

Universidade Federal do Ceará  
Centro de Ciências  
Departamento de Computação  
Mestrado em Ciência da Computação

*Dissertação de Mestrado*

**Teoria da Revisão de Primeira Ordem para a  
Modelagem do Estudante**

**Jerffeson Teixeira de Souza**

Orientador : Prof. Dr. Tarcísio Haroldo Cavalcante Pequeno  
Co-Orientador : Prof. Dr. Marcelino Cavalcante Pequeno

Fortaleza, Fevereiro de 2000

Universidade Federal do Ceará  
Centro de Ciências  
Departamento de Computação  
Mestrado em Ciência da Computação

**Jerffeson Teixeira de Souza**

**Teoria da Revisão de Primeira Ordem para a  
Modelagem do Estudante**

Dissertação apresentada ao curso de Mestrado em  
Ciência da Computação da Universidade Federal  
do Ceará como parte dos requisitos para obtenção  
do grau de Mestre em Ciência da Computação.

*”Trabalho intelectual é uma expressão errada.  
Não é ‘trabalho’ - é prazer, dissipação, nossa maior recompensa.”*

**Mark Twain**

© Jerffeson Teixeira de Souza, 2000.

Todos os direitos reservados.

# Agradecimentos

Inicialmente gostaria de agradecer a DEUS, que me iluminou por essa árdua jornada.

Aos meus orientadores, Prof. Tarcísio Pequeno e Prof. Marcelino Pequeno, que apesar do pouco tempo disponível conseguiram com muita boa vontade engrandecer os resultados obtidos com esse trabalho.

Aos meus pais, pelo apoio oferecido durante toda a minha educação. A meu irmão Cideley, que como um bom irmão mais velho participou ativamente de todas as decisões relacionadas ao desenvolvimento dessa dissertação.

À minha amada noiva Shirliane, por seu incentivo e apoio em todos os momentos. Por ter partilhado comigo todas as dificuldades e principalmente pelo seu interesse no que se referia à minha pesquisa, fazendo questão de entender e participar de todo o processo.

Ao Prof. Fernando Carvalho, que me indicou essa área de pesquisa e que enriqueceu meu trabalho com suas sugestões e seus artigos.

Aos meus colegas de mestrado, que sabem como ninguém as dificuldades por que passamos juntos.

Ao secretário do mestrado Orley e à secretária do LIA Débora, pelo apoio na resolução dos problemas burocráticos.

À FUNCAP por ter financiado meus estudos de Mestrado e consequentemente possibilitado o desenvolvimento dessa pesquisa.

# Resumo

Há vários anos, pesquisadores de todo o mundo têm reconhecido no computador um grande potencial como ferramenta educacional. Durante as últimas décadas um grande número de pesquisas vêm sendo desenvolvidas na tentativa de criação de sistemas computacionais capazes de simular o processo de ensino tradicional, surgem daí os sistemas tutores. Posteriormente, foi sugerido o desenvolvimento dos chamados sistemas tutores inteligentes, que introduziram características que possibilitam a criação de instruções individualizadas e adaptativas. Percebe-se que um componente crítico desses sistemas é a forma como o conhecimento do estudante é modelado internamente. Esse processo é conhecido como modelagem do estudante.

Várias técnicas de modelagem de estudantes em sistemas tutores inteligentes foram propostas, no entanto percebe-se que tais técnicas apresentam uma série de restrições, seja pela baixa eficiência de representação ou pela dificuldade computacional de desenvolvimento. Uma das abordagens mais bem sucedidas é a que propõe a utilização de uma técnica de aprendizagem de máquina conhecida como teoria da revisão proposicional, que revisa uma teoria proposicional inicial com o uso de um conjunto de exemplos. Uma série de vantagens podem ser obtidas com a utilização dessa técnica em relação a sistemas que induzem o conhecimento do estudante a partir do nada, ou seja, sem a utilização de uma teoria inicial. A principal delas é a de que a teoria da revisão permite o desenvolvimento de modelos eficientes a partir de um pequeno número de exemplos de treinamento, ou seja, erros

cometidos pelo estudante.

Apesar de apresentar uma série de vantagens consideráveis, o uso da teoria da revisão proposicional recai nas limitações de representação apresentadas pela lógica proposicional. Nesse contexto, esse trabalho propõe a utilização da lógica de primeira ordem no uso da teoria da revisão como modeladora de estudantes.

Adicionalmente, é proposta a geração e utilização de um modelo que representa o conhecimento de um certo grupo de estudantes. Tal modelo pode ser usado para o desenvolvimento de instruções para o grupo como um todo ou como teoria inicial na geração de modelos de novos estudantes, diminuindo ainda mais o tempo de geração de modelos precisos.

Os ganhos computacionais da utilização da teoria da revisão como modeladora de estudantes, em relação a sistemas de modelagem que se utilizam da indução pura, são comprovados através de uma série de simulações. Além disso, simulações de modelagem com a utilização do modelo do grupo são apresentadas e os ganhos obtidos são discutidos.

# Abstract

It has been long years since researchers have recognized the computer's enormous potential as an educational tool. During the past decades, a considerable amount of research have been developed on trying to create computational systems that simulate the traditional teaching process, then he so called tutoring systems. After this, the development of intelligent tutoring systems was suggested. They introduced characteristics that enable the creation of individual and adaptative instructions efficiently. It is noticeable that a critical component of these systems relies in the way the student's knowledge is modeled. This process is called student modeling.

Several student modeling approaches have been proposed. However, it seems that such approaches have lots of restrictions due to the low representation efficiency or due to the computational development. One the most successful approaches is the one that proposes the use of a machine learning technique called proposicional revision theory. This technique revises a proposicional initial theory by using a set of examples. Numerous advantages can be obtained by using this technique instead of systems that induce the student's knowledge from scratch, in other words, without the use of an initial theory. The main advantage is that the revision theory allows the development of effective models by using a small number of training examples, i. e., errors made by the student.

Despite the fact that this approach has a lot of considerable advantages,

the use of propositional revision theory has the representation limitations present in the propositional logic. In this context this work proposes the use of first order logic in the use of revision theory as a student modeler.

In addition, it is proposed the generation and use of a model that represents the knowledge of a certain student group. Such model can be used to develop instructions for the group or as the initial theory in the generation of new student models, so that it can diminish the time required for the generation of effective models.

The computational advantages of the revision theory as a student modeler, in relation to modeling systems that utilize pure induction, are demonstrated by a series of simulations. Moreover, modeling simulations that use the group model are presented and its advantages are discussed.

# Sumário

|          |                                      |          |
|----------|--------------------------------------|----------|
| <b>1</b> | <b>Introdução</b>                    | <b>1</b> |
| 1.1      | Introdução . . . . .                 | 2        |
| 1.2      | Motivação . . . . .                  | 2        |
| 1.3      | Objetivos . . . . .                  | 4        |
| 1.4      | Organização da Dissertação . . . . . | 4        |
| <b>2</b> | <b>Sistemas Tutores Inteligentes</b> | <b>6</b> |
| 2.1      | Introdução . . . . .                 | 7        |
| 2.2      | Arquitetura de um STI . . . . .      | 8        |
| 2.2.1    | Modelo do Domínio . . . . .          | 10       |
| 2.2.2    | Modelo do Estudante . . . . .        | 11       |
| 2.2.3    | Modelo Pedagógico . . . . .          | 12       |
| 2.2.4    | Interface . . . . .                  | 14       |
| 2.3      | Exemplos de STIs . . . . .           | 16       |
| 2.3.1    | SCHOLAR . . . . .                    | 16       |
| 2.3.2    | SOPHIE . . . . .                     | 18       |
| 2.3.3    | WEST . . . . .                       | 19       |
| 2.3.4    | GUIDON . . . . .                     | 20       |

|          |   |           |
|----------|---|-----------|
| 2.3.5    | HYDRIVE . . . . .                             | 22        |
| 2.4      | Considerações Finais . . . . .                | 24        |
| <b>3</b> | <b>A Modelagem do Estudante em STIs</b>       | <b>25</b> |
| 3.1      | Introdução . . . . .                          | 26        |
| 3.2      | Modelo Overlay . . . . .                      | 26        |
| 3.3      | Biblioteca de Erros . . . . .                 | 28        |
| 3.4      | Biblioteca de Erros Estendida . . . . .       | 29        |
| 3.5      | Modelagem Indutiva . . . . .                  | 30        |
| 3.6      | Teoria da Revisão Proposicional . . . . .     | 31        |
| 3.7      | Considerações Finais . . . . .                | 32        |
| <b>4</b> | <b>O Sistema de Revisão</b>                   | <b>33</b> |
| 4.1      | Introdução . . . . .                          | 34        |
| 4.2      | O Algoritmo de Revisão . . . . .              | 34        |
| 4.3      | Os Pontos de Revisão . . . . .                | 36        |
| 4.4      | Os Operadores de Revisão . . . . .            | 37        |
| 4.5      | Conclusões . . . . .                          | 40        |
| <b>5</b> | <b>Modelagem baseada na Teoria da Revisão</b> | <b>41</b> |
| 5.1      | Introdução . . . . .                          | 42        |
| 5.2      | O Uso da Teoria da Revisão . . . . .          | 42        |
| 5.3      | O Processo de Modelagem . . . . .             | 42        |
| 5.4      | Vantagens da Técnica . . . . .                | 43        |
| 5.5      | Considerações Finais . . . . .                | 44        |
| <b>6</b> | <b>O Uso da Lógica de Primeira Ordem</b>      | <b>45</b> |

|          |   |           |
|----------|---|-----------|
| 6.1      | Introdução . . . . .                            | 46        |
| 6.2      | O Uso da Lógica Proposicional . . . . .         | 46        |
| 6.3      | O Uso da Lógica de Primeira Ordem . . . . .     | 47        |
| 6.4      | Vantagens Obtidas . . . . .                     | 48        |
| 6.5      | Considerações Finais . . . . .                  | 49        |
| <b>7</b> | <b>Geração e Uso do Modelo do Grupo</b>         | <b>51</b> |
| 7.1      | Introdução . . . . .                            | 52        |
| 7.2      | Geração do Modelo do Grupo . . . . .            | 53        |
| 7.3      | Modelagem a partir do Modelo do Grupo . . . . . | 54        |
| 7.3.1    | O Algoritmo de Modelagem . . . . .              | 55        |
| 7.3.2    | Vantagens da Técnica . . . . .                  | 57        |
| 7.4      | Conclusões . . . . .                            | 58        |
| <b>8</b> | <b>Estruturação do Conteúdo</b>                 | <b>59</b> |
| 8.1      | Motivação . . . . .                             | 60        |
| 8.2      | Redes Semânticas . . . . .                      | 61        |
| 8.3      | Estruturação com Redes Semânticas . . . . .     | 65        |
| 8.3.1    | Inicialização da Rede . . . . .                 | 65        |
| 8.3.2    | Modificação da Rede . . . . .                   | 66        |
| 8.3.3    | Uso e Apresentação da Rede Final . . . . .      | 67        |
| 8.4      | Conclusões . . . . .                            | 67        |
| <b>9</b> | <b>Resultados Experimentais</b>                 | <b>69</b> |
| 9.1      | Introdução . . . . .                            | 70        |
| 9.2      | O Modelo do Domínio . . . . .                   | 71        |
| 9.3      | A Geração dos Erros dos Estudantes . . . . .    | 71        |

|           |  |           |
|-----------|--|-----------|
| 9.4       | A Modelagem dos Estudantes a partir do Modelo do Domínio | 73        |
| 9.5       | A Modelagem dos Estudantes Indutiva . . . . .            | 75        |
| 9.6       | A Geração do Modelo do Grupo . . . . .                   | 76        |
| 9.7       | A Modelagem dos Estudantes a partir do Modelo do Grupo . | 76        |
| 9.8       | Comparação dos Resultados . . . . .                      | 77        |
| 9.9       | Conclusões . . . . .                                     | 79        |
| <b>10</b> | <b>Conclusões e Trabalhos Futuros</b>                    | <b>81</b> |
| 10.1      | Trabalhos Futuros . . . . .                              | 82        |
| 10.1.1    | Testes Pedagógicos . . . . .                             | 82        |
| 10.1.2    | Técnicas de Representação mais Robustas . . . . .        | 83        |
| 10.1.3    | Desenvolvimento de um Shell de STI . . . . .             | 83        |
| 10.2      | Conclusões . . . . .                                     | 84        |
| <b>A</b>  | <b>Artigos</b>   | <b>87</b> |
| A.1       | Artigo SBIE'99 . . . . .                                 | 88        |
| A.2       | Artigo ENIA'99 . . . . .                                 | 107       |

# Lista de Figuras

|     |  |    |
|-----|--|----|
| 2.1 | Arquitetura de um STI. . . . .   | 9  |
| 3.1 | Exemplo de um Modelo Overlay. . . . .                                    | 27 |
| 3.2 | Exemplo de uma Biblioteca de Erros. . . . .                              | 28 |
| 3.3 | Exemplo de uma Biblioteca de Erros Estendida. . . . .                    | 30 |
| 4.1 | Algoritmo de revisão do FORTE. . . . .                                   | 35 |
| 7.1 | Algoritmo de Geração do Modelo do Grupo. . . . .                         | 53 |
| 7.2 | Algoritmo de Modelagem do Estudante a partir do Modelo do Grupo. . . . . | 56 |
| 8.1 | Rede Semântica. . . . .  | 62 |
| 9.1 | Modelo dos Relacionamento da Família. . . . .                            | 72 |
| 9.2 | Alterações feitas no Modelo dos Relacionamentos da Família. . . . .      | 73 |
| 9.3 | Resultados Comparativos. . . . .   | 78 |
| A.1 | Algoritmo de revisão do FORTE. . . . .                                   | 94 |
| A.2 | Interfaces do Sistema de Avaliação. . . . .                              | 96 |
| A.3 | Arquitetura do Cliente de Avaliação. . . . .                             | 97 |

|     |   |     |
|-----|---|-----|
| A.4 | Arquitetura do Servidor de Avaliação. . . . . | 99  |
| A.5 | FORTE Algorithm. . . . .                      | 113 |

# Lista de Tabelas

|     |   |    |
|-----|---|----|
| 9.1 | Precisão dos Modelos gerados a partir do Modelo do Domínio.   | 74 |
| 9.2 | Precisão dos Modelos gerados indutivamente. . . . .           | 75 |
| 9.3 | Precisão dos Modelos gerados a partir do Modelo do Grupo. . . | 77 |

# Capítulo 1

## Introdução

## 1.1 Introdução

Há vários anos, pesquisadores de todo o mundo têm reconhecido no computador um grande potencial como ferramenta educacional. Nesse sentido, vários esforços vêm sendo realizados na tentativa de desenvolvimento de sistemas computacionais de ensino que possam, efetivamente, colaborar com o processo ensino-aprendizado [Burns, Parlett e Redfield, 90] [Marietto] [Richards e Mooney]. Tais sistemas são conhecidos como sistemas tutores. No entanto, algumas décadas de pesquisa não foram suficientes para que se conseguisse gerar ferramentas realmente capazes de produzir instruções individualizadas e pedagogicamente eficientes, fato esse que, em parte, decorre da própria complexidade de entendimento e representação de estruturas pedagógicas.

## 1.2 Motivação

Diversas pesquisas vêm sendo feitas no sentido de criação de sistemas de ensino inteligentes, ou seja, sistemas tutores que, utilizando-se de técnicas da área de inteligência artificial, incorporem características inteligentes. Um dos principais componentes desses sistemas é o Modelo do Estudante, que modela dinamicamente o conhecimento do estudante que se utiliza desse sistema. A idéia é que o sistema tutor utilize esse modelo para disponibilizar uma abordagem de ensino individualizada para cada estudante, considerando as características particulares de aprendizagem do mesmo.

Infelizmente, a dificuldade na construção desse modelo tem impossibilitado o desenvolvimento de sistemas de ensino realmente eficientes. Apesar de mais de duas décadas de pesquisa nessa área, onde uma grande variedade de técnicas de modelagem de estudantes foram desenvolvidas, a tarefa prática de incorporar essas técnicas dentro de sistemas tutores reais tem se revelado difícil [Chaiben].

Dessa forma, o grande desafio atual para o ramo da modelagem do estudante é o desenvolvimento de técnicas que se mostrem tanto pedagogicamente práticas como computacionalmente eficientes. Esse é o objetivo dessa dissertação.

Proposto inicialmente por Baffes [Baffes e Mooney, 92], a técnica de aprendizagem de máquina conhecida como teoria da revisão proposicional vem sendo recentemente utilizada como ferramenta para modelagem de estudantes em sistemas tutores inteligentes. No entanto, a utilização da lógica proposicional como forma de representação tem restringido bastante o uso dessa técnica. Nesse sentido, nesse trabalho é proposta a utilização da lógica de primeira ordem na aplicação da teoria da revisão como modeladora do estudante.

Adicionalmente, a aplicação da teoria da revisão permite o desenvolvimento de uma teoria que modele o conhecimento de um grupo de estudantes, o modelo do grupo. A geração e utilização desse modelo é proposta também nesse trabalho.

### 1.3 Objetivos

Dentre os principais objetivos dessa dissertação pode-se destacar, de forma resumida, os seguintes :

- Propor a utilização da lógica de primeira ordem na aplicação da teoria da revisão na modelagem de estudantes em sistemas tutores;
- Propor e discutir a geração e utilização do modelo do grupo no processo de modelagem de novos estudantes;
- Analisar os ganhos computacionais obtidos tanto com o uso da teoria da revisão de primeira ordem como do modelo do grupo, de forma a comparar tais resultados com aqueles obtidos a partir da modelagem de estudantes de forma puramente indutiva.

### 1.4 Organização da Dissertação

Essa dissertação está organizada em oito capítulos. O capítulo 2 apresenta uma visão geral do desenvolvimento de sistemas tutores, discutindo cada um dos componentes de um sistema tutor inteligente e apresentando alguns exemplos de sistemas já desenvolvidos. No capítulo 3, as técnicas mais relevantes historicamente para modelagem de estudantes são apresentadas e discutidas. O capítulo 4 mostra com detalhes a ferramenta escolhida para processar as revisões necessárias. O capítulo 5 descreve a forma como a teoria

da revisão pode ser utilizada na modelagem de estudantes e as vantagens obtidas com essa abordagem. No capítulo 6 é proposta a utilização da lógica de primeira ordem no processo de modelagem de estudantes baseado na teoria da revisão, e os resultados dessa aplicação são analisados. No capítulo 7 é discutido o desenvolvimento do modelo de um grupo de estudantes e sua utilização na criação de novos modelos individuais. No capítulo 8 a utilização da tecnologia de Redes Semânticas é proposta no sentido de estruturação do conhecimento armazenado no sistema tutor. O capítulo 9 apresenta e discute uma série de experiências práticas que foram desenvolvidas no sentido de demonstrar a eficiência da técnica proposta nessa dissertação. Finalmente, o capítulo 9 discute uma série de propostas de trabalhos futuros e apresenta as conclusões desse trabalho.

## **Capítulo 2**

# **Sistemas Tutores Inteligentes**

## 2.1 Introdução

O desenvolvimento de sistemas tutores inicia-se com a proposição de um paradigma conhecido como CAI (*Computer-Aided Instruction*). Um programa CAI tem como objetivo apresentar automaticamente o conteúdo programático de certo domínio de conhecimento que se deseja ensinar. Isso funcionou inicialmente como um livro eletrônico, com a vantagem de o usuário (estudante) poder escolher o nível de detalhes da apresentação desse conteúdo. Tal abordagem permitia, de uma certa forma, que o aluno pudesse ter o controle sobre a forma como o material seria apresentado. Uma grande restrição desse sistema é o fato de tais escolhas terem que ser pré-programadas pelo desenvolvedor do sistema, o que gerava um grande esforço na criação destes e ainda assim não se podia garantir que todas as necessidades do estudante seriam satisfeitas. Várias críticas foram então feitas aos sistemas CAI [Baffes e Mooney, 96] [Chaiben].

A principal característica que se desejava incorporar em tais sistemas era a adaptabilidade desses em relação a cada estudante, ou seja, o sistema deveria tirar conclusões sobre as potencialidades e deficiências de cada aluno baseado na interação com esse estudante, assim como é feito pelo professor no processo de ensino convencional, e poder, através dessas conclusões, gerar instruções individualizadas.

## 2.2 Arquitetura de um STI

Novos esforços se multiplicaram no desenvolvimento de sistemas ICAI (*Intelligent CAI*), também chamados ITS (*Intelligent Tutoring Systems*) (STI - Sistemas Tutores Inteligentes) [Wenger, 87], que incorporam técnicas de inteligência artificial.

A proposta fundamental de tais sistemas é comunicar o conhecimento e/ou habilidades para o estudante resolver problemas dentro de um determinado domínio de conhecimento. As funções operacionais básicas, como descrito em [Chaiben], são determinadas por quatro componentes principais, que são :

- Modelo do Domínio : É o objeto da comunicação.
- Modelo do Estudante : Modela o conhecimento do estudante.
- Modelo Pedagógico : Representa os métodos e técnicas didáticas no processo de comunicação do conhecimento.
- Interface : É a forma como a comunicação será realizada com o meio externo ao sistema.

O desenvolvimento de um STI requer a aplicação integrada de todos esses quatro componentes, cujas inter-relações podem ser compreendidas de acordo com a Figura 2.1. Tal desenvolvimento requer, portanto, uma abordagem sistemática para integrar os vários tipos de especialidades dentro de

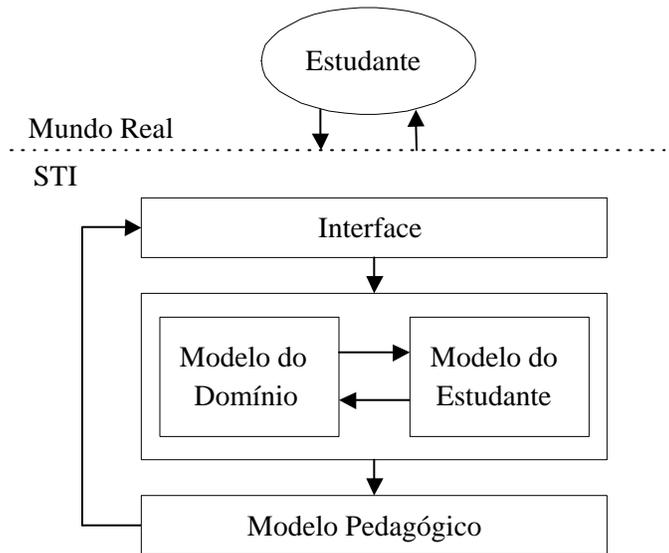


Figura 2.1: Arquitetura de um STI.

um único sistema. Durante uma sessão educacional, o sistema monitora a performance do estudante e tenta apurar o conhecimento que esse estudante detém. Este processo de diagnóstico é realizado pela comparação do estado de conhecimento atual do estudante com o conhecimento contido no modelo do domínio. Os resultados dessa comparação são passados para o modelo pedagógico, onde as decisões são tomadas sobre qual, quando e como a informação será transmitida através da interface do sistema com o estudante.

Em seguida é apresentada uma descrição mais detalhada das funções de cada um dos componentes :

### 2.2.1 Modelo do Domínio

Esse modelo é fundamentalmente uma base de conhecimento. Ele armazena o conteúdo disciplinar organizado de alguma maneira para representar o conhecimento de um especialista ou de um professor [Chaiben].

Esta base de conhecimento contém os elementos para que o estudante aprenda o conhecimento do domínio e os procedimentos necessários para que ele possa utilizá-los na resolução dos problemas em uma determinada área desse domínio. Para isso, esse conhecimento deve ser mapeado simbolicamente para que o computador possa armazená-lo e manipulá-lo durante as interações com o estudante.

Uma boa representação de conhecimento nesse ponto é fundamental para que o sistema possa realizar o ensino individualizado. Além de servir como uma fonte para o conhecimento a ser apresentado, o Módulo do Domínio também serve como um padrão para a avaliação do desempenho dos estudantes, no sentido em que o sistema conhece o conteúdo que o estudante deve ter.

Diversos esforços vêm sendo realizados no sentido de buscar boas representações que possam refletir da melhor forma possível o conteúdo disciplinar, porém sabe-se que a complexidade dessas representações cresce bastante na medida em que se tenta sofisticar uma representação. Dessa forma, o desafio é fornecer uma boa representação de um domínio particular, suficiente para suportar o nível desejado de compreensão e, portanto, proporcionar a flexi-

bilidade fundamental no ensino. No entanto, nenhuma forma particular de representação parece adequada para representar qualquer tipo de conhecimento, mas tipos diferentes de raciocínio e de conhecimento requerem diferentes representações para um uso eficiente. Logo, a escolha da representação do conhecimento em um sistema tutor depende do tipo de conhecimento a ser armazenado e da utilização pretendida. Alguns métodos de representação de Inteligência Artificial são utilizados para representar o conhecimento do domínio, entre eles o desenvolvimento de redes semânticas, a aplicação de regras de produção, representação procedimental e a construção de *frames* e *scripts*.

De forma geral, o que pode ser concluído é que a capacidade de ensino do sistema requer um conhecimento rico e suficientemente estruturado de modo a facilitar os tipos desejados de raciocínio envolvidos no processo ensino-aprendizado.

### 2.2.2 Modelo do Estudante

Esse modelo é a chave para o ensino individualizado e inteligente. Sendo esse então o componente mais importante e complexo de um STI.

A caracterização principal desse modelo deve contemplar todos os aspectos do conhecimento e do comportamento do estudante que tragam consequências para o seu desempenho e aprendizagem. Infelizmente, a construção desses modelos é vista ser uma tarefa bastante complexa para um sistema

computadorizado, pois os canais de comunicação entre o sistema e o estudante são bastante restritos comparados com a capacidade das pessoas de combinar informações em uma grande variedade de meios, como por exemplo o tom de voz e a expressão facial. Dessa forma, ainda hoje, a construção de tais modelos é um desafio para os sistemas computacionais.

Algumas características comuns devem estar presentes em qualquer que seja a forma que se realize a modelagem : o modelo deve ser dinâmico, contendo o conhecimento e as capacidades do estudante, seu comportamento de aprendizado passado, os métodos de apresentação para os quais ele responde melhor e sua área de interesse dentro do domínio [Chaiben].

Um sem número de pesquisas vêm sendo desenvolvidas na criação de novas formas de modelagem de estudantes em STI. Algumas dessas principais técnicas são apresentas com detalhes no capítulo seguinte.

### **2.2.3 Modelo Pedagógico**

Num processo convencional de ensino, o professor realiza a comunicação do conhecimento ao estudante guiado por estratégias e técnicas que são selecionadas e combinadas dinamicamente em reação às atitudes e necessidades dos estudantes [Chaiben], não seguindo simplesmente um protocolo de transferência de informações como acontece entre dois computadores.

A tarefa do modelo pedagógico é, então, bastante complexa. Este modelo contém o conhecimento necessário para tomar decisões sobre quais táticas de

ensino serão utilizadas dentre aquelas presentes no sistema, [Burns, Parlett e Redfield, 90]. Essas decisões são tomadas levando-se em consideração as diferenças existentes entre o modelo de domínio e o modelo do estudante. De maneira geral, as decisões são sobre qual informação apresentar ao estudante, quando e como apresentá-la.

A adaptação da instrução implica em uma escolha didática, que pode ser global ou local. Escolhas no nível global determinam a sequência dos episódios educacionais e, por exemplo, decisões sobre o número de repetições a serem utilizadas. Escolhas no nível local envolvem decisões sobre quando uma intervenção é necessária, se o estudante deve ou não ser interrompido em sua atividade, e o que pode e deve ser feito ou apresentado em um determinado momento. Isto inclui orientação no desempenho das atividades, explicações dos fenômenos e processos e decisões sobre quais informações serão oferecidas para reparar as deficiências dos estudantes [Wenger, 87].

Várias são as estratégias pedagógicas utilizadas atualmente em STIs. Porém, cada sistema geralmente implementa uma única estratégia. Isso decorre, em parte, da complexidade de entendimento e implementação dessas estratégias. Além disso, as estratégias pedagógicas são, em geral, dependentes do contexto geral do ambiente de aprendizagem embutido no sistema. A escolha do ambiente de ensino é ditada pela natureza do conteúdo a ser ensinado, pelo conhecimento e experiência do estudante e pelas hipóteses sobre aprendizagem inerentes à teoria básica sobre a qual o sistema está baseado.

As decisões pedagógicas tomadas pelo sistema são feitas dentro do con-

texto de um ambiente educacional que determina o grau de controle sobre a atividade e sobre a interação possuídos respectivamente pelo sistema tutor e pelo estudante [Wenger, 87]. Dessa forma, tais decisões dependem de uma série de fatores. Um preocupação adicional que se deve ter na implementação de uma dada estratégia pedagógica é o cuidado para não destruir a motivação pessoal do estudante ou o seu senso de descobrimento.

#### **2.2.4 Interface**

De maneira geral, as pesquisas de modelos de interfaces aqui se confundem com as pesquisas na área de interação homem-máquina. A meta da maioria dessas pesquisas é, de um modo ou de outro, proporcionar ao usuário um alto grau de eficiência com relação à utilização do sistema. Outra característica desejável é a de que o usuário não deve ter que se adaptar ao sistema; ao contrário, a interface deve ser projetada para que seja intuitiva e natural para ele aprender a utilizá-la [Burns, Parlett e Redfield, 90].

Falando estritamente sobre a interface em sistemas tutores, sua função é cuidar da forma final na apresentação de um conteúdo. Também é de sua responsabilidade controlar o fluxo de comunicação entre o sistema e o estudante e realizar a tradução entre a representação interna do sistema e a linguagem de interface de maneira compreensível ao estudante.

Como se percebe, a interface opera em estreita cooperação tanto com o diagnóstico do estudante como com o modelo pedagógico, no entanto as

decisões realizadas pela interface são de natureza distinta, requerendo um tipo diferente de conhecimento. Dessa forma, é interessante identificar a interface como um componente distinto [Chaiben]. Sua importância prática pode ser compreendida em dois níveis :

1. Uma vez decidida a forma na qual o sistema apresentará um determinado tópico, a interface tem o poder de determinar o grau de compreensão desta apresentação junto ao estudante.
2. Devido ao grande progresso da pesquisa em tecnologias gráficas, novas ferramentas para a construção e prototipação de interfaces podem direcionar todo o projeto do sistema.

Percebe-se que pesquisas nessa área têm se utilizado de estudos no campo da IA, mais intensivamente em processamento da linguagem natural. Muitos avanços já foram obtidos nesse campo, porém várias são as restrições ainda impostas pela utilização de linguagem natural em tais sistemas [Marietto].

Finalmente, é importante enfatizar que, em uma interação com um STI, o estudante não irá somente aprender o conteúdo das lições, mas também terá que aprender como utilizar o sistema tutor, portanto a facilidade de utilização deve ser uma das considerações principais no projeto dessas interfaces, diminuindo ao máximo possível a carga cognitiva sobre o estudante.

## 2.3 Exemplos de STIs

Nos últimos anos, as pesquisas em Sistemas Tutores Inteligentes têm produzido uma série de ferramentas de ensino com características bem diferentes. Uma descrição um pouco mais detalhada sobre algumas dessas ferramentas nos dá a exata noção das dificuldades referentes ao desenvolvimento de tal natureza.

A seguir serão descritos alguns dos esforços historicamente importantes no desenvolvimento de Sistemas Tutores Inteligentes. A maioria destes STIs tem sido extensivamente documentados, como por exemplo em [Wenger, 87].

### 2.3.1 SCHOLAR

O programa SCHOLAR [Carbonell, 70] foi o primeiro a tentar incluir a modelagem do conteúdo. O conhecimento do sistema é representado em uma rede semântica cujos nodos significam objetos e conceitos geográficos sobre a América do Sul. Esta rede contém uma série de elementos ligados por relações claramente especificadas, onde, por exemplo, o Brasil é um sub-conceito de país localizado na América do Sul.

Carbonell achava que uma rede semântica completa podia ser utilizada para modelar o conhecimento do estudante. Assim, ele sugeriu uma rede completa para modelar o "estudante perfeito" e progressivamente perturbá-lo para refletir o seu desempenho até aquele momento, apagando e até mesmo

modificando nodos e ligações. O modelo do estudante no sistema SCHOLAR, em termos de avaliações ligadas aos conceitos individuais de uma representação modular do conhecimento especialista, constitui uma primeira versão do que foi posteriormente chamado o modelo *overlay*, que será detalhado no próximo capítulo.

Um detalhe interessante neste sistema é que o estudante pode alterar a qualquer momento o modo de controle para o diálogo (teclando "Q/A"). Assim, o estudante pode solicitar ao sistema simplesmente responder suas questões, conduzir um diálogo de iniciativa mista, ou assumir o comando e fazer perguntas. Por exemplo, o estudante pode perguntar "Qual é a latitude do Brasil?" e o sistema irá localizar o elemento Brasil, e então a relação latitude, e apresentará o valor do elemento ligado com esta relação. Portanto, o sistema pode determinar a relação semântica entre dois nodos, simplesmente seguindo seus respectivos caminhos na hierarquia até um nodo em comum ser encontrado.

O sistema SCHOLAR representou uma grande descoberta nesta área. Com seu conhecimento de domínio representado como uma rede semântica, o sistema faz uso de procedimentos de inferência para uma interação tutorial simples. Usualmente os tutores de iniciativa mista dependem da fácil interação do sistema com o estudante através da linguagem natural.

### 2.3.2 SOPHIE

SOPHIE (*SOPHisticated Instructional Environment*) é um sistema ICAI desenvolvido por John Seely Brown, Richard Burton, e seus colegas na Bolt Beranek and Newman, Inc. [Brown, 75], para explorar a iniciativa do estudante durante uma interação tutorial. A meta do projeto SOPHIE era criar um ambiente de aprendizagem na qual os estudantes seriam desafiados a explorar idéias sobre suas próprias conjecturas ou hipóteses em situações de resolução de problemas.

Ao contrário do SCHOLAR, seu principal esquema representacional para o conhecimento do domínio é um modelo de simulação, ao invés de uma rede semântica. O programa apresenta ao estudante a simulação de uma parte de um equipamento eletrônico com defeito. O estudante deve diagnosticar o problema fornecendo as medidas adequadas ou formulando algumas questões específicas. O sistema é projetado para responder questões hipotéticas sobre o sistema sendo simulado, e também avaliar hipóteses. O SOPHIE proporciona ao estudante um ambiente de aprendizagem no qual ele adquire técnicas para a resolução de problemas experimentando suas idéias, ao invés de somente expor o material a ser ensinado. O sistema tem um modelo do conhecimento para resolução de problemas em seu domínio, assim como numerosas estratégias heurísticas para responder às questões dos estudantes, criticar suas hipóteses, e sugerir teorias alternativas. O SOPHIE permite que os estudantes tenham uma relação um-para-um com o especialista baseado em computador que os auxilia no surgimento de suas próprias idéias, experimentando estas idéias e depurando-as quando necessário.

Assim como o SCHOLAR, o projeto SOPHIE também foi marcante para estudos futuros na área, gerando uma longa e diversificada linha de pesquisa.

### 2.3.3 WEST

O projeto WEST [Burton, 79] foi iniciado no contexto do SOPHIE. Uma vez que o domínio da eletrônica parecia muito complexo para uma primeira investigação da arte de treinar, o domínio escolhido para este propósito foi o jogo educacional para computador chamado "How the WEST was won", desenvolvido para o projeto *PLATO Elementary Mathematics Project*. Este domínio estava de acordo portanto com o conceito de ambiente de aprendizagem reativo. O propósito do jogo é exercitar técnicas de aritmética.

Para acompanhar todas as necessidades do programa, determinar o que o estudante conhece, quando interromper, e o que dizer, Burton e Brown adicionaram um componente ao sistema e o denominaram *Coach*, porque ele agia como um "treinador", observando e auxiliando o jogador a melhorar. O *Coach* desenvolvido para o WEST baseia-se na idéia de *Issues* e *Examples*. *Issues* são conceitos usados no processo de diagnóstico para identificar, a qualquer momento, o que é relevante. *Examples* são instâncias concretas destes conceitos abstratos. O fundamento é que a melhor maneira de aumentar a técnica do estudante é ilustrando a ajuda educacional com exemplos concretos.

Há três níveis de *Issues* : habilidades matemáticas, regras e estratégias

particulares ao WEST, e habilidades relevantes para jogar, tal como a aprendizagem a partir do oponente. A maioria das informações sobre as quais o *Coach* baseia suas decisões vêm da comparação das ações do jogador com o que o especialista faria nas mesmas circunstâncias. Portanto, o WEST contém não somente uma representação do conhecimento do especialista, mas também técnicas tutoriais para utilizar a informação produtivamente.

A influência do WEST na área foi muito significativa, e o sistema é ainda hoje uma referência aos pesquisadores. Entretanto, o tutor em si foi utilizado somente durante alguns meses de experimentos preliminares. Isto pode ser devido ao custo do equipamento em que ele foi originalmente desenvolvido.

#### 2.3.4 GUIDON

O sistema GUIDON é um sistema tutor especialista para o ensino de diagnóstico de doenças infecciosas do sangue [Clancey, 87a], que foi desenvolvido a partir da base de conhecimento já formada do MYCIN, talvez o mais antigo e conhecido sistema especialista, e cuja concepção original foi inspirada pelas capacidades de diálogo do sistema SCHOLAR.

Os sistemas especialistas parecem oferecer uma base ideal para a construção de programas tutores. Além do fato óbvio de apresentarem grande quantidade de conhecimento especialista, uma outra vantagem é a usual separação da base de conhecimento contendo as regras de produção do interpretador procedimental que as utiliza. Ainda que um sistema especialista tenha uma

boa capacidade de explicação, ele pode somente justificar suas ações passivamente. Para ser capaz de ativamente apresentar o conhecimento, um sistema tutor necessita de técnicas adicionais para selecionar o material educacional, ser sensível ao estudante, e conduzir uma interação de modo bastante eficaz [Wenger, 87]. Por isso, os resultados da pesquisa de Clancey não foram os desejados, embora tenham deixado importantes contribuições tanto para as pesquisas em sistemas especialistas quanto para os sistemas educacionais.

A estratégia da apresentação pedagógica adotada pelo GUIDON utiliza o método de caso: um diálogo de iniciativa mista concentra sobre casos específicos para transmitir o conhecimento do MYCIN aos estudantes em um contexto de resolução de problemas bastante realístico.

Os pesquisadores do GUIDON perceberam que o importante conhecimento estrutural (hierarquias de dados e hipóteses de diagnóstico) e o conhecimento estratégico (pesquisa do espaço de problema através de refinamentos *top-down*) estavam implícitos nas regras. Isto é, o conhecimento procedimental que ocasionava um bom desempenho na resolução de problemas em uma consulta ao MYCIN não estava disponível para propósitos de ensino. Para tornar este conhecimento explícito, um novo sistema foi desenvolvido, o NEOMYCIN, que separa as estratégias de diagnóstico do conhecimento do domínio e faz bom uso da organização hierárquica de dados e hipóteses. Clancey [Clancey, 87b] faz uma boa análise dos problemas enfrentados : "... ensino e explicação, nós reconhecemos, exigem diferentes demandas de um especialista que simplesmente resolve problemas. Um professor pode fazer analogias, ter múltiplas visões, e níveis de explicação que são desconhecidos

do MYCIN. Na construção do MYCIN, nós não tornamos explícito como um especialista organiza seu conhecimento, como ele recorda este conhecimento, e quais estratégias ele utiliza para problemas próximos ...”.

O projeto GUIDON era até então o único na tentativa de transformar um sistema especialista em um tutor inteligente. A principal contribuição da primeira versão deste sistema é a identificação e o tratamento separado de diferentes tipos de conhecimento que devem ser disponibilizados para um tutor funcionar de forma eficaz.

### 2.3.5 HYDRIVE

HYDRIVE -*HYDR*aulics *I*nteractive *V*ideo *E*xperience é um STI que incorpora multimídia para resolver problemas de sistema hidráulico de um avião F-15. Apesar do HYDRIVE utilizar um disco laser externo para suportar imagens de vídeo, o conteúdo do disco laser (seqüência de animações e imagens) é representado na base de conhecimento utilizada pelo sistema.

O propósito deste STI é dar instrução aos técnicos de vôo para a solução de problemas complexos. O sistema, quando necessário, pode selecionar um conteúdo de vídeo, a partir de um disco laser, e então apresentá-lo. As pessoas que estão sendo treinadas podem ver os componentes da aeronave em operação. O sistema também pode apresentar instruções realísticas utilizando vídeo dos pilotos da aeronave e mecânicos.

No HYDRIVE, o material multimídia é disponível para o sistema através

da codificação de uma representação do conteúdo na forma de regras na base de conhecimento. Quando uma seqüência em particular necessita ser mostrada, ela pode ser localizada através desta representação. Esta incorporação de vídeo no HYDRIVE satisfaz as exigências de um STI para proporcionar uma boa instrução aos funcionários sendo treinados. Os modelos definem o que o sistema deve fazer após a ação da pessoa em treinamento, e a resposta do sistema pode então ser interpretada como uma determinada meta de apresentação.

O ambiente de implementação do HYDRIVE consiste de três elementos distintos: C, IL (*Interface Language*), e Arity Prolog. A linguagem C passa informação entre a interface e a máquina de inferência. IL é uma linguagem orientada a eventos utilizada para criar interfaces complexas consistindo de apresentações gráficas de alta qualidade e com recursos de controle de vídeo. O Arity Prolog foi utilizado para criar a máquina de inferência do HYDRIVE - os modelos do sistema, do estudante, e educacional [Kaplan e Rock, 95].

Quando a implementação do sistema HYDRIVE começou, haviam poucas alternativas para a implementação de um STI como este. Atualmente existem novas opções para a criação de sofisticadas interfaces, pois a ferramenta escolhida para a criação de um STI deve simplificar este processo ao máximo.

## 2.4 Considerações Finais

Vários anos de pesquisa na área de desenvolvimento de Sistemas Tutores Inteligentes mostraram a complexidade dessa tarefa. Isso decorre do fato de que ensinar é uma atividade bastante complexa e projetar um STI requer uma grande compreensão das várias dimensões envolvidas nesse processo, compreensão essa ainda pouco clara entre nós. Alguns exemplos de projetos de STI de maior sucesso consumiram anos de dedicação de várias pessoas e pode-se dizer que poucos projetos saíram dos laboratórios de pesquisa [Chaiben].

Percebe-se ainda que a maioria dos sistemas tutores convencionais (CAI) comercialmente disponíveis atualmente não apresentam qualquer tipo de característica que se possa chamar "inteligente". No entanto, quase a totalidade dos sistemas de ensino utilizados em escolas ao redor do mundo atualmente são desse tipo. Logo, o desenvolvimento de STI tem sido exatamente direcionado para superar as deficiências apresentadas por esses sistemas, entre as quais podemos citar a estática da instrução, o fraco conteúdo do domínio e a curva de retenção, que tende a reduzir o processo a uma simples transferência de informação.

Como se percebe, ainda existem vários problemas a serem resolvidos no campo dos STIs. Esforços nesse sentido necessitam voltar-se para questões profundas sobre a natureza do conhecimento, a comunicação, a aprendizagem e a compreensão.

## Capítulo 3

# A Modelagem do Estudante em STIs

### 3.1 Introdução

Como visto anteriormente, a Modelagem do Estudante é a análise do conhecimento e comportamento do estudante e a geração de um modelo que expresse tais características.

O desenvolvimento do modelo do estudante é uma das principais preocupações no projeto de sistemas tutores inteligentes. Isso se deve ao fato de praticamente toda a "inteligência" desses sistemas estar relacionada à eficiência na criação desse modelo.

O modelo do estudante é visto ser a chave do comportamento individualizado do sistema tutor inteligente, pois a adaptabilidade das instruções em relação a cada estudante depende de como o conhecimento desse estudante foi modelado internamente.

Diversas abordagens têm sido propostas para o desenvolvimento de modelos do estudante. Algumas delas são resumidas abaixo.

### 3.2 Modelo Overlay

Esse tipo de modelo considera o conhecimento do estudante como um subconjunto do conhecimento do domínio (veja Figura 3.1) . Quando o estudante realiza ações que ilustram que ele sabe um elemento particular do domínio do conhecimento, o modelador então marca esse elemento no modelo *Overlay*

[Baffes, 94].

Modelos *Overlay* mais sofisticados incluem ainda um intervalo de valores que indica quanto daquele elemento é conhecido pelo estudante. Dessa forma, os elementos não marcados são considerados desconhecidos pelo estudante e são então utilizados como objetivos do ensino.

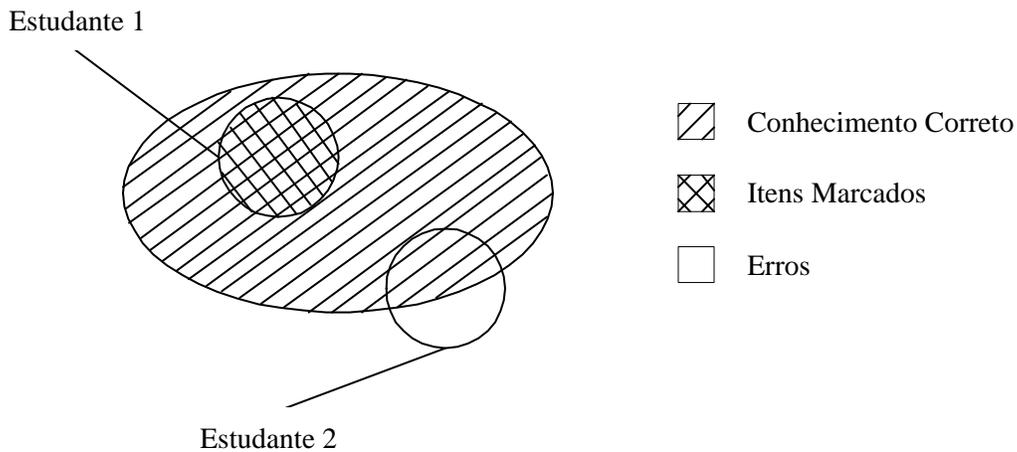


Figura 3.1: Exemplo de um Modelo Overlay.

Apesar de se mostrar bastante simples de implementar, essa técnica apresenta vários problemas. Um dos principais é o fato de não ser capaz de modelar os erros ou mal-entendidos cometidos pelo estudante, ou seja, tal modelo não captura a noção de falta de conhecimento do estudante.

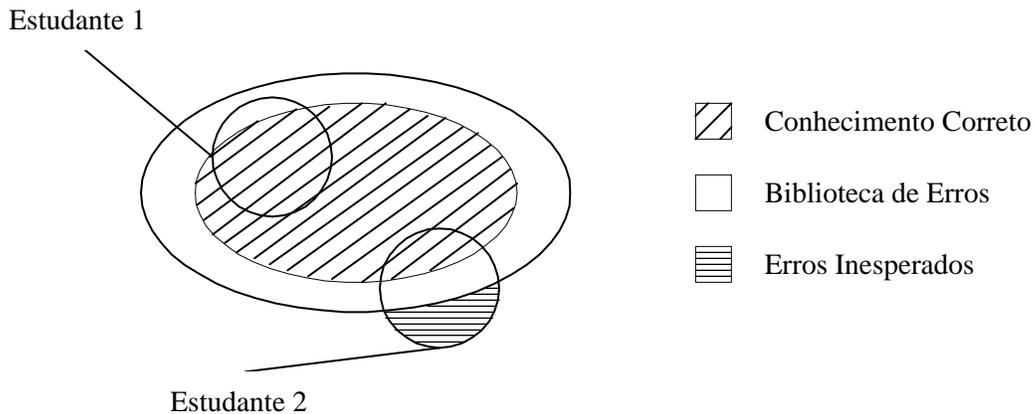


Figura 3.2: Exemplo de uma Biblioteca de Erros.

### 3.3 Biblioteca de Erros

Aqui, uma biblioteca de erros esperados é pré-construída manualmente. Essa biblioteca é gerada a partir de uma análise que determina os erros prováveis que os estudantes irão cometer durante sua interação com o sistema tutor. Dessa forma, quando um estudante comete algum desses erros, este é marcado na biblioteca de erros do mesmo modo como é marcado um conhecimentos correto no Modelo *Overlay*. Com isso, o sistema além de modelar o conhecimento correto do estudante, é também capaz de endereçar os erros cometidos por aquele estudante. Nesse modelo, o ensino é direcionado a partir dos erros marcados para cada estudante.

Apesar de essa extensão ao Modelo *Overlay* apresentar vantagens consideráveis em relação ao modelo original, percebe-se que dois problemas ainda

permanecem nessa técnica. Primeiramente, a construção dessa biblioteca de erros esperados é uma tarefa difícil e que consome muito tempo. Adicionalmente, mesmo que se tome grande cuidado na construção de tal biblioteca, não se pode garantir que ela cobrirá todos os casos possíveis de erros [Baffes e Mooney, 96], como mostrado na Figura 3.2

### 3.4 Biblioteca de Erros Estendida

Esse outro método de construção de modelos do estudante focaliza exatamente a eliminação das falhas do método anterior. A idéia aqui é atualizar dinamicamente a biblioteca de erros de acordo com a interação do estudante com o sistema no sentido de poder modelar todo e qualquer erro cometido por esse estudante, mesmo que este não tenha sido previsto inicialmente (veja Figura 3.3).

Essa abordagem de atualização da biblioteca de erros usa uma técnica analítica baseada no mecanismo fundamental de como mal-entendidos podem ser organizados para gerar extensões candidatas à biblioteca de erros [Sleeman et al., 90]. Esses candidatos são então apresentados ao autor do sistema tutor, que determina quais novos erros devem ser incorporados à biblioteca.

A grande desvantagem que se pode perceber nessa abordagem é que o sistema ainda depende do autor para determinar quais novos erros devem ser adicionados à biblioteca. Tal dependência impossibilita a atualização *on-line*

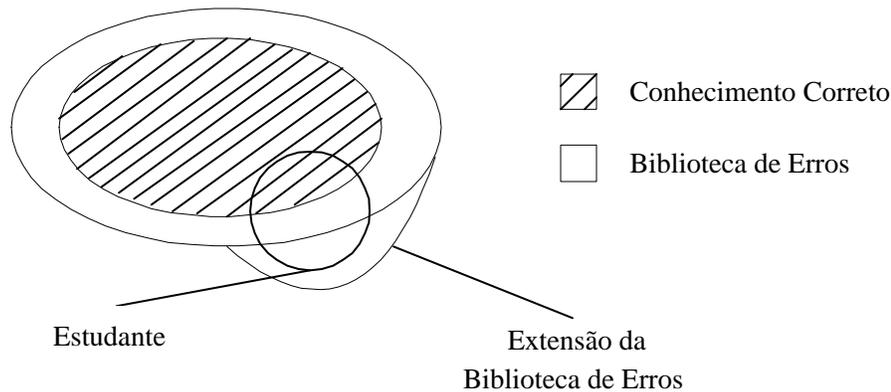


Figura 3.3: Exemplo de uma Biblioteca de Erros Estendida.

do modelo do estudante, dificultando sua utilização prática.

### 3.5 Modelagem Indutiva

Essa outra técnica de modelagem do estudante tem como objetivo principal acabar com a dependência do sistema em relação ao seu autor. Para isso foi proposta a utilização da Indução.

A Indução é uma técnica de aprendizagem de máquina utilizada para gerar hipóteses a partir de um conjunto de exemplos de treinamento. Tal hipótese deve ser coerente com o conjunto de treinamento, ou seja, ela deve incorporar as características desses dados [Michell, 97].

No contexto da modelagem de estudantes em STIs, essa técnica é utilizada

para moldar elementos do modelo do estudante a partir de exemplos do comportamento desse estudante. Iniciando sem qualquer noção preconcebida do conhecimento do estudante, o sistema observa como o estudante resolve os problemas propostos e constrói um modelo completo para esse estudante, incluindo tanto o conhecimento correto como o incorreto.

A maior desvantagem desse método é o fato de que a indução tipicamente requer um número grande de exemplos para produzir resultados precisos [Michell, 97]. Nessa aplicação, isso implica que, para que se possa gerar um modelo que efetivamente expresse o conhecimento de um dado estudante, é necessário que se tenha uma grande quantidade de informações acerca desse estudante. Para isso, necessita-se que a interação do aluno com o sistema já esteja acontecendo a um tempo razoavelmente grande. Essa restrição limita consideravelmente a utilização dessa técnica em sistemas reais.

### 3.6 Teoria da Revisão Proposicional

Inicialmente proposto por Baffes [Baffes, 94], o uso da teoria da revisão como ferramenta para modelagem de estudantes tem sido considerado. Sendo uma técnica de aprendizagem de máquina, a Teoria da Revisão surge nesse contexto exatamente para suprir as necessidades dos sistemas tutores de gerar modelos de estudantes a partir de um conjunto pequeno de informações acerca desse estudante. A forma de utilização da teoria da revisão como ferramenta de geração de modelos de estudante em STIs será detalhadamente

apresentada e discutida posteriormente.

Infelizmente, o uso de regras proposicionais limita consideravelmente a aplicabilidade desse método em sistemas reais, pois se verifica que tais sistemas são primariamente restritos à realização de tarefas de classificação de exemplos descritos a partir de vetores de características [Richards e Mooney].

Apesar de modelos de estudantes gerados a partir da teoria da revisão proposicional apresentarem a limitação discutida anteriormente, pode-se considerar os trabalhos de Baffes como predecessores conceituais da pesquisa contida nessa dissertação.

### 3.7 Considerações Finais

O desenvolvimento de um modelo computacional que possa expressar o conhecimento de um dado estudante é visto ser um dos maiores desafios dentro do ramo da Inteligência Artificial aplicada à Educação. Um grande número de pesquisas já foram desenvolvidas nesse sentido. No entanto, ainda não existe um consenso de qual técnica de modelagem é ideal de forma geral. O que se percebe é que uma dada técnica é melhor utilizada em um caso particular e outra técnica em outro caso.

Dessa forma, ainda existe uma extrema necessidade de que novas pesquisas possam apresentar resultados relevantes no desenvolvimento de tais modelos.

## Capítulo 4

### O Sistema de Revisão

## 4.1 Introdução

O sistema FORTE (*First-Order Revision of Theories from Examples*) [Richards e Mooney] tem como objetivo realizar a revisão de teorias de primeira ordem sob os princípios da teoria da revisão, ou seja, dada uma teoria inicial incorreta e um conjunto consistente de instâncias positivas e negativas, o sistema gera uma teoria revisada mínima que é coerente com as instâncias dadas. FORTE considera como uma teoria um programa Prolog puro.

## 4.2 O Algoritmo de Revisão

O sistema FORTE revisa teorias iterativamente, usando uma técnica *hill-climbing*. Cada iteração identifica pontos na teoria, chamados pontos de revisão, onde uma revisão tem o potencial de aumentar a precisão da teoria. Ele então gera um conjunto de revisões, baseado nos pontos de revisões, seleciona a melhor e a implementa. Esse algoritmo pode ser visto na Figura 4.1.

Para gerar os pontos de revisão, a teoria atual é testada no conjunto de treinamento. O FORTE anota as provas falhas de instâncias positivas e as provas bem sucedidas de instâncias negativas. Dessas anotações ele identifica pontos na teoria para possíveis revisões. Cada ponto de revisão têm um potencial, definido como o acréscimo máximo na precisão na teoria

1. Repita
2. Gere pontos de revisão
3. Ordene pontos de revisão pelo potencial
4. Para cada ponto de revisão
5. Gere revisões
6. Atualize melhor revisão encontrada
7. Até potencial do próximo ponto de revisão seja menor
8. do que a pontuação da melhor revisão até então
9. Se a melhor revisão melhora a teoria
10. Implemente a melhor revisão
11. Até que nenhuma revisão melhore a teoria

Figura 4.1: Algoritmo de revisão do FORTE.

na qual resultaria de uma revisão daquele ponto.

O sistema então gera um conjunto de revisões propostas dos pontos de revisão, começando com os pontos que têm um potencial mais alto e seguindo para os mais baixos. Cada revisão recebe uma pontuação, na qual é o atual aumento na precisão da teoria que ele atinge. FORTE então guarda a melhor revisão gerada até então, onde essa revisão é a que apresenta maior acréscimo de precisão. O algoritmo pára de gerar revisões quando o potencial do próximo ponto de revisão é menor do que o atual acréscimo de precisão da melhor revisão gerada. Nesse ponto, a melhor revisão é implementada, e o algoritmo recomeça. Esse processo continua até que o sistema seja incapaz de gerar qualquer revisão que melhore a teoria.

### 4.3 Os Pontos de Revisão

Existem dois tipos de pontos de revisão : pontos de especialização e pontos de generalização. Esses pontos são identificados a partir de provas anotadas ou provas tentadas de instâncias mal-classificadas. Pontos na teoria onde provas de instâncias positivas falham são pontos onde a teoria precisa ser generalizada, e cláusulas usadas na prova de instâncias negativas são pontos onde a teoria deve possivelmente ser especializada. O número de diferentes instâncias que marcam um ponto particular representa o potencial desse ponto.

A geração de pontos de especialização é simplesmente o processo de marcação daquelas cláusulas que participam da prova de instâncias negativas.

No entanto, a geração de pontos de generalização é bem mais complexa devido à existência de três tipos de operadores de generalização. Alguns operadores de generalização são baseados em antecedentes, isso significa que sua revisão focaliza um antecedente em particular em uma determinada cláusula, alguns são baseados em cláusulas, e outros em predicados. Deve-se gerar pontos de revisão para cada um desses tipos de operadores. Contudo, todos esses pontos de revisão são gerados de anotações feitas de provas falhas de instâncias positivas.

## 4.4 Os Operadores de Revisão

O sistema FORTE se utiliza de seis operadores para realizar as mudanças necessárias na teoria. Esses operadores estão divididos em dois grupos :

- Operadores de Especialização

A especialização é feita quando uma cláusula é usada na prova de uma instância negativa. Existem duas maneiras de se processar essa especialização: apagando regras ou adicionando antecedentes. Isso é feito com o uso dos operadores abaixo :

- Operador Deleta-Regra : Essa é a maneira mais simples de se realizar a especialização de uma cláusula, ou seja, simplesmente apagando-a. No entanto, existem algumas restrições a esse processo. Primeiro, se a cláusula é o único caso base de um predicado recursivo ela não pode ser apagada. Segundo, se essa é a única cláusula para um conceito de alto nível, nós substituímos a cláusula apagada pela regra `concept :- fail`.
- Operador Adiciona-Antecedente : Outra forma de se realizar a especialização é pela adição de antecedentes para distinguir entre instâncias positivas e negativas. FORTE adiciona antecedentes a uma cláusula na tentativa de tornar todas as instâncias negativas impossíveis de se provar. Porém, se a adição desses antecedentes também faz algumas das instâncias positivas não prováveis, o sistema adiciona a cláusula especializada à teoria e começa nova-

mente com a cláusula original, procurando especializações alternativas que retenham as provas de outras instâncias positivas enquanto ainda eliminem as negativas. Esse processo continua até que se tenha um conjunto de cláusulas que retenha a possibilidade de prova de todas as instâncias positivas prováveis originais.

- Operadores de Generalização

A generalização é feita quando uma instância positiva não pode ser provada. São usados quatro operadores na generalização : Deleta-Antecedente, Adiciona-Regra, Identificação e Absorção.

- Operador Deleta-Antecedente : Em muitos casos, FORTE deve ser capaz de criar uma boa revisão simplesmente deletando antecedentes de cláusulas existentes. Para desenvolver uma revisão, o sistema generaliza a cláusula original para cobrir o maior número possível de instâncias positivas, sem permitir que instâncias negativas passem a ser provadas. O sistema então adiciona a cláusula generalizada à teoria. Se existem mais cláusulas positivas a serem cobertas, ele começa novamente com a cláusula original e repete o processo. O algoritmo pára quando todas as instâncias positivas listadas no ponto de revisão são prováveis ou quando o sistema não é capaz de generalizar a cláusula original de forma a permitir uma prova para qualquer uma das instâncias não prováveis remanescentes.
- Operador Adiciona-Regra : O operador Adiciona-Regra é um operador de generalização baseado em cláusula que desenvolve uma

ou mais novas versões de uma regra existente, enquanto deixa a regra original na teoria. Seu objetivo é criar uma nova regra que permita a prova de instâncias positivas que identificaram a regra original como um ponto a ser revisado. Essa nova regra é construída em dois passos: primeiro, o sistema cria uma regra generalizada contendo apenas alguns dos antecedentes essenciais para evitar que instâncias negativas sejam prováveis, não interferindo com a prova das positivas. Para fazer isso, o sistema copia a regra original, deleta os antecedentes aos quais sua deleção não permite que qualquer instância negativa seja provada, e deleta os antecedentes aos quais sua deleção permite que uma ou mais instâncias positivas não prováveis anteriormente sejam provadas. Isso é feito com o operador Deleta-Antecedente. Segundo, cria-se uma ou mais especializações dessa regra, as quais permitirão provas das instâncias positivas desejadas enquanto elimina as negativas. Isso é feito usando-se o operador Adiciona-Antecedente.

- Operador Identificação : Identificação é um operador baseado em predicado que tenta generalizar a teoria com a criação de uma nova regra para o predicado existente. Ele constrói uma nova regra para generalizar a definição de um antecedente que faz com que uma ou mais provas de instâncias positivas falhem. Em lugar de desenvolver a cláusula do nada, ele realiza uma resolução inversa usando duas regras existentes na teoria do domínio.
- Operador Absorção : Absorção é o complemento da identificação. Em lugar de construir uma nova cláusula para o predicado corres-

pondente a um antecedente falho, absorção procura uma cláusula existente onde os antecedentes de possível falha aconteçam, e na qual tenha cláusulas alternativas que permitirão as instâncias positivas falhas serem provadas.

## 4.5 Conclusões

Esse capítulo descreve uma técnica de revisão completamente automática de teorias de primeira-ordem. Sua implementação, o sistema FORTE [Richards e Mooney], usa um algoritmo *hill-climbing* com uma grande coleção de operadores de generalização e especialização, na tentativa de encontrar uma teoria revisada mínima que seja consistente com o conjunto de exemplos de treinamento.

Alguns experimentos [Richards e Mooney] demonstraram a habilidade do sistema em efetivamente revisar teorias corrompidas e produzir resultados mais precisos do que sistemas que se utilizam da indução pura.

## Capítulo 5

# Modelagem baseada na Teoria da Revisão

## 5.1 Introdução

Como pode ser percebido, o ramo de desenvolvimento de técnicas para modelagem de estudantes tem produzido uma grande variedade de abordagens. No entanto, verifica-se que ainda são necessários grandes esforços para que se possa gerar uma técnica capaz de englobar todas as características necessárias a uma boa técnica de modelagem.

## 5.2 O Uso da Teoria da Revisão

A teoria da revisão é uma técnica relativamente recente de aprendizagem de máquina que refina uma dada teoria a partir de exemplos. Tipicamente, essa técnica realiza a retirada de erros em teorias incompletas ou incorretas pela incorporação de exemplos corretos nessa teoria. No entanto, o processo de revisão é indiferente em relação ao fato de ser a teoria inicial correta ou não em qualquer sentido. Nesse contexto foi proposto o uso da teoria da revisão de forma inversa, ou seja, com o objetivo de introduzir erros na teoria [Baffes e Mooney, 92]

## 5.3 O Processo de Modelagem

A forma descrita acima é a maneira na qual essa técnica pode ser usada na modelagem de conhecimento de um estudante. Começando-se com um

conhecimento correto de um conjunto de conceitos (Modelo do Domínio), o modelador pode introduzir os erros cometidos pelo estudante de forma a tornar aquela teoria coerente com o comportamento e o conhecimento desse estudante.

Esse processo foi descrito inicialmente por Baffes [Baffes e Mooney, 96]. Seu sistema de modelagem, o ASSERT, se utiliza da teoria da revisão no desenvolvimento de modelos de estudantes a partir de teorias proposicionais.

## 5.4 Vantagens da Técnica

Os sistemas de modelagem de estudante desenvolvidos até hoje a partir de uma abordagem puramente indutiva têm apresentado bons resultados. No entanto, como própria característica na indução, tais sistemas necessitam de um número razoavelmente grande de exemplos para produzir modelos eficientes.

Com a teoria da revisão, pode-se gerar modelos eficientes bem mais rapidamente, basicamente pelo fato de a modelagem não ser iniciada a partir do nada. A idéia que suporta tal resultado é a seguinte: quando se está modelando um estudante que apresenta um nível razoável de conhecimento, percebe-se que, se iniciarmos o processo de modelagem a partir do modelo correto do domínio, o modelo ideal para esse estudante será alcançado de forma bem mais rápida do que se iniciarmos a modelagem a partir do nada, como nos sistemas que se utilizam da indução pura.

## 5.5 Considerações Finais

O uso da teoria da revisão se apresenta como uma boa solução para geração de modelos de estudantes eficientes. No entanto, percebe-se que uma série de restrições de representação são impostas pelo uso da lógica proposicional. Adicionalmente, a eficiência desse método está condicionada ao nível de conhecimento do estudante, pois, quanto maior for o grau de conhecimento inicial deste, mais rapidamente o modelo ideal será alcançado. Da mesma forma, se considerarmos um estudante com pouco conhecimento, os ganhos no tempo de modelagem serão irrelevantes em relação à utilização de sistemas de modelagem de indução pura.

## Capítulo 6

# O Uso da Lógica de Primeira Ordem

## 6.1 Introdução

A escolha da forma de representação de conhecimento no desenvolvimento de qualquer sistema inteligente é de vital importância. Percebe-se que uma boa representação é capaz de garantir uma série de vantagens, podendo facilitar a captura rápida de informações e a geração de novos conhecimentos a partir dos já existentes.

Dentro do contexto do desenvolvimento de sistemas tutores inteligentes, a forma de representação influencia tanto na modelagem do conhecimento correto (Modelo do Domínio) como na modelagem dos estudantes. A forma de representação utilizada deve possibilitar uma captura rápida de certo conhecimento contido no modelo do domínio, de forma que esse possa ser apresentado ao estudante. Em relação aos modelos dos estudantes, deve-se possuir uma técnica de representação capaz de dinamicamente atualizar esses modelos sem grandes custo computacionais.

## 6.2 O Uso da Lógica Proposicional

A lógica proposicional é uma forma de representação bastante atraente em relação a outras técnicas por ser bastantes simples de se lidar e por existir um procedimento de decisão para ela. Com essa lógica pode-se representar um enorme número de fatos da vida real e deduzir outros a partir dos já existentes. Uma regra produzida com a lógica proposicional pode ser vista

basicamente como um vetor de características que se associam a uma certa classe. Dessa forma, sistemas de desenvolvimento a partir dessa técnica mostram-se primordialmente como sistemas de classificação. Assim, o desenvolvimento de sistemas tutores inteligentes ficam restritos a esse tipo de problema. Além disso, se desejarmos produzir modelos com estruturas mais complexas, uma série de dificuldades serão encontradas se utilizarmos a lógica proposicional.

O sistema de modelagem de estudantes ASSERT [Baffes e Mooney, 96] utiliza a lógica proposicional como forma de representação tanto do modelo do conhecimento como do modelo dos estudantes. Tal característica tem restringido sua utilização em sistemas reais a problemas de classificação.

### 6.3 O Uso da Lógica de Primeira Ordem

É nesse contexto que surge a utilização da Lógica de Primeira Ordem como forma de representação tanto do Modelo do Domínio como do próprio Modelo do Estudante. Nesse trabalho, é proposta a utilização de um sistema de revisão de teorias de primeira ordem chamado FORTE [Richards e Mooney].

O uso dessa técnica de representação habilita o desenvolvimento de sistemas tutores inteligentes para vários outros propósitos diferentes dos simples problemas de classificação. Estruturas mais complexas podem agora ser facilmente representadas. Considere o seguinte caso : deseja-se produzir um sistema tutor inteligente para ensinar o domínio de Educação Moral e Cívica.

Sem o uso de regras de primeira ordem, dificilmente se poderia modelar fatos simples como: "Todo país tem uma bandeira".

## 6.4 Vantagens Obtidas

A utilização da teoria da revisão de primeira ordem como modeladora do estudante em sistemas tutores inteligentes apresenta uma série de vantagens em relação ao uso da lógica proposicional como forma de representação. Podemos destacar as seguintes:

- Representação Robusta :

A utilização da lógica de primeira ordem traz vantagens iminentes em relação à representação proposicional. O principal problema com essa forma de representação é o fato de que sistemas desenvolvidos com lógica proposicional serem primariamente restritos à tarefa de classificação de exemplos descritos como vetores de características, restringindo a utilização de muitas outras estruturas.

- Aproveitamento de Teorias Existentes :

Uma das principais vantagens de utilização do sistema FORTE, além deste possuir uma forma de representação mais robusta, decorre do fato de que ele considera como teoria um programa Prolog puro. Dentro do nosso contexto de desenvolvimento de sistemas tutores, isso pode ser visto como uma imensa vantagem. Isso porque sabe-se que uma grande

quantidade de sistemas inteligentes desenvolvidos até hoje foram produzidos com a utilização da linguagem Prolog. Dessa forma, pode-se facilmente encontrar na literatura modelos lógicos de representação de um grande número de domínios diferentes. Isso possibilita o desenvolvimento de sistemas tutores com maior rapidez, visto que o Modelo do Domínio pode ser aproveitado diretamente, eliminando um dos maiores obstáculos no desenvolvimento de qualquer sistema inteligente - a produção do modelo para o domínio desejado.

- Modelagem do Grupo :

Uma grande vantagem, até então não explorada, apresentada pelo uso da teoria da revisão é o fato de que, a partir da modelagem de alguns estudantes, o sistema pode processar esses modelos de forma a retirar os erros comuns e construir o que viria a ser um modelo daquele grupo. Esse modelo tanto pode ser usado em um processo de aprendizagem cooperativa como pode servir como a teoria inicial na modelagem de novos estudantes, de forma a aumentar a precisão da nova modelagem e sua rapidez de obtenção. Essa forma de utilização da técnica será detalhadamente discutida no capítulo posterior.

## 6.5 Considerações Finais

As restrições impostas pela utilização da lógica proposicional na aplicação da teoria da revisão na modelagem de estudantes em sistemas tutores in-

teligentes podem ser contornadas a partir do momento que a lógica de primeira ordem seja utilizada.

Diversas vantagens iminentes de representação de estruturas complexas de conhecimento são assim obtidas. Adicionalmente, o uso do sistema FORTE como ferramenta de revisão possibilita o aproveitamento de modelos previamente desenvolvidos.

# Capítulo 7

## Geração e Uso do Modelo do Grupo

## 7.1 Introdução

Como visto anteriormente, uma das principais vantagens na utilização da teoria da revisão na modelagem de estudantes é o fato de podermos gerar automaticamente modelos correspondentes ao conhecimento de um grupo de alunos, o Modelo do Grupo. A idéia aqui é colecionar dentro de um modelo único as características comuns apresentadas pelos membros desse grupo.

A habilidade de geração de modelos de grupo traz uma série de vantagens consideráveis. Esse modelo pode ser usado no desenvolvimento de uma abordagem coletiva de ensino. Além disso, a geração de novos modelos de estudantes individuais pode ser facilitada a partir do momento que possa ser utilizado o modelo de grupo apropriado para participar do processo de geração. Uma descrição mais detalhada dessas vantagens será apresentada posteriormente.

No entanto, para que o modelo do grupo possa ser utilizado de forma eficiente, ele deve realmente concentrar as características dos membros do grupo. Para isso, um processo de geração eficiente para esse modelo é essencial. O objetivo é desenvolver modelos precisos de grupos de estudantes sem grandes custos computacionais, pois, como esse será usado na geração de modelos de estudantes individuais, o fato de o processo necessitar de uma grande quantidade de tempo pode prejudicar a eficiência educacional do método.

## 7.2 Geração do Modelo do Grupo

O algoritmo mostrado na Figura 7.1 apresenta o procedimento de geração do modelo do grupo.

```
1. Inicio
2.    $ErrosComuns \leftarrow SeleccionaErrosComuns(G)$ 
3.    $ModeloGrupo \leftarrow Revisa(ModeloDominio, ErrosComuns)$ 
4. Fim
```

Figura 7.1: Algoritmo de Geração do Modelo do Grupo.

Percebe-se que tal procedimento é bastante simples. Inicialmente, a função  $SeleccionaErrosComuns(G)$  obtém algumas regras que serão usadas no desenvolvimento do modelo do grupo. O objetivo de tal função é conseguir de modelagens anteriores erros comuns cometidos pelos estudantes pertencentes ao grupo  $G$ . Essa função, na prática, pode cumprir seu objetivo de diversas formas diferentes, a serem discutidas posteriormente. Essas regras comuns são atribuídas finalmente à variável  $ErrosComuns$ . Posteriormente, o sistema de revisão incorpora as regras pertencentes a  $ErrosComuns$  ao modelo do domínio. Essa revisão é realizada com o auxílio do sistema descrito no capítulo anterior. O resultado dessa revisão será então o modelo do grupo, armazenado aqui em  $ModeloGrupo$ .

Como apresentado anteriormente, o objetivo da função  $SeleccionaErrosComuns(G)$ , ou seja, a escolha das regras que serão uti-

lizadas no desenvolvimento do modelo do grupo  $G$  pode ser feita na prática de diversas formas :

- A utilização de uma dada regra, se essa regra foi utilizada na modelagem de todos os estudantes do grupo  $G$ . Essas regras podem ser facilmente obtidas checando-se as bases de regras dos estudantes do grupo e retirando-se as comuns.
- No entanto, a ocorrências de erros comuns a todos os estudantes, ou seja, a presença de uma mesma regra em todas as bases de erros, pode não ocorrer frequentemente. Dessa forma, pode-se determinar um parâmetro de ocorrências para a utilização de uma regra, ou seja, um certo percentual de ocorrência mínima seria determinado. Quando uma certo erro ocorrer com uma frequência maior do que esse percentual mínimo, essa regra seria utilizada.

### 7.3 Modelagem a partir do Modelo do Grupo

Desenvolvido o modelo do grupo como descrito anteriormente, pode-se utilizá-lo na geração de modelos individuais de novos estudantes. Esse processo visa o desenvolvimento de modelos mais precisos e de forma rápida.

A idéia aqui é utilizar o modelo do grupo como teoria inicial no nosso processo de modelagem. O objetivo fundamental dessa técnica é aproximar o modelo inicial da teoria revisada ótima, fazendo com que essa última se-

ja alcançada mais rapidamente. Para isso, em vez de utilizar o modelo do domínio (modelo correto) como teoria inicial na modelagem de novos estudantes, será utilizado um modelo de grupo que apresenta características gerais semelhantes àqueles provavelmente apresentadas pelo estudante que se deseja modelar. Dessa forma, a revisão irá convergir de forma mais rápida para a teoria revisada ótima.

Essa caracterização preliminar que determinará o modelo do grupo a ser utilizada pode ser feita considerando-se uma série de fatores, tais como a idade do estudante, tipo de educação anterior, grau de escolaridade, região de origem, ou seja, qualquer característica que possa aproximar o modelo do grupo do estudante que será modelado.

O processo de modelagem de novos estudantes a partir do modelo do grupo é descrito abaixo com mais detalhes.

### **7.3.1 O Algoritmo de Modelagem**

O algoritmo de modelagem de novos estudantes a partir do modelo do grupo é bastante semelhante àquele desenvolvido a partir do modelo do domínio. A diferença básica apresentada é que o processo se inicia a partir do modelo do grupo que apresenta maiores semelhanças com o estudante a ser modelado e não a partir do modelo do domínio. O algoritmo de modelagem pode ser visto de forma detanhada na Figura 7.2.

Como descrito anteriormente, esse processo se inicia a partir do Modelo

```
1.  Início
2.    TeoriaAtual ← ModeloGrupo
3.  Faça
4.    RegrasAtuais ← PegaRegras()
5.    TeoriaAtual ← Revisa(TeoriaAtual, RegrasAtuais)
6.  Até CondicaoParada
7.    ModeloEstudante ← TeoriaAtual
8.  Fim
```

Figura 7.2: Algoritmo de Modelagem do Estudante a partir do Modelo do Grupo.

do Grupo. Posteriormente, o algoritmo apresenta um laço responsável pela revisão da *TeoriaAtual*, inicialmente setada para *ModeloDominio*. Nesse laço, verifica-se a existência de uma função chamada *PegaRegras()*. Essa função é responsável por obter as regras lógicas correspondentes as ações do estudante que se deseja modelar a partir de sua interação com o sistema tutor. Essas regras são atribuídas a variável *RegrasAtuais*. O próximo passo do algoritmo é o de revisão propriamente dita. Aqui, as regras existentes em *RegrasAtuais* são incorporadas a *TeoriaAtual* com o uso do sistema de revisão descrito anteriormente. O resultado dessa revisão é atribuído a variável *TeoriaAtual*. Novas regras podem ser obtidas e novamente a revisão será realizada. Esse laço termina assim que uma dada condição de parada seja satisfeita. Na prática, essa condição de parada pode ser implementada de diversas formas. Ela pode ser uma função do número de regras revisadas,

pode ser uma função do tempo de revisão ou pode ainda ser controlada pelo usuário do sistema. Sendo essa condição verdadeira, o processo de revisão é finalizado setando-se o *ModeloEstudante* como sendo a *TeoriaAtual*.

### 7.3.2 Vantagens da Técnica

A utilização de um modelo de grupo no desenvolvimento de novos modelos de estudantes oferece uma série de vantagens em relação as técnicas de modelagens propostas anteriormente, dentre as quais pode-se destacar :

- Modelagem Rápida e Eficiente :

Com o uso de um modelo de grupo que apresente características semelhantes àquelas esperadas para um dado estudante, percebe-se que a convergência para o modelo ideal desse estudante é mais rápida, pois algumas das suas características já estão incorporadas. Essa vantagem, na prática, faz com que o sistema tutor possa instruir de forma individualiza mais rapidamente, aprimorando o processo de ensino.

- Instrução Coletiva :

Dentro de uma visão do grupo de estudantes, o modelo do grupo pode ser utilizado para determinar as deficiências gerais daquele grupo e conseqüentemente estar apto a gerar instruções correspondentes a essas deficiências, de forma a satisfazer, de uma só vez, as necessidades de um maior número de estudantes possível.

## 7.4 Conclusões

Uma das maiores dificuldades apresentadas pelos pesquisadores na área de modelagem do estudante é o fato de as técnicas atuais necessitarem de um tempo relativamente grande de interação do estudante com o sistema para que tal sistema possa gerar um modelo ótimo para esse estudante. Essa característica tem impossibilitado a implementação de várias técnicas dentro de sistemas reais.

O desenvolvimento de um modelo de grupo pode facilitar ainda mais a criação de modelos individuais de novos estudantes. Felizmente, o desenvolvimento de tal modelo é visto ser computacionalmente simples e rápido. Outra facilidade encontrada é o fato de nenhum tipo de técnica adicional ser necessária, pois a criação do modelo do grupo se utiliza também da teoria da revisão, ou seja, do mesmo sistema de revisão que a modelagem individual.

Os ganhos de precisão obtidos a partir dessa técnica podem ser facilmente verificados com testes experimentais. Tais testes são apresentados no capítulo seguinte.

## Capítulo 8

### Estruturação do Conteúdo

## 8.1 Motivação

A utilização da lógica de primeira ordem como ferramenta de representação tanto do Modelo do Domínio como do Modelo do Estudante apresenta uma série de vantagens principalmente no que se refere a manipulação dessas informações. No entanto, a visualização e manipulação dessa informação enquanto um conteúdo programático de uma disciplina se torna bastante dificultada pela ausência de uma estruturação desse conhecimento.

A partir de um conteúdo estruturado, o sistema pode com maior facilidade determinar, de acordo com o conhecimento que já apresenta o estudante, qual será o próximo conteúdo a ser apresentado a este. Adicionalmente, foi proposta anteriormente a apresentação da avaliação realizada pelo sistema ao educador. Essa visualização, como se sabe, se torna bastante complexo se partirmos de uma representação com a utilização da lógica, requerendo assim um conhecimento computacional prévio do educador que deseja trabalhar com tal conteúdo, o que é visto ser uma situação altamente indesejada no desenvolvimento de um sistema de ensino.

Nesse sentido é proposta a utilização de uma técnica de representação de conhecimento conhecida com Redes Semânticas na estruturação do conhecimento armazenado no Modelo do Domínio. De forma resumida, pode-se destacar as seguintes motivações para tal estruturação :

- Facilitar a visualização da avaliação feita pelo sistema de forma que o educador possa compreendê-la sem que qualquer tipo de conhecimento

computacional prévio seja requerido.

- Organizar o conteúdo a ser ensinado de forma a facilitar a escolha do próximo conteúdo que será apresentado ao estudante de acordo com seu conhecimento e comportamento anterior.

## 8.2 Redes Semânticas

Semântica é o estudo do significado de conceitos individuais utilizados na linguagem. É uma tentativa de descrever os significados das palavras e as condições sob as quais eles podem interagir para serem compatíveis com outros aspectos de uma linguagem.

Uma rede é um conjunto ou um grafo de nós conectados por ligações, como na Figura 8.1. Os nós em uma rede semântica usualmente representam os conceitos ou significados, por exemplo, "POTENCIAL DE MEMBRANA", "PROCESSO ELETRO-QUÍMICO". As ligações usualmente representam as relações existentes entre estes nós, por exemplo, "O potencial de membrana É UM processo eletro-químico".

Portanto, uma rede semântica é uma estrutura para a representação do conhecimento definida como um padrão de nós interconectados por arcos rotulados. As redes deste tipo não só captam as definições dos conceitos mas também, inerentemente, proporcionam ligações com outros conceitos. Uma variedade de redes semânticas tem sido desenvolvidas como variações deste

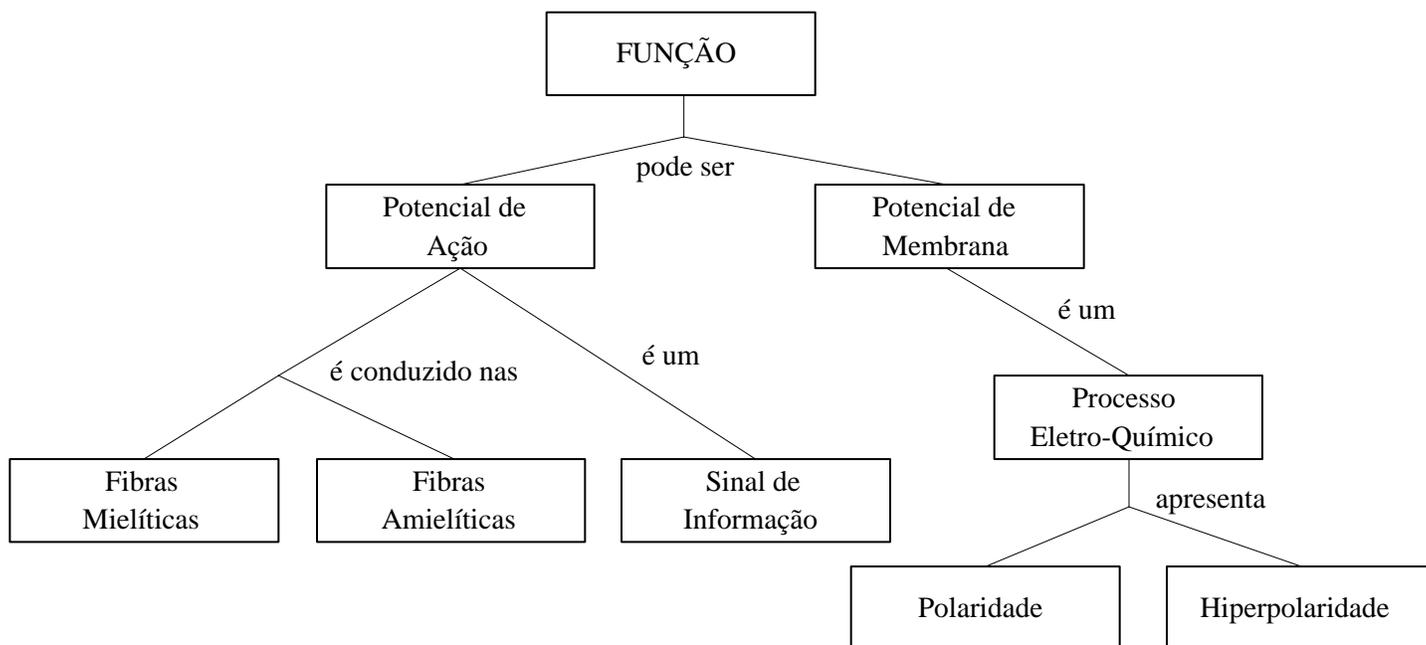


Figura 8.1: Rede Semântica.

simples padrão. Algumas são propostas como modelos da memória humana e significado de representação, enquanto outras são usadas como componentes de compreensão de linguagem e sistemas de raciocínio.

Embora existam algumas controvérsias quanto à denominação "redes semânticas", acredita-se que são assim chamadas por razões puramente históricas. Em seu modelo precursor "Semantic Memory", Quillian [QUILLIAN 68] tentava representar o significado do Inglês, ou seja, a semântica das palavras em Inglês. Ele queria definir um modelo de como o significado das palavras eram representados na memória humana, para isso, construiu uma rede associativa

de nós que representavam relações entre os conceitos. Portanto, a denominação de "redes semânticas", dizia respeito à aplicação destas redes, e não à alguma característica especial das mesmas.

Independentemente das razões que sugeriram esta denominação, as origens das redes semânticas vêm do associacionismo de Aristóteles - onde o comportamento é controlado totalmente por associações aprendidas entre os conceitos - e do reducionismo - onde os conceitos são sempre constituídos de conceitos mais elementares [RINGLAND 88].

A teoria das redes semânticas afirma que [Chaiben]:

- A memória humana é associativa por natureza, e as idéias e informações na memória estão interrelacionadas umas com as outras.
- Esta interrelação entre as idéias pode ser retratada pelos nós e ligações rotuladas que conectam estes nós podendo ser representados por algumas estruturas formais e precisas.
- Sob o ponto de vista cognitivo das teorias gerais de organização da memória, com relação ao sistema de Quillian, a rede semântica é organizada de acordo com os princípios de significado distribuído, economia cognitiva e do raciocínio *default*. O princípio do significado distribuído refere-se ao fato de que a determinação do significado de um simples conceito é encontrado considerando-se sua interação com o conjunto inteiro de conceitos na rede semântica. O princípio da economia cognitiva refere-se ao fato de que, se possível, os atributos comuns aos vários

conceitos não são codificados especificamente com cada conceito, mas com um conceito supra-ordenado. O princípio do raciocínio *default* refere-se ao fato de que, qualquer verdade assumida junto ao conceito supra-ordenado pode ser assumida verdade para seu conceito subordinado.

Embora exista uma grande variedade de terminologias e notações nos sistemas de representação do conhecimento por redes semânticas, as seguintes considerações são comuns na maioria delas [SOWA 91]:

- Nós na rede representam conceitos de entidades, atributos, eventos e estados.
- Arcos na rede geralmente chamados de relações conceituais, representam as relações mantidas entre os conceitos. Os rótulos sobre os arcos especificam os tipos de relações.
- Algumas relações conceituais representam casos de lingüística, como agente, paciente, receptor ou instrumento. Outros, representam conectivos espaciais, temporais, causais, lógicos. Ainda outros especificam o papel de uma entidade com relação à outro, como por exemplo, mãe, proprietário, residência, etc.
- Os tipos de conceitos são organizados em uma hierarquia de acordo com os níveis de generalidade. Esta hierarquia é freqüentemente chamada de hierarquia de tipos ou hierarquia taxionômica.

- Relações mantidas entre todos os conceitos de um determinado tipo são herdadas através da hierarquia por todos os seus sub-tipos. Por exemplo, se todo animal necessita de oxigênio para sobreviver, então esta característica será herdada pelos carnívoros, felinos, gatos, etc.

## 8.3 Estruturação com Redes Semânticas

A idéia proposta nessa dissertação é a utilização de uma rede semântica no sentido de estruturar o conhecimento a ser ensinado ao estudante, ou seja, o conhecimento contido no Modelo do Domínio.

### 8.3.1 Inicialização da Rede

Quando do desenvolvimento do Modelo do Domínio pelo desenvolvedor do sistema tutor, deve-se adicionalmente desenvolver a rede semântica correspondente ao conteúdo contido nesse modelo. A rede inicial deve ser criada sob os seguintes princípios :

- Cada tópico do conteúdo a ser ensinado será representado por um nó na rede semântica.
- O relacionamento entre os nós representará a relação de pré-requisito, ou seja, se possuímos em uma rede semântica um nó inferior No ligado

a um nó superior  $N_1$ , temos que o conteúdo contido em  $N_0$  é dependente daquele contido em  $N_1$ . Esse tipo de relação permite ao sistema determinar a ordem de apresentação do conteúdo programático.

- Dessa forma, o primeiro nível da rede (nível superior) determina o conhecimento que deve possuir aquele estudante quando de seu ingresso ao sistema. Analogamente, o último nível (nível inferior) apresenta o conhecimento que será obtido pelo estudante ao final do processo de ensino.

### 8.3.2 Modificação da Rede

Cada aluno que interage com o sistema deve possuir uma rede semântica própria que apresentará dinamicamente o conteúdo que é conhecido naquele momento por esse estudante. Isso permite ao próprio sistema e até mesmo ao educador acompanhar a qualquer momento o desenvolvimento desse estudante.

Para isso, o sistema deve atualizar a rede de forma dinâmica. Isso pode ser feito da seguinte forma :

1. Inicialmente os nós da rede são apresentados sem qualquer cor.
2. Sempre que um novo conhecimento é apresentado ao estudante e o sistema checa se o estudante absorveu tal conhecimento, o sistema deve

localizar a rede semântica desse estudante e colorir o nó da rede referente ao tópico em questão.

### 8.3.3 Uso e Apresentação da Rede Final

Com a rede atualizada dinamicamente, o sistema tem a possibilidade de verificar qual novo tópico deve ser apresentado ao estudante. Para isso, basta ao sistema verificar na rede quais são os tópicos já conhecidos pelo estudante e, seguindo as relações de pré-requisitos apresentadas na rede, verificar quais novos conhecimentos são passíveis de serem apresentados nesse momento.

No que se refere a apresentação da avaliação ao educador responsável pelo acompanhamento do processo de ensino, basta ao sistema apresentar graficamente a rede atualizada de um dado estudante para que o educador tenha a exata noção de como seu processo de aprendizado está evoluindo.

## 8.4 Conclusões

A utilização da tecnologia de Redes Semânticas permite, sem grandes custos computacionais, a estruturação do conhecimento armazenado no sistema tutor, visto que tal conhecimento é inerentemente estruturado. Com tal estruturação, tanto o próprio sistema tutor como o educador responsável pela avaliação dos estudantes têm a capacidade de, a qualquer momento, verificar o desenvolvimento do aprendizado de qualquer estudante e, dessa forma,

planejar a continuidade de processo de ensino.

## Capítulo 9

# Resultados Experimentais

## 9.1 Introdução

Acredita-se que a utilização da teoria da revisão possa trazer uma grande quantidade de ganhos no desenvolvimento de modelos de estudantes. Tal crença vem do fato de que um número de exemplos de treinamento bem menor do que o necessário no processo puramente indutivo possa produzir modelos bastante precisos. Adicionalmente, a criação de um modelo de grupo pode, e deve, ainda mais aprimorar tal modelagem. No entanto, tais resultados previstos devem ser checados.

Os experimentados apresentados nesse capítulo têm como objetivo mostrar a eficiência da técnica apresentada na modelagem de estudantes e grupo através de simulações. Tais simulações visam "imitar" a utilização de um sistema tutor inteligente por um grupo de estudantes. A partir de tais simulações, são gerados modelos de conhecimento, de acordo com o método de revisão apresentado nesse trabalho, correspondentes a cada estudante e ao grupo. Posteriormente, foram verificadas as precisões correspondentes a cada modelo.

No sentido de se poder comparar a eficiência do método apresentado nessa dissertação em relação ao método puramente indutivo, vários testes foram realizados também com essa técnica.

Os experimentos foram realizados com o uso de uma versão sistema FORTE [Richards e Mooney] desenvolvida para o compilador Quintus Prolog. Foi usada uma versão Shareware do Quintus Prolog 3.4 para Windows

95, 98 e NT.

## 9.2 O Modelo do Domínio

Nesses experimentos, foi utilizado o Modelo dos Relacionamentos da Família simplificado, apresentado inicialmente por [Quinlan, 90]. Tal modelo foi escolhido por se tratar de uma base de regras bastante utilizada na vida prática e assim de fácil entendimento. Esse modelo pode ser visto na Figura 9.1.

## 9.3 A Geração dos Erros dos Estudantes

Como visto anteriormente, a geração dos modelos dos estudantes é feita incorporando-se os erros cometidos por esses estudante durante sua interação com o sistema tutor. Dessa forma, para que se possa realizar a simulação desse processo, um conjunto de erros aleatórios foi gerado automaticamente. Esse erros, num total de 2590, foram gerados da seguinte forma : Inicialmente, algumas alterações foram realizadas nas regras do modelo dos relacionamentos da família. Essas alterações foram feitas para que se pudesse gerar um conjunto de erros coerentes entre si. As modificações realizadas no modelo correto são as apresentadas da Figura 9.2.

A partir desse modelo alterado e de uma base de fatos que representa uma família exemplo, um novo conjunto de fatos, na maioria errados, foi obtido.

```
esposa(A, B) :- sexo(A, feminino), casado(A, B).
marido(A, B) :- sexo(A, masculino), casado(A, B).
mae(A, B) :- sexo(A, feminino), pai-mae(A, B).
pai(A, B) :- sexo(A, masculino), pai-mae(A, B).
filha(A, B) :- sexo(A, feminino), pai-mae(B, A).
filho(A, B) :- sexo(A, masculino), pai-mae(B, A).
irma(A, B) :- sexo(A, feminino), irmaos(A, B).
irmao(A, B) :- sexo(A, masculino), irmaos(A, B).
tia(A, B) :- sexo(A, feminino), tio-tia(A, B).
tio(A, B) :- sexo(A, masculino), tio-tia(A, B).
sobrinha(A, B) :- sexo(A, feminino), tio-tia(B, A).
sobrinho(A, B) :- sexo(A, masculino), tio-tia(B, A).
tio-tia(A, B) :- casado(A, C), irmaos(C, D), pai-mae(D, B).
tio-tia(A, B) :- irmaos(A, C), pai-mae(C, B).
irmaos(A, B) :- pai-mae(C, A), pai-mae(C, B), A ≠ B.
primos(A, B) :- pai-mae(C, A), pai-mae(D, B), irmaos(C, D).
```

Figura 9.1: Modelo dos Relacionamento da Família.

Desse conjunto são retirados aleatoriamente os erros de cada estudante para serem incorporados ao modelo correto, gerando assim o modelo desse estudante. Da mesma forma, tais erros são utilizados na criação do modelo de um estudante indutivamente, podendo assim esse ser comparado ao primeiro.

```
pai(A, B) :- sexo(A, feminino), pai-mae(A, B).
irma(A, B) :- sexo(A, feminino), irmaos(B, A).
tia(A, B) :- sexo(A, feminino), tio-tia(A, B), irmaos(A, B).
tio-tia(A, B) :- casado(A, C), irmaos(E, D), pai-mae(B, B).
irmaos(A, B) :- pai-mae(C, A), pai-mae(C, B).
```

Figura 9.2: Alterações feitas no Modelo dos Relacionamentos da Família.

## 9.4 A Modelagem dos Estudantes a partir do Modelo do Domínio

Para a geração dos modelos de um estudante, um certo subconjunto aleatório foi retirado do conjunto total de erros. Esses erros tentam simular os erros reais cometidos por esse estudante. Esse conjunto será chamado aqui de conjunto de treinamento. Posteriormente, esses erros são incorporados ao modelo correto gerando-se assim o modelo daquele estudante. Esse modelo gerado é testado sob o conjunto de erros restantes, chamado conjunto de teste, para que se possa medir sua precisão.

Foram utilizados conjuntos de treinamento de tamanhos que variam de 10 até 300 erros. Para cada conjunto, foram feitos 5 testes. As médias das precisões obtidas nesses 5 testes são as precisões consideradas. Os resultados das precisões podem ser vistas na Tabela 9.1.

Pode-se verificar na tabela que, como esperado, a precisão obtida na mo-

| Num. de Erros Incorporados | Precisão (%) |
|----------------------------|--------------|
| 0                          | 86,86        |
| 10                         | 85,01        |
| 20                         | 90,37        |
| 30                         | 97,68        |
| 50                         | 97,66        |
| 75                         | 97,69        |
| 100                        | 97,8         |
| 150                        | 97,57        |
| 200                        | 97,74        |
| 300                        | 97,73        |

Tabela 9.1: Precisão dos Modelos gerados a partir do Modelo do Domínio.

delagem com um conjunto de treinamento pequeno é relativamente boa por ser considerado o modelo do domínio como modelo inicial. No entanto, esse resultado depende do grau de conhecimento do estudante a ser modelado. No exemplo, foi considerado um estudante que apresenta uma quantidade pequena de erros, logo, as precisões iniciais dos modelos desse estudante são muito boas. Adicionalmente, também como esperado, verifica-se que a precisão do modelo do estudante aumenta de acordo com o aumento do número de exemplos (erros) incorporados ao modelo inicial.

| Num. de Erros Incorporados | Precisão (%) |
|----------------------------|--------------|
| 0                          | 57,45        |
| 10                         | 65,61        |
| 20                         | 74,47        |
| 30                         | 77,97        |
| 50                         | 82,79        |
| 75                         | 90,88        |
| 100                        | 96,63        |
| 150                        | 95,9         |
| 200                        | 96,13        |
| 300                        | 96,83        |

Tabela 9.2: Precisão dos Modelos gerados indutivamente.

## 9.5 A Modelagem dos Estudantes Indutiva

Para a geração dos modelos de um estudante a partir do nada, foi utilizado o mesmo sistema FORTE, no entanto, a revisão foi realizada a partir de uma teoria vazia, o que simula a modelagem puramente indutiva. Da mesma forma que anteriormente, subconjuntos do conjunto total de erros são retirados aleatoriamente e utilizados na indução do modelo do estudante. Analogamente, o conjunto de erros restante foi utilizado como conjunto de teste. Além disso, aqui também foram utilizados conjuntos de treinamento de tamanhos variando entre 10 até 300 e a precisão dos modelos gerados foi obtida pela médias de 5 testes. Os resultados são os mostrados na Tabela 9.2.

A precisão obtida nesse caso não é muito boa quando se considera um conjunto de treinamento pequeno. No entanto, com o aumento desse conjunto de treinamento, a precisão do modelo gerado é incrementado substancialmente.

## 9.6 A Geração do Modelo do Grupo

Para que se conseguisse gerar um modelo referente ao conhecimento do grupo, um certo número de erros foram considerados como sendo os erros comuns cometidos pelos membros desse grupo. Com esses erros comuns em mãos, basta incorporá-los ao modelo do domínio para que se obtenha o modelo de conhecimento desse grupo.

## 9.7 A Modelagem dos Estudantes a partir do Modelo do Grupo

Pode-se utilizar o modelo do grupo como base inicial na geração de novos modelos de estudantes que apresentem conhecimentos similares aos membros desse grupo. Para isso, retira-se do conjunto de erros um subconjunto que será considerado como os erros cometidos por esse novo estudante que se deseja modelar. Posteriormente, esses erros são incorporados agora ao modelo do grupo e não ao modelo do domínio. Os resultados são apresentados na Tabela 9.3.

| Num. de Erros Incorporados | Precisão (%) |
|----------------------------|--------------|
| 0                          | 86,88        |
| 10                         | 90,42        |
| 20                         | 93,04        |
| 30                         | 97,65        |
| 50                         | 97,02        |
| 75                         | 97,69        |
| 100                        | 97,13        |
| 150                        | 97,68        |
| 200                        | 97,61        |
| 300                        | 97,63        |

Tabela 9.3: Precisão dos Modelos gerados a partir do Modelo do Grupo.

Como visto, os resultados obtidos a partir de um conjunto de erros pequeno é muito bom. Esse resultado já era o esperado, pois se está partindo de um modelo (modelo do grupo) que apresenta características semelhantes aquelas apresentadas pelo estudante, fazendo com que o processo de geração do modelo do novo estudante se torne mais rápido e produza melhores resultados.

## 9.8 Comparação dos Resultados

A Figura 9.3 mostra os gráficos dos desenvolvimentos das precisões na geração de modelos de estudantes a partir do modelo do domínio, do modelo do grupo

e do nada (processo indutivo) de acordo com o número de erros utilizados como conjunto de treinamento.

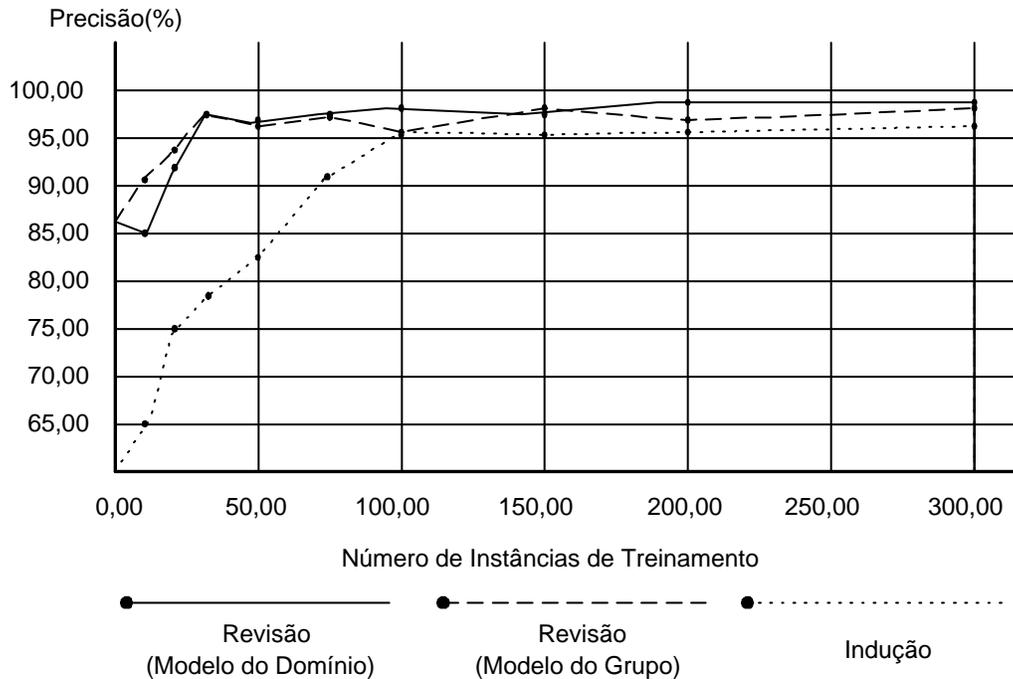


Figura 9.3: Resultados Comparativos.

Como pode ser observado, consegue-se gerar modelos de estudantes bastante precisos a partir de um conjunto pequeno de erros do estudante com a utilização da teoria da revisão de primeira ordem, tanto a partir do modelo do domínio como teoria inicial assim como com o modelo do grupo, apesar de as precisões apresentadas com a utilização do modelo do domínio dependerem do conhecimento do estudante modelado.

O uso do modelo do grupo propicia a geração de modelos melhores em relação ao processo que utiliza o modelo do domínio como teoria inicial quando é utilizado um conjunto de erros pequeno. No entanto, quando o número de erros utilizados no treinamento aumenta, os dois métodos geram modelos bastante semelhantes.

Em relação a utilização do modelo puramente indutivo, como era esperado, percebe-se que os resultados obtidos a partir de um pequeno conjunto de erros são bem inferiores aos obtidos com a teoria da revisão. Esse resultado mostra-se bastante interessante pois na prática o número de informações disponíveis para a modelagem de um estudante é geralmente pequeno. Esse é exatamente o resultado fundamental que justifica a utilização dessa técnica na modelagem de estudantes. Mesmo com um número bastante grande de exemplos de erros, percebe-se que o modelo gerado indutivamente nunca apresenta uma precisão superior ao modelo gerado pela revisão com um mesmo número de erros.

## 9.9 Conclusões

Os resultados obtidos através das simulações mostram a eficiência computacional do método proposto na modelagem de estudantes quando o número de exemplos de treinamento ( erros cometidos pelo estudante ) é pequeno. A simulação do método puramente indutivo mostrou sua inabilidade na modelagem a partir de poucos dados. Tal resultado desabilita esse método na

prática, pois o conjunto de treinamento nessa caso é geralmente pequeno.

O uso do modelo do grupo, por apresentar resultados animadores e por sua eficiência não depender do conhecimento do estudante, é visto ser o método ideal para o desenvolvimento de modelos de estudantes precisos a partir de um pequeno número de exemplos de treinamento.

## Capítulo 10

### Conclusões e Trabalhos Futuros

## 10.1 Trabalhos Futuros

A partir das pesquisas realizadas nesse trabalho, pode-se verificar a existência de uma grande variedade de perspectivas de novas pesquisas complementares. Algumas propostas de estudos complementares são detalhadas abaixo.

### 10.1.1 Testes Pedagógicos

O propósito de qualquer técnica de modelagem de estudantes é evidentemente oferecer um melhor desempenho ao sistema tutor em sua tarefa de ensinar. Em outras palavras, podemos considerar como propósito principal o melhoramento do aprendizado do aluno.

A abordagem dos testes mostrados nessa dissertação é puramente computacional. Acredita-se que ganhos computacionais, principalmente referentes a precisão do modelo do estudante gerado e sua rapidez de criação, proporcionem consideráveis ganhos educacionais. Os testes demonstraram a eficiência computacional da técnica proposta, no entanto, seria relevante considerar os ganhos pedagógicos efetivos obtidos.

Nesse sentido, o desenvolvimento de testes pedagógicos poderiam justificar o desenvolvimento de sistemas reais com a técnica proposta. A idéia fundamental de tais testes seria observar o desenvolvimento educacional de dois grupos de estudantes : um alimentado de informações por um sistema de ensino convencional e outro por um sistema tutor inteligente que incorporasse

a técnica de modelagem descrita nesse trabalho.

### **10.1.2 Técnicas de Representação mais Robustas**

O uso da teoria da revisão como modeladora do estudante traz uma série de vantagens em relação a outras técnicas. A utilização da lógica de primeira ordem como técnica de representação, adicionalmente, possibilita a modelagem de muitas estruturas.

No entanto, sabe-se que ainda existem uma série de situações no mundo real que necessitam de representações diferenciadas para serem modeladas. Dessa forma, o desenvolvimento de um sistema que possibilitasse a revisão de teorias representadas por outros métodos poderia oferecer grandes contribuições. Tal fato ofereceria uma maior gama de opções de desenvolvimento de sistemas tutores, dependendo da forma de conhecimento particular.

### **10.1.3 Desenvolvimento de um Shell de STI**

O desenvolvimento de sistemas tutores inteligentes ainda esbarra em imensas barreiras em relação ao custo e ao tempo de desenvolvimento. Isso decorre basicamente do fato de que o desenvolvimento de tais sistemas exige um conhecimento multidisciplinar e, além disso, dificilmente se consegue reutilizar módulos desenvolvidos para outro sistema.

O desenvolvimento de um Shell de sistema tutor inteligente poderia am-

enizar muitos desses problemas. O objetivo seria criar um sistema sem o Módulo do Conhecimento, que seria incorporado posteriormente pelo desenvolvedor do novo sistema, de acordo com o domínio a ser ensinado. Tal sistema poderia incorporar um conjunto de técnicas pedagógicas, com características distintas, que poderiam ser escolhidos de acordo com a aplicação. O modelo de modelagem do estudante aqui proposto seria incorporado automaticamente, visto que sua utilização é completamente independente do domínio.

## 10.2 Conclusões

A principal contribuição da pesquisa apresentada nessa dissertação é uma nova abordagem de modelagem de estudantes em sistemas tutores inteligentes a partir de modelos de grupo.

Recentes debates sobre a utilização da modelagem de estudantes têm criticado tais modelos por serem muito difíceis de se construir e ainda assim não apresentarem ganhos educacionais consideráveis.

Para alcançar os objetivos desejados, um sistema de modelagem deve exibir duas características essenciais :

- Deve ser de fácil uso : Especificamente, um sistema deve ser capaz de desenvolver modelos de estudantes sem que seja requerido do autor do sistema tutor a incorporação de todos os possíveis erros que um

estudante possa cometer.

- Deve ser útil : ou seja, deve disponibilizar informações que possibilitem um significativo impacto na performance dos estudantes.

A abordagem de modelagem apresentado aqui apresenta as duas características. Primeiro, ele requer somente o modelo correto do domínio para construir seus modelos de estudantes. Todas as funcionalidades adicionais, como criação do modelo do grupo e modelagem a partir desse modelo, são processadas automaticamente, sem qualquer intervenção do autor do sistema. Além disso, o sistema proposto aqui executa essas tarefas de forma extremamente rápida, possibilitando sua utilização em sistemas interativos. Finalmente, os resultados experimentais apresentados anteriormente mostram que a modelagem de estudantes com o uso da técnica aqui proposta pode influenciar significativamente o desempenho do estudante.

Adicionalmente, a ganho obtido com a utilização da lógica de primeira ordem é inestimável, pois percebe-se que as restrições impostas por outras formas de representação utilizadas em sistemas tutores anteriores praticamente desabilitaram os ganhos esperados pelo processo de modelagem do estudante.

Essa pesquisa ilustra que o processo de geração de um componente de modelagem de estudantes em sistemas tutores inteligentes não é, essencialmente, uma tarefa complexa e que consome muito tempo e dinheiro. Visto que, com o uso dessa técnica, o único componente adicional que deve ser incorporado é o modelo correto do domínio. No entanto, como o sistema

se alimenta de teorias na forma de programas Prolog, acredita-se que essa tarefa pode ser ultrapassada pela reutilização de teorias já projetadas.

A possibilidade de desenvolvimento e utilização de um modelo de grupo é visto ser a maior contribuição apresentada nessa pesquisa. O fato de tal desenvolvimento e utilização não necessitarem de qualquer componente diferenciado daqueles necessários essencialmente habilita a sua utilização em sistemas reais. Outro aspecto importante é a possibilidade que esse modelo de grupo oferece de aplicação de uma abordagem grupal de ensino, na tentativa de endereçar as necessidades comuns entre os membros desse grupo.

Em relação ao desempenho apresentado nos testes, verifica-se que os ganhos computacionais conseguidos comparados com o método puramente indutivo mostram-se altamente relevantes quando consideramos um pequeno conjunto de treinamento (erros). Esse fato, na prática, possibilita a criação de modelos de estudantes precisos de forma mais rápida. Tal característica pode, e deve, influenciar no desempenho educacional do estudantes, pois este recebe o *feedback* do sistema tutor prontificadamente.

Finalmente, acredita-se que as pesquisas contidas nessa dissertação possam efetivamente colaborar com o ramo de aplicação da inteligência artificial na educação na geração de sistemas computacionais capazes de, não substituir, mas auxiliar professores na atividade fundamental de ensinar.

# Apêndice A

## Artigos

## **A.1 Artigo SBIE'99**

Esse artigo foi publicado nos anais do X Simpósio Brasileiro de Informática Educativa de 1999 realizado na cidade de Curitiba em Novembro de 1999.

## Desenvolvendo um Ambiente Distribuído de Auxílio à Avaliação Conceitual em Sistemas Tutores

JERFFESON TEIXEIRA DE SOUZA,

TARCÍSIO CAVALCANTE PEQUENO, MARCELINO CAVALCANTE PEQUENO

Laboratório de Inteligência Artificial (LIA),

Universidade Federal do Ceará (UFC),

Bloco 910 – Campus do Pici

60.455-760 - Fortaleza, CE – Brasil

**Resumo :** Esse artigo apresenta um ambiente distribuído de auxílio a avaliação de estudantes no uso de sistemas computacionais de ensino. Seu objetivo principal é realizar essa avaliação de forma distribuída e integrada, provendo ao educador informações essenciais na tomada de decisões pedagógicas. O ambiente se utiliza de uma técnica de aprendizagem de máquina chamada teoria da revisão para a modelagem dos estudantes. A arquitetura orientada a objetos distribuídos do ambiente permite sua utilização em projetos de educação a distância sobre qualquer tipo de rede, inclusive a Internet.

**Palavras-Chave :** Avaliação do Estudante, Modelagem do Estudante, Sistemas Tutores, Teoria da Revisão.

## 1. Introdução

Já há vários anos, pesquisadores de todo o mundo têm reconhecido no computador um grande potencial como ferramenta educacional. Dessa forma, vários esforços vêm sendo realizados na tentativa de desenvolvimento de sistemas computacionais de ensino que possam, efetivamente, colaborar com o processo ensino-aprendizado [Burns 90] [Marietto] [Richard]. No entanto, algumas décadas de pesquisa não foram suficientes para que se conseguisse gerar ferramentas realmente capazes de produzir instruções individualizadas e pedagogicamente eficientes, fato esse que em parte decorre da própria complexidade de entendimento e representação de estruturas pedagógicas.

Esse trabalho propõe o desenvolvimento de um ambiente distribuído de auxílio à avaliação conceitual em sistemas de ensino baseados em computador (Sistemas Tutores). Tal ambiente possibilita ao educador um acompanhamento eficiente do processo de aprendizagem dos alunos no uso de um determinado sistema tutor. Esse acompanhamento permite ao educador propor suas próprias estratégias pedagógicas coerentes com cada caso. A idéia principal é construir e atualizar um modelo do conhecimento de cada estudante e do grupo que possam simular seus aprendizados de acordo com a interação desses com o sistema. Para isso, propõe-se a utilização de uma técnica de aprendizagem de máquina conhecida como teoria da revisão [Baffes 92], que revisa automaticamente uma base de conhecimento tornando-a consistente com um conjunto de exemplos. Essa técnica é aqui projetada para usar o domínio do conhecimento correto do sistema tutor como domínio inicial de forma a torná-lo coerente com os exemplos provenientes das interações dos estudantes com o sistema. Nesse trabalho é utilizado o sistema FORTE

(*First-Order Revision of Theories from Examples*) [Richard] que implementa a teoria da revisão. Uma grande vantagem da utilização dessa técnica é a possibilidade de desenvolvimento de um modelo de conhecimento para o grupo, gerado a partir dos modelos dos estudantes individualmente. Tal modelo oferece ao educador uma visão geral do aprendizado de todo o grupo. Essa avaliação do desempenho, tanto de cada estudante, como do grupo, é então apresentada de forma amigável ao educador que pode tomar suas decisões pedagógicas.

## 2. Sistemas Tutores

O desenvolvimento de sistemas tutores iniciou-se com a proposição de um paradigma conhecido como CAI (*Computer Aided Instruction*) [Chainben]. Um programa CAI tem como objetivo apresentar automaticamente o conteúdo programático de certo domínio de conhecimento. Isso funcionou inicialmente como um livro eletrônico, com a vantagem de o usuário (estudante) poder escolher o nível de detalhes da apresentação desse conteúdo. Tal abordagem permitia, de uma certa forma, que o aluno pudesse ter o controle sobre a forma como o material seria apresentado. Uma grande restrição desse sistema é o fato de tais escolhas terem que ser pré-programadas pelo desenvolvedor, o que gerava um grande esforço na criação desses sistemas e ainda assim não se podia garantir que todas as necessidades dos estudantes seriam satisfeitas, fazendo com que várias críticas fossem feitas aos sistemas CAI.

A principal característica que se desejava incorporar em tais sistemas era a adaptabilidade desses em relação a cada estudante. Tal adaptabilidade

pode ser vista como a capacidade do sistema de tirar conclusões sobre as potencialidades e deficiências de cada aluno baseado na interação do sistema com este de forma a poder gerar instruções individualizadas, assim como é feito pelo professor no processo de ensino convencional.

Novos esforços se multiplicaram posteriormente no desenvolvimento de sistemas ICAI (*Intelligent CAI*), também chamados ITS (*Intelligent Tutoring Systems*) [Wenger 87] [Burns 90], que incorporam técnicas de inteligência artificial. Tais sistemas monitoram a performance do estudante e tentam descobrir o conhecimento que o estudante detém sobre o domínio ensinado. Esse modelo é então comparado com o modelo do domínio para se descobrir suas diferenças. Essa informação é repassada ao modelo pedagógico que decide qual, quando, e como a informação será transmitida ao estudante por uma interface.

### 3. Modelagem baseada na Teoria da Revisão

Uma série de pesquisas vêm sendo realizadas para o desenvolvimento de novas técnicas que possam ser usadas na construção de modelos do estudante eficientes. A teoria da revisão tem sido apresentada como uma técnica de aprendizagem de máquina que refina uma dada teoria a partir de exemplos, ou seja, retira os erros em teorias incompletas ou incorretas pela incorporação de exemplos corretos nessa teoria [Richard]. Contudo, em princípio, não existe razão pela qual não se possa usar essa técnica de forma oposta, ou seja, com o propósito de introduzir erros na teoria [Souza 99] [Baffes 94]. Essa é a forma na qual essa técnica pode ser usada na modelagem do conhecimento de um estudante. Começando-se com um conhecimento correto de um conjunto

de conceitos (Modelo do Domínio), o modelador pode introduzir os erros cometidos pelo estudante de forma a tornar aquela teoria coerente com o comportamento e o conhecimento deste. Uma grande vantagem dessa técnica é o fato de que a partir da modelagem de alguns estudantes, o sistema pode processar esses modelos de forma a retirar os erros comuns e construir o que viria a ser um modelo daquele grupo. Esse modelo tanto pode ser usado em um processo de aprendizagem cooperativa, como pode servir como a teoria inicial na modelagem de novos estudantes, de forma a aumentar a precisão da nova modelagem e sua rapidez de obtenção.

#### 4. O Sistema FORTE

O sistema FORTE (*First-Order Revision of Theories from Examples*) [Richard] tem como objetivo realizar a revisão de teorias de primeira-ordem sob os princípios da teoria da revisão, ou seja, dada uma teoria inicial incorreta e um conjunto consistente de instâncias positivas e negativas, o sistema gera uma teoria revisada mínima que é coerente com as instâncias dadas. O FORTE considera como uma teoria um programa Prolog puro.

O sistema FORTE revisa teorias iterativamente, usando uma técnica *hill-climbing*. Cada iteração identifica pontos na teoria, chamados *pontos de revisão*, onde uma revisão tem o potencial de aumentar a precisão da teoria. Baseado nos pontos de revisões o sistema gera um conjunto de revisões, seleciona a melhor e a implementa. Esse algoritmo pode ser visto na Figura A.1.

Para gerar os pontos de revisão, a teoria atual é testada no conjunto de treinamento. O FORTE anota as provas falhas de instâncias positivas

e as provas bem sucedidas de instâncias negativas. Dessas anotações, ele identifica pontos na teoria para possíveis revisões. Cada ponto de revisão têm um *potencial*, definido como o acréscimo máximo na precisão na teoria na qual resultaria de uma revisão daquele ponto.

1. Repita
2. Gere pontos de revisão
3. Ordene pontos de revisão pelo potencial
4. Para cada ponto de revisão
5. Gere revisões
6. Atualize melhor revisão encontrada
7. Até potencial do próximo ponto de revisão seja menor
8. do que a pontuação da melhor revisão até então
9. Se a melhor revisão melhora a teoria
10. Implemente a melhor revisão
11. Até que nenhuma revisão melhore a teoria

Figura A.1: Algoritmo de revisão do FORTE.

O sistema gera um conjunto de revisões propostas para cada ponto de revisão, começando com os pontos que têm um potencial mais alto e seguindo para os mais baixos. Cada revisão recebe uma pontuação, na qual é o atual aumento na precisão da teoria que ele atinge. O FORTE, então, guarda a melhor revisão gerada até o momento, onde essa revisão é a que apresenta maior acréscimo de precisão. O algoritmo pára de gerar revisões quando o potencial do próximo ponto de revisão é menor do que o atual acréscimo de precisão da melhor revisão gerada. Nesse ponto, a melhor revisão é imple-

mentada, e o algoritmo recomeça. Esse processo continua até que o sistema seja incapaz de gerar qualquer revisão que melhore a teoria.

## 5. O Ambiente

Percebe-se que a utilização do computador, como ferramenta de tomada de decisões pedagógicas, sofre uma série de restrições que praticamente impossibilitam um ensino eficiente. Dessa forma, pode ser observada a necessidade e as vantagens da utilização de um ambiente que permita ao educador acompanhar o desenvolvimento do aluno na utilização de um sistema computacional de ensino de forma a lhe proporcionar a tomada de decisões de modo eficiente.

Nesse contexto, esse artigo propõe um ambiente distribuído para auxílio à avaliação conceitual que possibilita ao educador a avaliação do desempenho de certo estudante no uso de um sistema tutor. Essa avaliação habilita o educador a tomar as decisões pedagógicas coerentes com cada aluno. Além disso, o mesmo sistema de avaliação pode ser projetado de forma a integrar essa avaliação em sistemas tutores diferentes.

Para modelar esse ambiente distribuído, foi escolhida a tecnologia de objetos distribuídos. Essa tecnologia permite a modelagem das partes do ambiente em forma de objetos e utiliza o padrão CORBA (*Common Object Request Broker Architecture*) do OMG [OMG 97] como meio de comunicação entre esses objetos.

O principal objetivo desse ambiente é permitir a avaliação dos estudantes de forma realmente distribuída e integrada. Desse modo, foram identificados dois módulos distintos para o ambiente: O Cliente de Avaliação, que funciona

como interface de cada sistema tutor com a plataforma de distribuição, e o Servidor de Avaliação, que integra e trabalha os dados colhidos dos Clientes de Avaliação. A Figura A.2 apresenta a forma com que esses módulos se relacionam no ambiente distribuído.

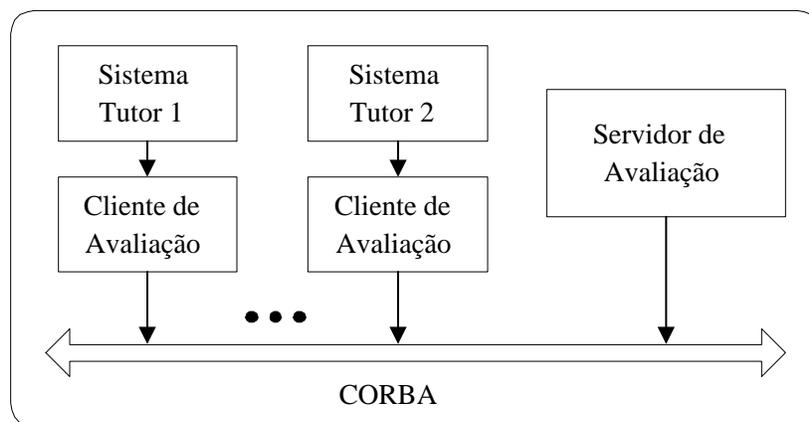


Figura A.2: Interfaces do Sistema de Avaliação.

Como pode ser observado na figura 2, as informações dos sistemas tutores são tratadas pelos Clientes de Avaliação, que as transformam em um formato comum a fim de serem integradas e correlacionadas com as informações dos demais sistemas tutores. Essas informações são, por sua vez, repassadas ao Servidor de Avaliação para serem tratadas de forma integrada.

## 5.1 O Cliente de Avaliação

Como citado anteriormente, o Cliente de Avaliação tem como principal função coletar os dados de utilização dos sistemas tutores pelos estudantes,

modelar essas informações em um formato comum, e repassar os dados modelados ao Servidor de Avaliação.

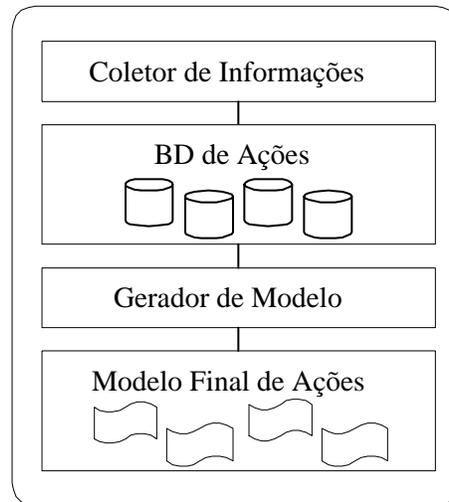


Figura A.3: Arquitetura do Cliente de Avaliação.

Para atingir esse objetivo, os Clientes de Avaliação foram modelados com a seguinte arquitetura (Figura A.3)

- Coletor de Informações : Esse módulo é utilizado para extrair as informações dos sistemas tutores e armazená-las em um banco de dados (Banco de Dados de Ações);
- Banco de Dados de Ações : As informações capturadas pelo Coletor de Informações são armazenadas nessa base de dados, em um formato conhecido pelo ambiente, a fim de ser tratado pelo Gerador de Modelo;

- Gerador de Modelo : Esse é o módulo mais importante dos Clientes de Avaliação. Ele é responsável por gerar o modelo comum do sistema de avaliação. A partir das informações do Banco de Dados de Ações, o Gerador de Modelo produz um modelo comum (cláusulas de Horn de Primeira Ordem) que será repassado para o Servidor de Avaliação e esse poderá tratar esses dados de forma realmente unificada;
- Modelo Final de Ações : Esse é o arquivo resultante da modelagem das informações de ações dos estudantes nos sistemas tutores. Ele é um arquivo texto comum que está em um formato reconhecido pelo Servidor de Avaliação e com as informações modeladas no formato comum do sistema.

## 5.2 O Servidor de Avaliação

O Servidor de Avaliação é responsável por armazenar e gerar os modelos tanto de cada estudante como do grupo, além de apresentar esses modelos ao educador. A arquitetura desse servidor pode ser vista abaixo na Figura A.4:

- Módulo de Comunicação : Esse módulo é responsável por receber os modelos finais de ações enviados pelos Clientes de Avaliação. Recebido um modelo, inicialmente o módulo de comunicação identifica a qual estudante esse se refere. Depois, ele retira as regras do modelo e as inclui na base de revisão correspondente àquele estudante no Banco de Dados de Revisão.

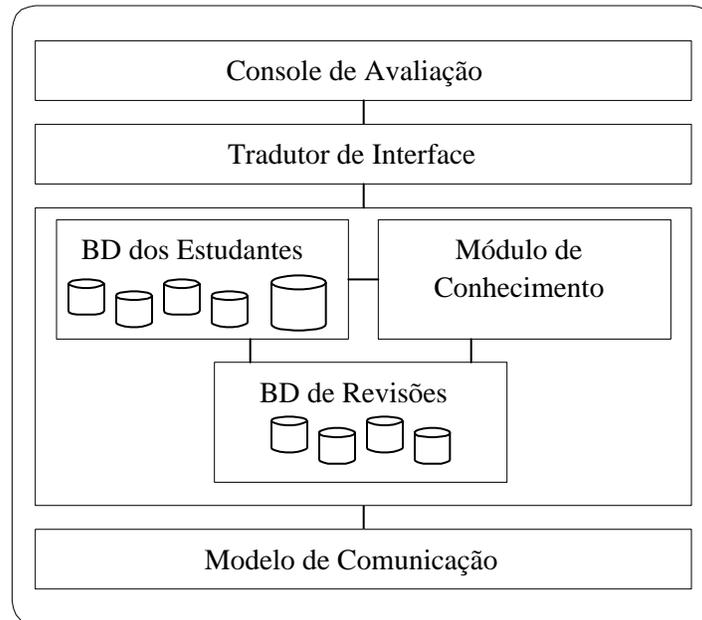


Figura A.4: Arquitetura do Servidor de Avaliação.

- Banco de Dados de Revisões : Nessa base de dados são armazenadas todas as Bases de Revisão de cada estudante. Uma base de revisão é uma tabela que guarda todas as regras referentes ao comportamento de cada estudante. Além disso, essa base indica se essa regra já foi ou não usada na revisão do modelo desse estudante. Além de guardar essas informações, esse módulo realiza atualizações periódicas nos modelos do estudante e do grupo, utilizando as regras que ainda não foram usadas na revisão desses modelos. Essa revisão é feita com a utilização do sistema FORTE.

- Módulo do Conhecimento : Nesse módulo estão armazenadas todas as informações do domínio que devem ser avaliadas.
- Banco de Dados dos Estudantes : Aqui são guardados os modelos, tanto de cada estudante, como do grupo.
- Console de Avaliação : Esse componente é responsável pela interação do sistema de avaliação com o educador. Seus objetivos principais são : permitir que o educador escolha a forma na qual a avaliação realizada pelo sistema será a ele apresentada e apresentar essa avaliação.
- Tradutor de Interface : Esse módulo é responsável pela conversão da forma de representação interna do sistema para a forma de apresentação da avaliação escolhida pelo educador.

## 6. Estudo de Caso

Nessa seção será apresentado um estudo de caso simples no sentido de possibilitar um melhor entendimento do ambiente de avaliação proposto. Nesse exemplo, o domínio considerado será o de relacionamentos na família. A teoria que define esse domínio pode ser vista abaixo. Essa é a teoria armazenada no Módulo de Conhecimento do Servidor de Avaliação.

Seja considerada a seguinte árvore genealógica como exemplo :

```

esposa(A, B) :- sexo(A, feminino), casado(A, B).
marido(A, B) :- sexo(A, masculino), casado(A, B).
mae(A, B) :- sexo(A, feminino), pai-mae(A, B).
pai(A, B) :- sexo(A, masculino), pai-mae(A, B).
filha(A, B) :- sexo(A, feminino), pai-mae(B, A).
filho(A, B) :- sexo(A, masculino), pai-mae(B, A).
irma(A, B) :- sexo(A, feminino), irmaos(A, B).
irmao(A, B) :- sexo(A, masculino), irmaos(A, B).
tia(A, B) :- sexo(A, feminino), tio-tia(A, B).
tio(A, B) :- sexo(A, masculino), tio-tia(A, B).
sobrinha(A, B) :- sexo(A, feminino), tio-tia(B, A).
sobrinho(A, B) :- sexo(A, masculino), tio-tia(B, A).
tio-tia(A, B) :- casado(A, C), irmaos(C, D), pai-mae(D, B).
tio-tia(A, B) :- irmaos(A, C), pai-mae(C, B).
irmaos(A, B) :- pai-mae(C, A), pai-mae(C, B), A B̄.
primos(A, B) :- pai-mae(C, A), pai-mae(D, B), irmaos(C, D).

```

```

          Marta |
              | |-- Susana
        /-- Marcos |
        /
Carine |                /-- Raquel
      | |----- Lorena |
Carlos |                | |
        \              Leo |
        \                \-- Paulo

```

\-- Andre

Suponha que existam vários sistemas tutores distribuídos em uma rede ensinando esse domínio. Um certo aluno em um dos sistemas recebe o seguinte exercício :

1) De acordo com a árvore genealógica acima determine o parentesco de Leo em relação à Susana :

a) Irmãos b) Sobrinho c) Tio d) Primo

E marca o item d) como correto.

Dessa forma, como a resposta está errada, o módulo Coletor de Informações do Cliente de Avaliação irá gravar essa ação do estudante em seu Banco de Dados de Ações. Posteriormente, essa informação será utilizada pelo Gerador de Modelo para a geração de uma ou mais regras de primeira ordem que expressem essa ação, nesse caso temos que a regra gerada será :

(+) primos(Leo, Susana) :- casado(Leo, Lorena),irmãos(Lorena, Marcos), pai\_mãe(Marcos, Susana).

O que mostra que o estudante considera como primo aquele que é na verdade tio\_tia. O sinal (+) indica que a regra é uma instância positiva que deve passar a ser provada na teoria.

Essa regra é então incluída no Modelo Final de Ações daquele aluno. O Cliente de Avaliação, por sua vez, repassa esse modelo para o Servidor de Avaliação através da plataforma de distribuição.

Ao receber esse modelo, o Módulo de Comunicação reconhece a qual estudante a regra se trata e realiza a inserção dessa regra no Banco de Dados de Revisões referente àquele estudante. Posteriormente, o sistema FORTE é utilizado para a geração do modelo de cada estudante. Essa operação é realizada considerando-se a teoria presente no Módulo de Conhecimento como teoria inicial e o BD de Revisões daquele aluno como conjunto de exemplos. Adicionalmente, o sistema pode retirar regras comuns nos BDs de Revisões e utilizá-las na geração de um modelo do grupo. O modelo do aluno em questão será gerado como se segue, apenas incluindo-se essa nova regra :

Dessa forma, o educador poderá visualizar esse modelo e utilizá-lo como uma base de conhecimento onde poderá realizar testes que corresponderão ao comportamento esperado daquele aluno. Como exemplo, pode ser verificado que a regra primos(Marta, Paulo) é concluída como verdadeira nessa teoria, o que indica uma boa probabilidade de essa ser uma conclusão desse estudante. No exemplo acima, como o predicado primos não é utilizado como antecedente em nenhuma regra, o educador poderá verificar que os outros conceitos não foram afetados por essa resposta. No entanto, se isso tivesse acontecido, ele poderia checar que tipo de conclusões errados o aluno poderia cometer em virtude desse erro.

## **7. Conclusões**

Esse artigo propõe um ambiente eficiente de avaliação de estudantes na utilização de sistemas tutores, onde essa avaliação pode ser usada como parâmetro na tomada de decisões pedagógicas pelo educador. Essa abor-

```

esposa(A, B) :- sexo(A, feminino), casado(A, B).
marido(A, B) :- sexo(A, masculino), casado(A, B).
mae(A, B) :- sexo(A, feminino), pai-mae(A, B).
pai(A, B) :- sexo(A, masculino), pai-mae(A, B).
filha(A, B) :- sexo(A, feminino), pai-mae(B, A).
filho(A, B) :- sexo(A, masculino), pai-mae(B, A).
irma(A, B) :- sexo(A, feminino), irmaos(A, B).
irmao(A, B) :- sexo(A, masculino), irmaos(A, B).
tia(A, B) :- sexo(A, feminino), tio-tia(A, B).
tio(A, B) :- sexo(A, masculino), tio-tia(A, B).
sobrinha(A, B) :- sexo(A, feminino), tio-tia(B, A).
sobrinho(A, B) :- sexo(A, masculino), tio-tia(B, A).
tio-tia(A, B) :- casado(A, C), irmaos(C, D), pai-mae(D, B).
tio-tia(A, B) :- irmaos(A, C), pai-mae(C, B).
irmaos(A, B) :- pai-mae(C, A), pai-mae(C, B), A B̄.
primos(A, B) :- pai-mae(C, A), pai-mae(D, B), irmaos(C, D).
* primos(A, B) :- casado(A, C), irmaos(C, D), pai-mae(D, B). *

```

dagem, diferentemente da utilizada em ITSs onde as decisões pedagógicas são realizadas pelo próprio sistema, se apresenta como uma forma mais adequada de se realizar esse tipo de decisões, onde o computador é utilizado como uma ferramenta educacional. Isso decorre do fato de as estruturas pedagógicas serem de difícil entendimento e implementação.

O uso da teoria da revisão para a modelagem do conhecimento do estudante mostra-se uma técnica potencialmente poderosa. Essa técnica trás uma série de vantagens em relação a outras utilizadas em ITS. Uma delas é

a possibilidade de utilização de modelagens anteriores como ponto de partida. Isso aumenta a eficácia da próxima modelagem e diminui seu tempo de obtenção. Outra vantagem é a de que pode-se, a partir dos modelos de um grupo de estudante, analisar o desenvolvimento educacional desse grupo por completo, ou seja, pode-se desenvolver um modelo unificado que represente o conhecimento do grupo em geral. Tal modelo possibilitaria ao professor uma percepção geral do desenvolvimento do aprendizado naquele grupo.

Por ser projetado sobre uma plataforma de distribuição como o CORBA, o ambiente proposto pode ser utilizado sobre qualquer rede heterogênea ou até mesmo sobre a Internet. Tal característica habilita o ambiente como acessório importante no desenvolvimento de cursos a distância, pois possibilita a avaliação dos alunos independente de suas localizações geográficas.

### Referências Bibliográficas

[Burns 90] Burns, I., Parlett, J. W. e Redfield C. L. (1990). Intelligent Tutoring Systems : Evolutions in Design. Hillsdale, NJ: Lawrence Erlbaum Associates.

[Baffes 92] Baffes, P. and Mooney, R. J. (1992). Using theory revision to model students and acquire stereotypical errors. Thirteenth Annual Conference of the Cognitive Science Society, pag. 617-622. Bloomington, IN.

[Baffes 94] Baffes, P. (1994). Automatic student modeling and bug library construction using theory refinement. Tese de Doutorado. Universidade do Texas.

[**Chainben**] Chainben, H. Inteligência Artificial na Educação.  
<http://www.cce.ufpr.br/hamilton/iaed/iaed.htm>.

[**Marietto**] Marietto, M. G. B. Tendências nas Áreas de Sistemas Tutores Inteligentes e Modelagem do Estudante.  
<http://www.ita.cta.br/marietto/tipapaper.html>.

[**OMG 97**] Object Management Group – OMG. The Common Object Request Broker Architecture and Specification. 1997. Revisão 3.0.

[**Richard**] Richard, B. L. e Mooney, R. J. Automated Refinement of First-Order Horn-Clause Domain Theories. *Machine Learning*, pag. 1-39.

[**Souza 99**] Souza, J. T., Pequeno, T. C. e Pequeno, M. C. (1999). First-Order theory revision to student modeling. *Anais do Encontro Nacional de Inteligência Artificial*, Rio de Janeiro, RJ.

[**Wenger 87**] Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems*. Los Altos, CA: Morgan Kaufmann.

## **A.2 Artigo ENIA'99**

Esse artigo foi publicado nos anais do II Encontro Nacional de Inteligência Artificial de 1999 realizado na cidade do Rio de Janeiro em Junho de 1999.

## First-Order Theory Revision to Student Modeling

JERFFESON TEIXEIRA DE SOUZA,

TARCÍSIO CAVALCANTE PEQUENO, MARCELINO CAVALCANTE PEQUENO

Laboratório de Inteligência Artificial (LIA),

Universidade Federal do Ceará (UFC),

Bloco 910 – Campus do Pici

60.455-760 - Fortaleza, CE – Brasil

**Abstract :** One of the most important components in an intelligent tutoring system (ITS) is the student model, which must capture the conceptual state of the student and which enables the system to address its feedback to suit individual instructions. A good modeling technique must be practical and effective. This paper presents a technique to produce effective student models. Such technique uses a machine learning approach called theory revision, which is a method for automatically revising a knowledge base to be consistent with a set of examples. This way, starting with a representation of the correct knowledge of the domain, and given examples of erroneous student behavior, theory revision will revise the knowledge base until it is consistent with the student's behavior. The system FORTE (First-Order Revision of Theories from Examples), which refines first-order theories, is used in this research as the revision tool.

## 1. Introduction

One of the most important components in an intelligent tutoring system (ITS) is the student model, which must capture the conceptual state of the student and which enables the system to address its feedback to suit individual instructions [Baffes 96]. The goal of this model is to produce a representation that accounts for the differences between the correct knowledge of the domain and the behavior of the student. However, this models is not easy to construct. Many researches has been done to attempt constructing of practical and effective models.

Many new machine learning methods have been developed and some of these methods can be used to automate the techniques for student modeling [Sison 96] [Zhang 97]. This join between student modeling and machine learning is the subject of this paper. The main idea is to use a machine learning method to automatically produce effective student models. This method is called the theory revision. It automatically revises an incorrect or incomplete knowledge base, in order to make it consistent with a set of examples [Richard 93] [Sison 96]. However, the theory revision can be used to introduce errors in a correct theory. In this way, theory revision provides a basis for student modeling. Starting with a representation of the correct knowledge of the domain, and examples of erroneous student behavior, theory revision can introduce faulty knowledge that accounts for the student's mistakes.

## 2. Background on Student Modeling

The simplest approach developed to produce a student model is the overlay model [Carr 77]. An overlay model relies in the assumption that a student's knowledge is always a subset of the correct domain knowledge. As the student performs actions which illustrate that he understands a particular element in the domain, these are marked in an overlay model. This way, the unmarked elements are used to focus on tutoring. The main advantage of this approach is its simplicity. However, it presents many disadvantages. One of them is the restricted representations placed on the model. Then, this means that misconceptions cannot be modeled.

To address the limitations of overlay models, some researchers have developed another structure called bug library [Brown 78], which stores the databases of students misconceptions. With the bug library, models are built by matching student behavior against a catalog of expected bugs which are preconstructed by hand through an analysis of student errors. This approach has some advantages. First, the constructions of such catalogs by hand is a difficult task which must be repeated for every new domain. Second, even if great care is taken, the resulting library may still fail to cover a wide enough range of behaviors to function successfully [Baffes 94].

Others researchers have focused on constructing dynamic bug libraries [Sleeman 90]. Here, the idea is to extend a bug library with new student misconceptions. This approach attempts to generate new misconceptions when a student exhibits a problem that cannot be modeled using the misconceptions already in the bug library. The disadvantage is its reliance upon a user who decides which new misconceptions are appropriate extensions for the bug library.

To avoid the problems associated with handconstructed bug libraries, some others researches have used machine learning techniques, which are used to diagnose of misconceptions. Then, the system can operate automatically, constructing models that capture both correct and buggy knowledges. However, the construction of a complete student knowledge model from scratch, from limited training data, has been shown inefficient.

The system ASSERT [Baffes 94] has attempts to use the machine learning technique propositional theory revision to produce student models revising a correct theory to be consistent with a set of examples. Although ASSERT is limited to propositional domains, it is the conceptual predecessor of this research.

### 3. The FORTE System

The main objective of the system FORTE (First-Order Revision of Theories from Examples) [Richard 93] is to develop methods for revising first-order theories using the revision theory, in other words, given an incorrect initial theory and a consistent set of positives and negatives instances, find a revised theory that is correct on the given instances. FORTE views theories as pure Prolog programs.

#### 3.1. Components

- Theory translator : The theory translator is an optional module used to translate between the native representation of a theory and the representation required by FORTE. This is necessary when the native

representation of a theory is not pure Prolog.

- Example translator : FORTE requires examples to be provided as Prolog terms. As with theories, the FORTE representation may not be convenient in all domain. The example translator can be used to translate between a native domain representation and that required by FORTE.
- Language bias : The language bias is used to limit the system's search space when the user knows that certain restrictions must apply to the output theory.
- Revision verifier : The revision verifier is an optimal module that allows the user to insert domain-specific consistency checks in the revision process.
- Fundamental domain theory : The fundamental domain theory is an optional module which provides a place for predicates which the user wishes be desirable.

### 3.2. Algorithm

The system revises theories iteratively, using a hillclimbing approach. Each iteration identifies points in a theory, called revision points, where a revision has the potential to improve the theory's accuracy. It then generates a set of revisions, based on the revision points, selects the best one, and implements it. This algorithm is shown in Figure A.5.

In order to generate revision points, the current theory is tested on the training set. It annotates failed proofs of positive instances and successful

1. Repita
2. Gere pontos de revisão
3. Ordene pontos de revisão pelo potencial
4. Para cada ponto de revisão
5. Gere revisões
6. Atualize melhor revisão encontrada
7. Até potencial do próximo ponto de revisão seja menor
8. do que a pontuação da melhor revisão até então
9. Se a melhor revisão melhora a teoria
10. Implemente a melhor revisão
11. Até que nenhuma revisão melhore a teoria

Figura A.5: FORTE Algorithm.

proofs of negatives. From these annotations it identifies points in a theory for possible revision. Each revision point has a potential, defined as the maximum increase in the theory accuracy which could result from a revision of that point.

It then generates a set of proposed revisions from the revision points, beginning at the point that has the highest potential, and goes down in the list. Each revision receives a score, which is the actual increase in theory accuracy it achieves. Then, it retains the best revision generated so far, where the best revision is the one increasing the accuracy the most. It stops generating revisions when the potential of the next revision point is less than the actual accuracy increase of the best revision generated to date. At that point, the best revision is implemented, and the cycle begins again. This

process continues until the system is unable to generate any revision that would improve the theory.

There are two types of revision points : specialization points and generalization points. The system identifies revision points by annotating proofs of misclassified instances. Points in the theory where proofs of positive instances fail are places where the theory may need to be generalized, and clauses used in successful proofs of negatives instances are points where the theory may need to be specialized. The system FORTE works with six revision operators.

- Operators for Specialization

1. Operator delete-rule :

The simplest way to specialize a clause is to delete it. There are two restrictions. First, if the clause is the only base case of a recursive predicate then it cannot be deleted. Second, if this is the only clause for a top-level concept, it replaces the deleted clause with the rule "concept :- fail". This provides us with a starting point for later revision to the predicate.

2. Operator add-antecedent :

Another approach is to specialize a clause by adding antecedents do discriminate between positive and negative instances. FORTE adds antecedents to a clause in an attempt to make all negative instances unproved. If adding these antecedents also make some positive instances unprovable, it adds the specialized clause to the theory and begins again with the original clause, looking for alter-

native specializations that retain the proofs of the other positive instances while still eliminating the negatives. This process continues until it had a set of clauses that retains the provability of all of the originally provable positive instances.

- Operators for Generalization

1. Operator delete-antecedent :

In many cases, FORTE may be able to create a good revision simply by deleting antecedents from an existent clause. In order to develop a revision, it generalizes the original clause to cover as many positives as possible, without allowing proofs of any negatives. It then adds the generalized clause to the theory. If there are more positives to be covered, it begins again with the original clause and repeat the process. It stops when all of the positive instances listed in the revision point are provable or it is unable to generalize the original clause to allow proof of any of the remaining unprovable instances.

2. 2. Operator add-rule :

This operator develops one or more new versions of an existent rule, while leaving the original rule in the theory. Its objective is to create a new rule that allows proof of the positive instances that identified the original rule as a failure point. Building this rule is a two-step process. First, it creates a generalized rule containing only the core of antecedents essential to keep negatives from being proven, while not interfering with proofs of positives.

Second, it creates one or more specializations of this core rule, which will allow proofs of the desired positives while eliminating the negatives.

3. Operator identification :

It is a predicate-based operator which attempts to generalize the theory by creating a new rule for an existent predicate. It constructs a new clause to generalize the definition of an antecedent that caused one or more proofs of positives instances to fail.

4. Operator absorption :

This operator is a complement of identification. Rather than constructing a new clause for the predicate corresponding to a failing antecedent, absorption looks for an existing clause whose antecedents subsume the failing antecedent, and which has alternative clauses that will allow the failing positive instances to be proven.

#### **4. FORTE as a tool for Student Modeling**

The theory revision is a method for automatically revising a knowledge base to be consistent with a set of examples. Typically, the knowledge base is considered incorrect or incomplete, and the examples represent the correct behavior which the knowledge base should be able to emulate. However, the revision procedure itself is blind to whether or not the input knowledge base is correct in any absolute sense. The theory revision process merely modifies the knowledge base until it is consistent with the examples. Thus,

one can also use theory revision by inputting a correct knowledge base and examples of erroneous behavior, and it will introduce whatever modifications are necessary to cause the knowledge base to simulate the incorrect examples.

This way, we can use the theory revision as a tool for student modeling. Starting with a representation of the correct knowledge of the domain, and given examples of erroneous student behavior, theory revision will revise the knowledge base until it is consistent with the student's behavior. In this point, we will have a model that represents the student's knowledge that can be directly used to guide tutoring feedback by comparing this model with the correct knowledge of the domain.

The use of the FORTE system has some advantages in its use as a student modeler. Firstly, it can produce effective student models without the intervention of the student. In addition, this is the only approach to generate student models that can be produced using models already. For example, a model produced previously can be used as the initial knowledge base. This way, the system will produce a more efficient and faster model.

#### **4.1. Bug Library Generation**

Another advantage of the use of the FORTE system is the possibility of the construction of a bug library. Such bug library will store the common errors presented by the students. It can be generated starting with the models of each student. Then, an algorithm will compare each model, remove the common errors and produce a model that will represent the group's bug library. This library can be used to generate a group model. An ITS could use this group model to produce instructions to the group that would be

coherent with its knowledge. The group model could also be used to represent the initial knowledge base for producing a new student model. This new model will have the same characteristics of the group.

The use of the first-order logic to represent both the domain model and the student model presents some advantages, because it is a much more powerful representation tool than the propositional logic. In addition, as FORTE uses pure Prolog programs, many knowledge bases, developed previously with the use of Prolog, can be used directly.

## 5. Conclusion

This paper presents a method for constructing student models in intelligent tutoring systems (ITS). It is able to construct models efficiently and automatically. The use of theory revision seems to be a powerful tool to produce student models, because it does not require a large number of examples to produce accurate models. In addition, it can use the results of previous modeling to generate models more efficiently and faster. The system is also able to construct a bug library automatically using the student models and to use it into future modeling efforts. Finally, the use of first-order logic makes this system more powerful than other revision systems previously explored.

## References

[Baffes 94] Baffes P. T. Automated student modeling and bug library construction using theory refinement. PhD thesis, University of Texas at Austin, 1994.

[**Baffes 96**] Baffes P. T. and Mooney R. J. Refinement-Based student Modeling automated bug library construction. *Journal of Artificial Intelligence in Education*, 1996.

[**Brown 78**] Brown J. S. and Burton R. R. Diagnostic models for procedural bugs in basic mathematical skills. *Cognitive Science*, 1978.

[**Carr 77**] Carr B. and Goldstein I. *Overlays : a theory of modeling for computer-aided instruction*. Technical Report 1971, Cambridge, 1977.

[**Murray 98**] Murray T. Authoring knowledge based tutors : tools for content, instructional strategy, student model, and interface design. *Journal of the Learning Science*, 1998.

[**Parllet 90**] Parllet J. W., Burns H. and Redfield C. L. *Intelligent tutoring systems : evolutions in design*. Lawrence Erlbaum Associates, Hilldale, NJ, 1990.

[**Richard 93**] Richard B. L. and Mooney R. J. *Automated Refinement of First-Order Horn-Clause Domain Theories*. *Machine Learning*, 1993.

[**Sison 96**] Sison R. and Shimura M. The application of machine learning to student modeling : survey and analysis. Technical report TR96-0010, Tokio Institute of Technology, June 1996.

[**Sleeman 90**] Sleeman D., Hirsh H., Ellery I., and Kim I. Extending domain theories : two case studies in student modeling. Machine Learning, 1990.

[**Wenger 87**] Wenger, E. Artificial intelligence and tutoring systems, Morgan Kaufmann, Los Altos, CA, 1987.

[**Zhang 97**] Zhang X. and Numao M., Knowledge acquisition from complex domains by combining inductive learning and theory revision. Technical report TR97-0012, Tokio Institute of Technology, December,1997.

# Referências Bibliográficas

- [Baffes e Mooney, 92] Baffes, P. e Mooney, R. J. (1992). Using theory revision to model students and acquire stereotypical errors. *In Proceedings of the Thirteenth Annual Conference of the Cognitive Science Society*, pag. 617-622 . Bloomington, IN.
- [Baffes, 94] Baffes, P. (1994). *Automatic student modeling and bug library construction using theory refinement*. Tese de Doutorado. Universidade do Texas.
- [Baffes e Mooney, 96] Baffes, P. e Mooney, R. J. (1996). Refinement- Based student modeling Automated bug library construction. *Journal of Artificial Intelligence in Education*, pag. 75-116.
- [Brown, 75] Brown, J. S., Burton, R. R., e Bell, A. G. (1975), SOPHIE: A Step Towards a Reactive Learning Environment, *International Journal of Man-Machine Studies*, Vol. 7, pag. 675-696.
- [Burns, Parlett e Redfield, 90] Burns, I., Parlett, J. W. e Redfield C. L. (1990). *Intelligent Tutoring Systems : Evolutions in Design*. Hillsdale, NJ: Lawrence Erlbaum Associates.

- [Burton, 79] Burton, R. R. e Brown, J. S. (1979), An Investigation of computer coaching for informal learning activities, *International Journal of Man-Machine Studies*, Vol. 11, pp. 5-24.
- [Carbonnel, 70] Carbonnel, J.R. (1970), AI in CAI: An Artificial Intelligence Approach to Computer Assisted Instruction, *IEEE Transactions on Man-Machine Systems*, Vol. 11, N. 4, pp. 190-202.
- [Chaiben] Chaiben, H. *Inteligência Artificial na Educação*. <http://www.cce.ufpr.br/hamilton/iaed/iaed.htm>.
- [Clancey, 87a] Clancey, W.J. (1987), *Knowledge-Based Tutoring: The GUIDON Program*, The MIT Press.
- [Clancey, 87b] Clancey, W.J. (1987), Methodology for Building an Intelligent Tutoring System, *Artificial Intelligence and Instruction - Applications and Methods*, pp. 193-227.
- [Goodkovsky, 96] Goodkovsky, V. A. (1996). Intelligent tutor : shell, toolkit and technology. *In Proceedings of the Workshop on architectures and methods for designing cost-effective and reusable ITSs*, Montreal, Junho, 1996.
- [Jerinic e Devedzic, 99] Jerinic, L. e Devedzic, V. (1999). A survey of components for intelligent tutoring. SIGCUE/OUTLOOK. Acm press. Vol. 27, Janeiro, 1999.
- [Kaplan e Rock, 95] Kaplan, R. e Rock, D. (1995), New Directions for Intelligent Tutoring, *AI Expert*, pp. 30-40 .

- [Kumar, 92] Kumar, V. S. (1992) *Collaborative intelligent tutoring system : a learning environment*. <http://www.cs.usask.ca/grads/vsk719/academic/publications/afpap/afpap.html>.
- [Marietto] Marietto, M. G. B. *Tendências nas Áreas de Sistemas Tutores Inteligentes e Modelagem do Estudante*. <http://www.ita.cta.br/marietto/tipapaper.html>.
- [Michell, 97] Mitchell, T. M. (1997) *Machine Learning*. McGraw Hill.
- [Mills e Shelly, 96] Mills, K. e Shelly, B. (1996). A collaboration to apply advanced information technology in education. SIGCUE/OUTLOOK. Acm press. Vol. 24, Outubro, 1996.
- [Murray, 98] Murray, T. (1998). Authoring knowledge based tutors : tools for content, instructional strategy, student model, and interface design. *Journal of the Learning Sciences*, Vol. 7, pag. 5-64.
- [Quinlan, 90] Quinlan, J. R. (1990). Learning logical definitions from relations. *Machine Learning*, pag. 239-266.
- [Richards e Mooney] Richards, B. L. e Mooney, R. J. Automated Refinement of First-Order Horn-Clause Domain Theories. *Machine Learning*, pag. 1-39.
- [Sleeman et al., 90] Sleeman, D., Hirsh, H., Ellery, I., e Kim, I. (1990). Extending domain theories: two case studies in student modeling. *Machine Learning*, pag. 11-37.

- [Souza et al., 99a] Souza, J. T., Pequeno, T. H. C. e Pequeno, M. C. (1999). First-order theory revision to student modeling. *Anais do Encontro Nacional de Inteligência Artificial*, Brasília, Julho, 1999.
- [Souza et al., 99b] Souza, J. T., Pequeno, T. H. C. e Pequeno, M. C. (1999) Desenvolvendo um Ambiente Distribuído de Auxílio à Avaliação Conceitual em Sistemas Tutores. *Anais do Simpósio Brasileiro de Informática Educativa*, Curitiba, Novembro, 1999.
- [Sison e Shimura, 96] Sison, R. e Shimura, M. (1996). The application of machine learning to student modeling : survey and analysis. Relatório Técnico TR91-0010. Department of Computer Science, Tokyo Institute of Technology, Junho, 1996.
- [Tekinerdogan] Tekinerdogan, B. *Design of a reflective tutoring system shell*. <http://wwwhome.cs.utwente.nl/~bedir/itsel/>.
- [Wenger, 87] Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems*. Los Altos, CA: Morgan Kaufmann.
- [Woods e Warren, 96] Woods, P. e Warren, J. R. (1996). Adapting teaching strategies in intelligent tutoring systems. *In Proceedings of the Workshop on architectures and methods for designing cost-effective and reusable ITSs*, Montreal, Junho, 1996.
- [Zhang e Numao, 97] Zhang, X. e Numao, M. (1997). Knowledge acquisition from complex domain by comparing inductive learning and theory revision. Relatório Técnico TR90-0012. Department of

Computer Science, Tokyo Institute of Technology, Dezembro,  
1997.