo considerado como avanço no plano a liberação da aplicação dos operadores interpreta e demonstra, que no exemplo apresentado na Tabela 4.9, foi alterado para desabilitar somente um, o interpreta.

A seguir, iremos apresentar a seqüência de execução da implementação para esclarecer melhor o processo reativo, porém antes precisamos saber quais foram os parâmetros alterados que orientam a reatividade e com esses valores em mente, verificaremos se a execução e a aplicação das regras de reação estão corretas. Para o curso em questão utilizamos como valores de referência os dados apresentados na Tabela 4.9. Estes valores foram inseridos na base de dados conforme as especificações apresentadas no subcapítulo 4.3.2.

Vamos considerar que na primeira avaliação o estudante tenha obtido como resultado um baixo desempenho, pelos parâmetros apresentados na Tabela 4.9 o processo de reação deverá fazer uma revisão com todos os objetos de ensino apresentados por meio da reaplicação dos operadores define, interpreta e demonstra. A Figura 4.15 mostra o momento em que a avaliação de baixo desempenho foi identificada, aqui simulamos por meio de uma opção a ser escolhida na janela, e a Figura 4.16 mostra o mesmo plano já com a reação aplicada, utilizando somente os operadores definidos pelo professor/autor, esse processo é descrito na estratégia de reatividade da Figura 4.8 na linha 27.

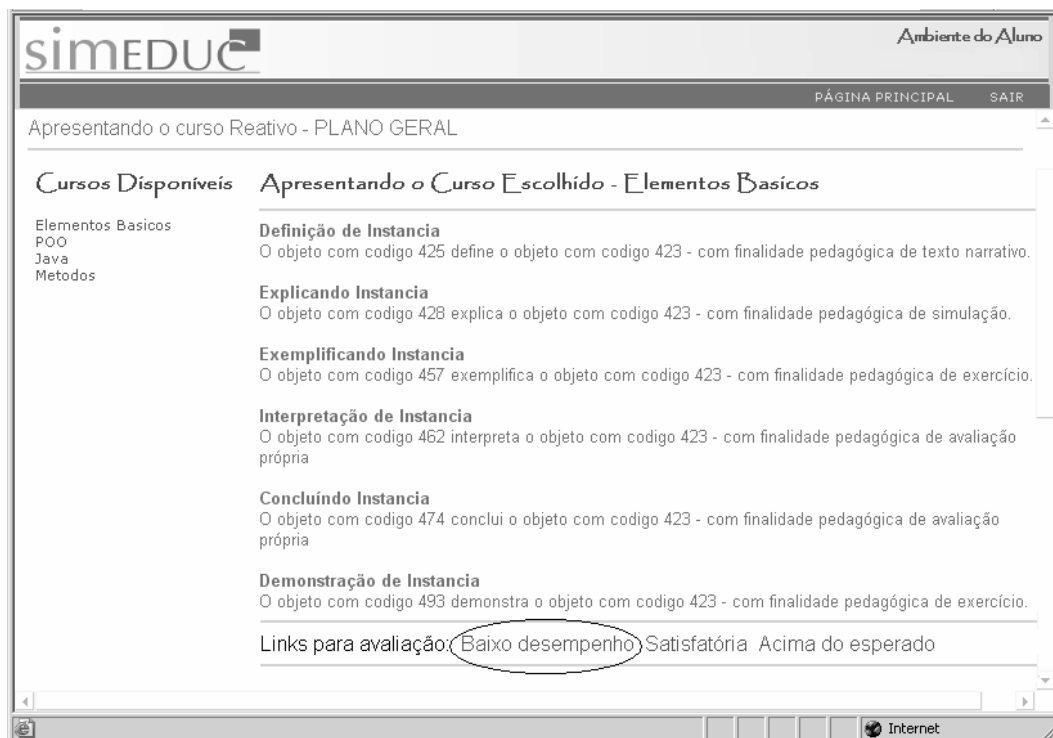


Figura 4.15 - Retorno da avaliação de baixo desempenho

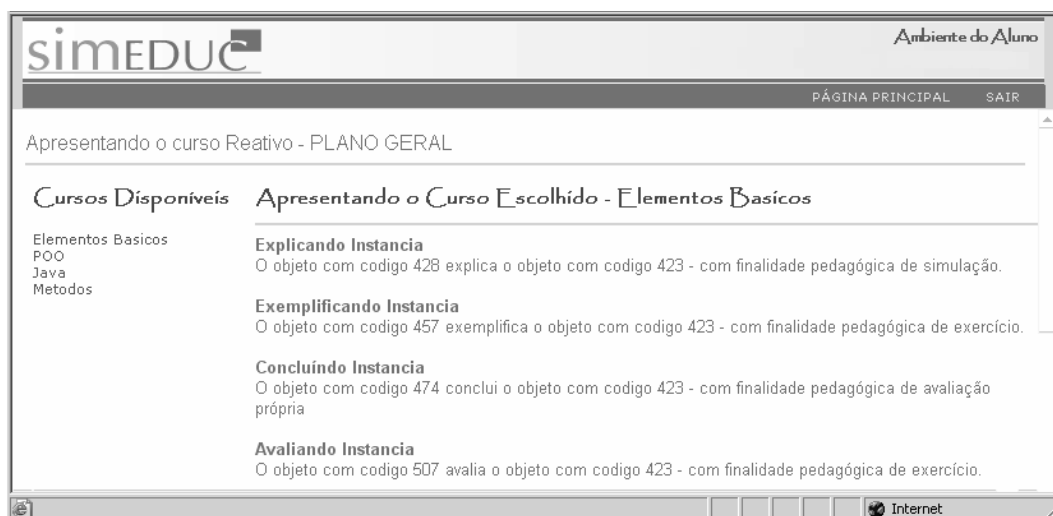


Figura 4.16 - Reação no plano aplicando a revisão de objetos

No próximo exemplo, o estudante obteve o nível de avaliação satisfatório e com este resultado o plano é atualizado e todos os objetos de ensino são marcados como já apresentados e os próximos objetos do plano são habilitados e apresentados, com todos os seus operadores. Esse exemplo está sendo demonstrado por meio das Figuras 4.17 e 4.18.

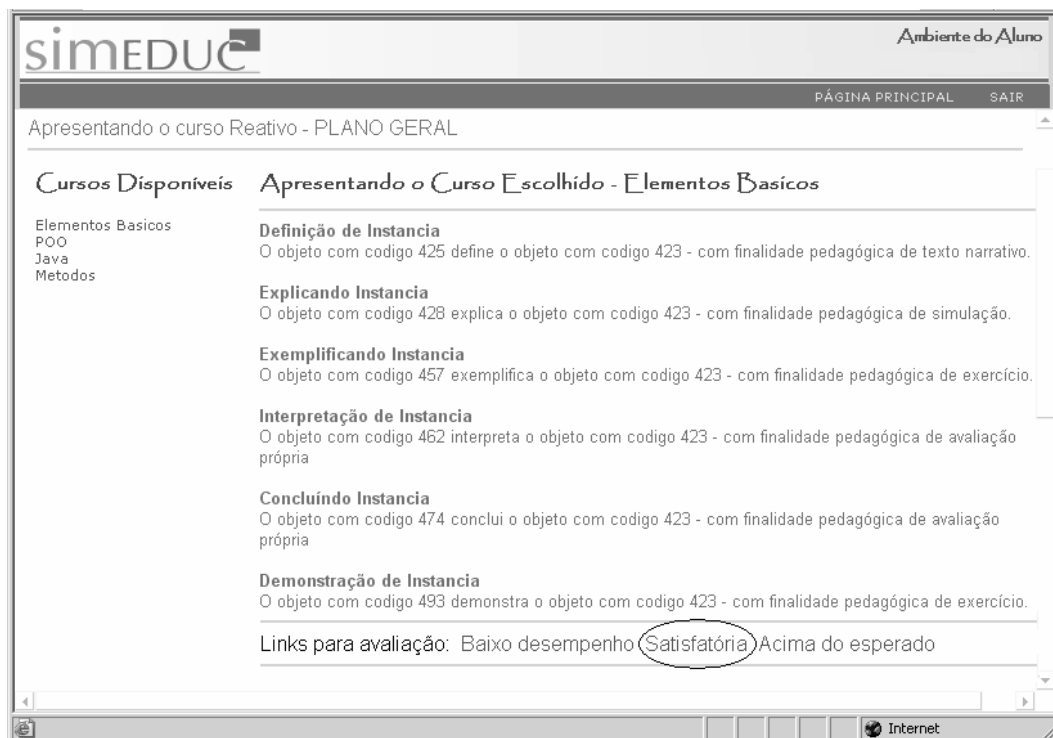


Figura 4.17 - Retorno da avaliação satisfatória

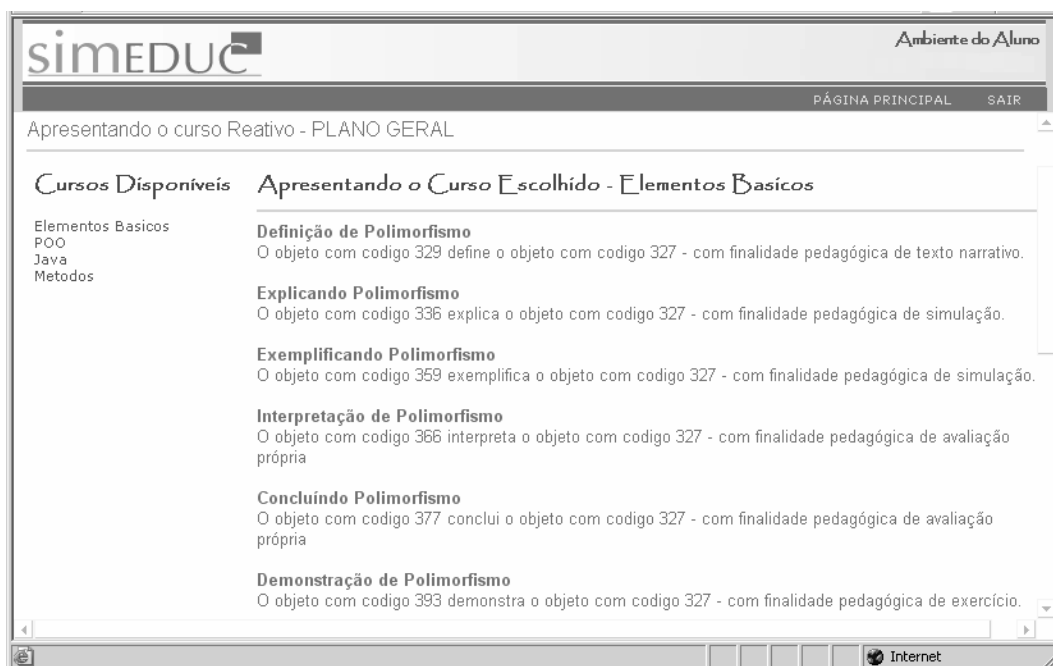


Figura 4.18 - Visualização da continuação normal do plano.

O último exemplo é similar ao processo realizado na avaliação do tipo satisfatória, a diferença é que o professor/autor definiu que neste curso os estudantes que apresentarem um desempenho acima do esperado não necessitam da aplicação

do operador interpreta, por isso, na continuação do plano o objeto que é tratado por este operador não será apresentado, a continuação do plano após a aplicação dessa reatividade é mostrada na Figura 4.19. Esse procedimento que avança no plano é demonstrado na estratégia de reatividade na linha 29 da Figura 4.8.

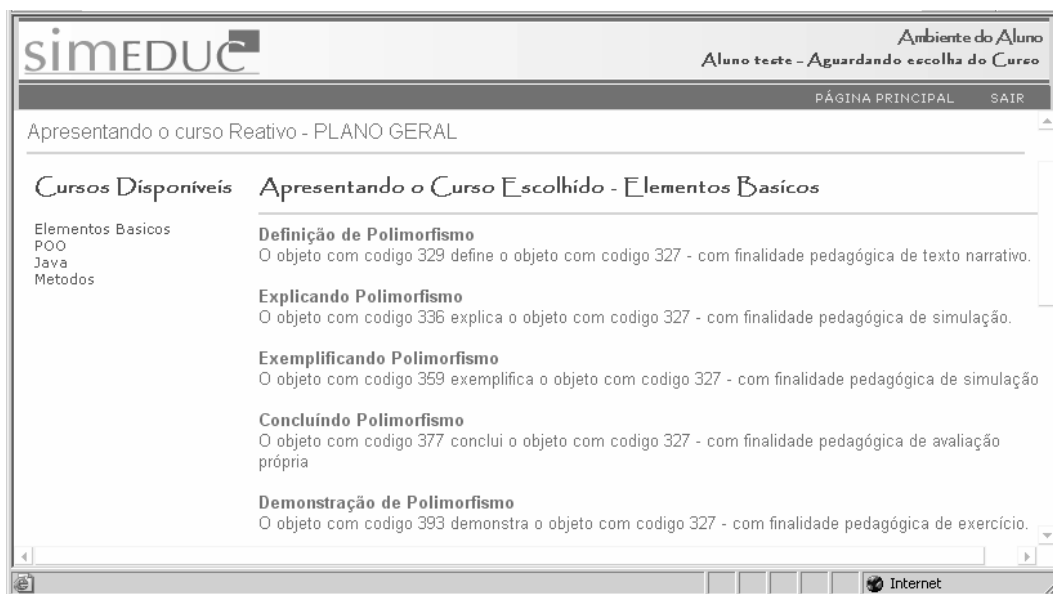


Figura 4.19 - Plano apresentado após a avaliação acima do esperado

Todos os testes realizados, modificando os parâmetros de reação, obtiveram sucesso, o plano de curso foi visualizado várias vezes e conforme os resultados da avaliação os objetos de ensino foram dispostos para apresentação de forma diferente, conforme especificado pelo professor/autor ou pelas definições padrões do agente.

Capítulo 5

Conclusão e Perspectivas Futuras

Como resultado deste trabalho tem-se uma proposta para o gerenciamento automático de apresentação dos conteúdos presentes nos cursos destinados a Educação a Distância apoiada na Web.

A abordagem utiliza técnicas de IA possibilitando que a apresentação dos cursos possa ser ajustada conforme as interações dos estudantes. A técnica demonstra-se inovadora no que diz respeito à adaptação do curso aos estudantes e à aplicação de preferências pedagógicas definidas pelo professor/autor. A abordagem faz uso de uma arquitetura multiagente, na qual os agentes executam as suas tarefas e os resultados são compartilhados, possibilitando assim uma integração total do sistema.

Este trabalho foi desenvolvido com base no sistema SIMEDUC, que possui característica de STI, utilizando como objeto principal de atuação as informações presentes no plano de curso gerado pelo Agente Pedagógico. A geração do plano é baseada em técnicas de planejamento instrucional para ambientes dinâmicos. A manipulação do plano por parte do Agente Assistente consiste em aplicar técnicas de reatividade para ajustar a apresentação do plano conforme as características de navegação do estudante. As regras para a reatividade são pré-definidas na

arquitetura do Agente Assistente, com a possibilidade de inserção de alguns parâmetros de ativação por parte do professor/autor, viabilizando assim uma maior personalização das reações nos cursos.

Procurou-se nessa proposta fornecer ao SIMEDUC maior interatividade com o estudante através da observação dos resultados nas avaliações. Qualquer que seja o resultado obtido pelo estudante, o Agente Assistente irá verificar a possibilidade de aplicar uma reação, com o objetivo de motivá-lo durante o processo ensino/aprendizagem. Essas reações consistem em promover uma revisão dos conteúdos nos quais os estudantes apresentarem dificuldades de interpretação ou resumir partes dos próximos conteúdos a serem apresentados, levando em consideração o alto desempenho dos estudantes. O agente possuiu a capacidade de alterar essas reações, ou seja, se for aplicado a reação que resume o plano e o estudante apresentar dificuldades nos objetos futuros, a próxima reação do agente será de ajustar a apresentação do plano para o nível em que o estudante se encontra.

Tais funcionalidades atribuem ao sistema uma participação direta no processo de ensino/aprendizagem, oferecendo ao estudante um tratamento personalizado, dinâmico e de forma transparente. Consideramos que, a partir disso, o aluno possa encontrar motivação para continuar o curso, pois a apresentação do curso acompanhará o seu nível de entendimento.

É importante mencionar a importância de extensões futuras para o trabalho proposto. Inicialmente, o Agente Assistente deveria receber mais funcionalidades, possibilitando assim maiores interações com o estudante.

Para isto, podem-se considerar a inclusão de mensagens de retorno para os estudantes e professores/autores, possibilitando assim uma melhora nos diagnósticos das dificuldades apresentadas pelos estudantes. Essas mensagens seriam definidas na criação do curso, porém, essa tarefa irá agregar maior complexidade à criação do curso, já que a quantidade de mensagens pode ser consideravelmente grande.

Outra extensão poderia ser a implementação de uma ferramenta de autoria que ofereça facilidades na inclusão dos conteúdos instrucionais no SIMEDUC e, ao mesmo tempo, tenha opções avançadas para gerenciar novas ações dos agentes presentes na arquitetura.

A avaliação também é um fator que deverá receber atenção em implementações futuras, possibilitando que o professor/autor possa elaborar questões de forma livre e que as mesmas sejam aplicadas aos estudantes por meio de avaliações dinâmicas, respeitando o nível de conhecimento de cada estudante.

A utilização de sistemas que utilizem técnicas de personalização tem muito a contribuir para a área da Educação, pois oferece aos professores e estudantes a oportunidade de promoverem uma interação dentro do processo ensino/aprendizagem não oferecidos ou desenvolvidos nos modelos tradicionais de ensino.

Referências Bibliográficas

ARETIO, G. **Educación a distancia hoy**. Madrid: UNED, 1994.

AROYO, L.; KOMMERS P. Intelligent agents for educational computer-aided systems. **Jornal Of Interactive Learning Research**, [Norfolk], v.10, n.3/4, p. 235-242, 1999.

BELLIFEMINE, F.; POGGI, A.; RIMASSA, G. Developing multiagent systems with JADE. In: INTERNATIONAL WORKSHOP ON AGENT THEORIES, ARCHITECTURES, AND LANGUAGES, 7, 2000, Boston. **Proceedings...** [Springer: Heidelberg], 2000. p. 89-103.

BELLIFEMINE, F.; CAIRE, G.; TRUCCO, T. **JADE programmer's guide**. Parma: University of Parma, 2002. Documentação do software. Disponível em: <<http://jade.tilab.com/>>. Acesso em 16 fev. 2005.

BIGUS, J. **Constructing intelligent agents using Java**. 2nd. ed. New York: John Wiley & Sons, 2001.

BLYTHE, J. AI planning in dynamic, uncertain domains. In: AAAI SPRING SYMPOSIUM, 1995, Stanford. **Extending theories of actions: formal theory & practical applications**. Menlo Park: AAAI Press, 1995. Abstract.

BONET, B.; GEFFNER, H. Planning as heuristic search: new results. In: EUROPEAN CONFERENCE ON PLANNING, 5, 1999, Durham. **Recent advances in AI planning**. Berlin: Springer, c2000.

BRUSILOVSKY, P. Adaptive and intelligent technologies for web-based education. **KI - Kunstliche Intelligenz**, v. 13, n. 4, p. 19-25, 1999. Disponível em: <citeseer.nj.nec.com/article/brusilovsky99adaptive.html>. Acesso em: 16 de fev. 2005.

BRUSILOVSKY, P. Adaptive educational systems on the world wide web: a review of available technologies. In: INTERNATIONAL CONFERENCE, ITS, 4, 1998, San Antonio. **Intelligent Tutoring Systems**. Berlin: Springer, c1998. Disponível em: <citeseer.nj.nec.com/article/brusilovsky98adaptive.html>. Acesso em 16 de fev. 2005.

- BRUSILOVSKY, P. Methods and techniques of adaptive hypermedia. **User modeling and user adapted interaction**, Kluwer, v.6, n.2/3, p.87-129, 1996, Special issue on adaptive hypertext and hypermedia.
- BRUSILOVSKY, P.; VASSILEVA, J. Course sequencing techniques for large-scale web-based education. **International journal engineering education and lifelong learning**, Olney, v.13, n. 1/2, 2003.
- CAPUANO, N.; MARSELLA, M.; SALERNO, S. Abits: an agent based intelligent tutoring system for distance learning. INTERNATIONAL WORKSHOP ON ADAPTIVE AND INTELLIGENT WEB-BASED EDUCATIONAL SYSTEMS, Montreal, 2000. **Proceedings...** [S.l.: s.n.], 2000.
- CASAS, L. **Contribuições para a modelagem de um ambiente inteligente de educação baseado em realidade virtual**. 1999. 255 f. Tese (Doutorado) - Universidade Federal de Santa Catarina, Florianópolis, 1999.
- CHO, B. **Dynamic planning models to support curriculum planning and multiple tutoring protocols in intelligent tutoring systems**. 2000. Tese (Doutorado) - Illinois Institute of Technology, 2000.
- CONVERSE, T.; PARK, J. **PHP: a bíblia**. Rio de Janeiro: Campus, 2003.
- COSTA, E. B. Inteligência Artificial em Educação Presencial e à Distância: modelos, tecnologias, aplicações e tendências. In: SIMPÓSIO BRASILEIRO DE INTELIGÊNCIA ARTIFICIAL, 2000, Atibaia. **Anais...** Porto Alegre: SBC, 2000.
- DORÇA, F. A. **Um sistema inteligente multiagente para educação a distância apoiado em web**. 20004. 183 f. Dissertação (Mestrado) - Faculdade de Computação, Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Uberlândia, Uberlândia, 2004.
- DORÇA, F. A.; FERNANDES, M. A.; LOPES, C. R. A multiagent architecture for distance education systems. In: IEEE INTERNATIONAL CONFERENCE ON ADVANCED LEARNING TECHNOLOGIES, 3, Atenas, 2003. **Proceedings...** [Washington: IEEE], 2003. p. 368-369.
- DRUMMOND, M. Situated control rules. INTERNATIONAL CONFERENCE ON PRINCIPLES OF KNOWLEDGE REPRESENTATION AND REASONING, 1989, Toronto. **Proceedings...** San Mateo: M. Kaufmann, c1989.
- EBERSPÄCHER, H. F.; KAESTNER, C. A. A. Arquitetura de um sistema de autoria para a construção de tutores hipermédia. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 7, 1997, São José dos Campos(SP). **Anais...** Porto Alegre: SBC, 1997.
- ESPINOZA, F., HÖÖK, K. **A WWW interface to an adaptive hypermedia system**, 1996. 14 p. Disponível em: <http://www.sics.se/~kia/PAAM_submission.html>. Acesso em: 16 de fev. 2005.

FERREIRA, F. B., LOPES, C. R. Estratégia de personalização reativa na apresentação dos conteúdos de cursos baseados na Web através de um agente assistente In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 15, 2004, Manaus. **Anais...** EDUA, 2004. v.1. p. 687 - 697.

FOUNDATION FOR INTELLIGENT PHYSICAL AGENTS (FIPA). **FIPA ACL message structure specification**, 2000. Document number XC00061D. Disponível em: <<http://www.etse.urv.es/recerca/banzai/toni/MAS/FIPA2000/XC00061D.doc>>. Acesso em 16 fev. 2005.

FIRBY, R. J. An Investigation Into Reactive Planning in Complex Domains. In: NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, 6, 1987, Seattle. **Proceedings...** Los Altos: M. Kaufmann, c1987. p. 202-206.

GAT. E. Integrating planning and reacting in a heterogenous asynchronous architecture for controlling real-world mobile robots. In: NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE, 10, 1992, San Jose. **Proceedings...** Menlo Park: AAAI Press, c1992 p. 809-815.

GEORGEFF, M. **Many agents are better than one**. Stanford: SRI International, 1987. (Technical Report Technical, Note 417).

GEORGEFF, M. Reasoning about procedural knowledge. In: CONFERENCE ON COMPUTERS IN AEROSPACE, 5, 1985, Long Beach. **Proceedings...** [S.l.: s.n.], 1985. p. 41-49.

GEORGEFF, M. et al. **Reasoning and planning in dynamic domains: an experiment with a mobile robot**. Stanford: SRI International, 1987. (Technical Report Technical, Note 390).

HASEMANN, J. M. Robot control architectures: application requirements, approaches, and technology. In: INTELLIGENT ROBOTS AND COMPUTER VISION, 14, 1995, Philadelphia. **Proceedings...** Bellingham, 1995. p. 748.

HODGINS, W. **Draft Standard for Learning Object Metadata, IEEE Learning Technology Standards Committee (LSTC)**. 2001. Database online. Disponível em: <http://ltsc.ieee.org/wg12/files/LOM_1484_12_1_v1_Final_Draft.pdf>. Acesso em 13 fev. 2005.

HOFFMANN, J.; Nebel, B. The FF planning system: fast plan generation through heuristic search. **Journal of Artificial Intelligence Research**, Marina del Rey, v. 14, p. 253-302, 2001.

Kautz, H. **BLACKBOX a SAT technology planning system**, 2003. Disponível em: <<http://www.cs.washington.edu/homes/kautz/satplan/blackbox/>>. Acesso em: 24 de fev. 2005.

KEMP, R.; SMITH, S. Using planning techniques to provide feedback in interactive learning environments. In: IEEE INTERNATIONAL CONFERENCE ON TOOLS WITH ARTIFICIAL INTELLIGENCE, 6, New Orleans, 1994. **Proceedings...** [S.l.: s.n.], 1994. p. 700–703.

LOPES, C. R. **Planejamento baseado em expectativas**. 1998. Tese (Doutorado) - Universidade de São Paulo, São Paulo, 1998.

LOVEJOY, W. S. A survey of algorithmic methods for partially observed Markov decision processes. **Annals of Operations Research**, Basel, n. 28, p. 47-66, 1991.

MILLS, B. I. Using the atlas planning engine to drive an intelligent tutoring system: CIRCSIM-tutor version 3. In: FLAIRS CONFERENCE 2001. [s.n.], 2001. p. 211–215. Disponível em: <citeseer.nj.nec.com/449649.html>. Acesso em: 16 de fev. 2005.

MOREIRA, M. MASINI, E. S. **Aprendizagem significativa**: A teoria de David Ausubel. São Paulo: Moraes, 1982.

MURRAY, T. Authoring intelligent tutoring systems: an analysis of the state of the art. **International Journal of Artificial Intelligence in Education**, Leeds, v. 10, n. 1, p. 98-129, 1999.

MUSA, D. L. **Um sistema de alertas inteligentes para ambientes de ensino na internet**. 2001. 123 f. Dissertação (Mestrado) - Programa de Pós-graduação em Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2001.

MUSLINER, D.; DURFEE, E. D.; SHIN, K. G. Circa: a cooperative intelligent-real-time control architecture. **IEEE Trans. Systems, Man, and Cybernetics**, New York. v.23, n.6, p.1561-1574, 1993.

MUSLINER, D.; DURFEE, E. D.; SHIN, K. G. World modeling for the dynamic construction of real-time control plans. **Artificial Intelligence**, v.74, n1, p.83-127, 1995.

NILSSON, N. J. Teleo-reactive programs for agent control. **Journal of Artificial Intelligence Research**, 1, p.139-158, Aug. 1993/ June 1994.

NUNES, I. B. Noções de educação a distância. **Revista Educação a Distância**, Brasília. v.1, n.5, p.7–25, abr. 1994.

PALAZZO, L. A. M. **Modelos proativos para hipermídia adaptativa**. Tese (Doutorado) - Programa de Pós-graduação em Computação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2000.

QUEIROZ, B. et al. Uma proposta de um agente pedagógico para planejamento instrutivo. In: SEMINÁRIO NACIONAL DE TECNOLOGIA PARA EAD, 2, 2000, Uberlândia. **Anais...** Uberlândia: UFU, 2002.

QUEIROZ, B. P. et al. Planejamento instrucional reativo para o sequenciamento automático do conteúdo. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 14, 2003, Rio de Janeiro. **Anais...** Porto Alegre: SBC, 2003. p.593-603.

QUEIROZ, B. **Sequenciamento automático de conteúdo em um sistema de educação a distância**. 2003. Dissertação (Mestrado) - Faculdade de Computação, Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Uberlândia, Uberlândia, 2003.

QUEIROZ, B.; LOPES, C.; FERNANDES, M. Geração automática de currículo para um sistema educacional baseado na Web. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 13, 2002, São Leopoldo. **Anais...** Unisinos, 2002. p.515-518.

RICH, E. **Inteligência artificial**. São Paulo: McGraw-Hill, 1988.

RUSSEL, S.; NORVING, P. **Artificial intelligence: a modern approach**. New Jersey: Prentice Hall, 1995.

SILVA, F. **Algoritmos para planificação baseada em STRIPS**. 2002. Dissertação (Mestrado) - Universidade Federal do Paraná, Curitiba, 2000.

SMITH, D.; FRANK, J.; J'ONSSON, A. **Bridging the gap between planning and scheduling**. 2000. Disponível em: <citeseer.nj.nec.com/smith00bridging.html>. Acesso em: 16 fev. 2005.

SOUZA, J. T. **Teoria da revisão de primeira ordem para a modelagem do estudante**. 2000. 142 f. Dissertação (Mestrado) - Departamento Computação, Universidade Federal do Ceará, Ceará, 2002. Disponível em: <<http://www.site.uottawa.ca/~jsouza/Papers/Dissertacao.pdf>>. Acesso em: 16 fev. 2005.

SUEHRING; S. **MySQL: a bíblia**. Rio de Janeiro: Campus, 2002.

THIEBAUX, S. et al. **Stochastic model of actions for anytime planning under uncertainty**. Berkeley: The International Computer Science Institute, 1993. (Technical Report, TR-93-027)

COSTA, M.T.C. da. **Uma arquitetura baseada em agentes para suporte ao ensino a distância**. Tese (Doutorado em Engenharia de Produção) - Programa de Pós-Graduação em Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 1999.

VASSILEVA, J. Reactive instructional planning to support interacting teaching strategies. In: WORLD CONFERENCE ON AI AND EDUCATION AIED'95, 7, 1995, Washington. **Proceedings...** [S.l.: s.n.], 1995. p.334-342.

WASSON, B. **Determining the Focus of Instruction: Content Planning for Intelligent Tutoring Systems**. 1990. Tese (Doutorado) - University of Saskatchewan, Saskatchewan, 1990.

WASSON, B. PEPE: A computational framework for a content planner. In: DIJKSTRA, S.; KRAMMER, H.; MERRIENBOER, J. VAN (Ed.). **Instructional Models in Computer-based learning environments**. New York: Springer-Verlag: 1992. v. 104, p. 153–170. NATO ASI Series F.

WELD, D. S. Recent advances in AI planning. **AI Magazine**, La Canadá, v.20, n.2, p.93–123, 1999. Disponível em: <citeseer.nj.nec.com/article/weld99recent.html>. Acesso em: 16 fev. 2005.

WENGER, E. **Artificial intelligence and tutoring systems**. Los Altos: M. Kaufmann, 1987.

Livros Grátis

(<http://www.livrosgratis.com.br>)

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)
[Baixar livros de Literatura de Cordel](#)
[Baixar livros de Literatura Infantil](#)
[Baixar livros de Matemática](#)
[Baixar livros de Medicina](#)
[Baixar livros de Medicina Veterinária](#)
[Baixar livros de Meio Ambiente](#)
[Baixar livros de Meteorologia](#)
[Baixar Monografias e TCC](#)
[Baixar livros Multidisciplinar](#)
[Baixar livros de Música](#)
[Baixar livros de Psicologia](#)
[Baixar livros de Química](#)
[Baixar livros de Saúde Coletiva](#)
[Baixar livros de Serviço Social](#)
[Baixar livros de Sociologia](#)
[Baixar livros de Teologia](#)
[Baixar livros de Trabalho](#)
[Baixar livros de Turismo](#)