Tese apresentada à Divisão de Pós-Graduação do Instituto Tecnológico de Aeronáutica como parte integrante dos requisitos para obtenção de título de Mestre no Curso de Engenharia Eletrônica e Computação, Área de Informática.

Marcus Vinicius Carvalho Guelpeli

UTILIZAÇÃO DE APRENDIZADO POR REFORÇO PARA MODELAGEM AUTÔNOMA DO APRENDIZ EM SISTEMAS TUTORES INTELIGENTES

Prof.Carlos Henrique Costa Ribeiro, PhD
Orientador

Prof. Homero Santiago Maciel
Chefe da Divisão de Pós-Graduação

Campo Montenegro
São José dos Campos, SP – Brasil
2003

UTILIZAÇÃO DE APRENDIZADO POR REFORÇO PARA MODELAGEM AUTÔNOMA DO APRENDIZ EM SISTEMAS TUTORES INTELIGENTES

Marcus Vinicius Carvalho Guelpeli

Composição da Banca Examinadora:

Prof. Celso Massaki Hirata.....Presidente – ITA

Prof. Carlos Henrique Costa Ribeiro.....Orientador – ITA

Prof. Vakulathil Abdurahiman.....ITA

Prof. Nizam Omar.....MACKENZIE

Profa. Maria das Graças Bruno Marietto.....UCDB

ITA

Dedicatória

A Deus, o autor de todas as Obras.

A minha esposa Alison, em reconhecimento aos muitos momentos roubados do nosso convívio durante a realização deste trabalho.

Agradecimentos

Agradeço meu Mestre e Orientador Professor Carlos Henrique Costa Ribeiro pela sua incansável orientação e sua paciência na condução desde árduo trabalho.Seus conselhos, seus ensinamentos, sua experiência docente vão ficar para sempre como grande aprendizado para minha vida.

Agradeço ao Mestre Nizam Omar por ter sido meu primeiro professor no ITA e ter mostrado o caminho a seguir e pelo qual tenho profunda admiração.

Agradeço ao colega e irmão Ricardo Machado Maia pelo apoio e ajuda.

Resumo

Este trabalho apresenta um novo modelo de módulo de diagnóstico para ser incluído na arquitetura tradicional de Sistemas de Tutores Inteligentes. Neste módulo, são aplicadas técnicas de Aprendizado por Reforço (algoritmo Q-Learning), o que possibilita modelar autonomamente o aprendiz. Um valor de utilidade é calculado baseado em uma tabela de pares estado-ação, a partir da qual o algoritmo estima reforços futuros que representam os estados cognitivos do aprendiz. A melhor política a ser usada pelo tutor para qualquer estado cognitivo do aprendiz é disponibilizada pelo próprio algoritmo de Aprendizado por Reforço, sem que seja necessário um modelo explícito do aprendiz.

Abstract

The goal of this work is to present a new diagnostic module model to be included in an Intelligent Tutoring System (ITS) architeture. In this module, Reinforcement Learning techniques (Q-Learning algorithm) are applied, making it possible to autonomously model the learner. A utility value is calculated based on a state-action table upon which the algorithm estimates future rewards which represent the cognitives states of the learner. The best action policy to be used by the tutor at any learner's cognitive state is made available by a Reinforcement Learning algorithm, with no need for an explicit learner model.

Sumário

CAPÍTULO I INTRODUÇÃO **13**

I.1. O CONTEXTO DESTE TRABALHO **13**
I.2. RAZÕES MOTIVADORAS **15**
I.3. OBJETIVOS **17**
I.3.1. GERAL 17
I.3.2. ESPECÍFICOS 17
1.4 CONTRIBUIÇÕES **18**
I.5. ESTRUTURA DA TESE **18**

CAPÍTULO II SISTEMAS TUTORES INTELIGENTES **20**

II.1. INTRODUCÃO **20**
II.1.1. EVOLUÇÃO HISTÓRICA DOS STI'S 21
II.2. ARUITETURA CLÁSSICA DO STI **23**
II.2.1.MÓDULO DO ESTUDANTE 24
II.2.1.1. Modelo de Overlay 25
II.2.1.2. Modelo de Perturbação 26
II.2.1.3- Modelo Diferencial 26
II.2.1.4- Modelo Procedural 27
II.2.2.MÓDULO DE TUTORIA 28
II.2.3.MÓDULO DE DOMÍNIO 33
II.2.3.1.Modelo de Caixa Preta 33
II.2.3.3.Modelo Cognitivo 34
II.2.4.MÓDULO DE INTERFACE 35
II.3. STI's COMO FERRAMENTAS EDUCACIONAIS E SUAS ESTRATÉGIAS DE ENSINO **36**
II.4. COMO MODELAR O APRENDIZ NOS TUTORES ATUAIS ? **38**

CAPÍTULO III APRENDIZAGEM AUTÔNOMA **40**

III.1. APRENDIZAGEM AUTÔNOMA **40**
III.1.1.PROCESSOS DECISÓRIOS DE MARKOV -PDM 42
III.1.1.1. Papel do Crítico no Processo Decisório de Markov -PDM 43
III.1.1.2. Critérios de Desempenho para Utilidade de Estado 43
III.1.2.TÉCNICA DE APRENDIZADO POR REFORÇO 44
III.2. ESTUDO COMPARATIVO DE MÉTODOS DE MODELAGEM DO APRENDIZ EM STI **49**

CAPÍTULO IV USO DA TÉCNICA DE APRENDIZADO POR REFORÇO EM UM TUTOR INTELIGENTE **52**

IV.1. INTRODUÇÃO AO APRENDIZADO DE MÁQUINA **52**
IV.3. ESTRUTURA DO SISTEMA **55**
IV.3.1. INCLUSÃO NA ARQUITETURA TRADICIONAL DOS STI'S DO MÓDULO DIAGNÓSTICO USANDO TÉCNICA DE AR 57
IV.3.2. FUNCIONAMENTO DO TUTOR COM O MÓDULO DIAGNÓSTICO 58

CAPÍTULO V RESULTADOS E CONCLUSÕES **60**

V.1.IMPLEMENTAÇÃO DO PROTÓTIPO **60**
V.1.1. MÓDULO DO TUTOR 61
V.1.2.MÓDULO DE MODELAMENTO 61
V.1.3.MÓDULO PEDAGÓGICO 61
V.1.4. MÓDULO TABELA (S,A) 61
V.1.5. MÓDULO DIAGNÓSTICO 61
V.2. EXPERIMENTOS REALIZADOS COM O PROTÓTIPO **62**
V.3. HISTÓRICO E ETAPAS **73**
V.4. RESULTADOS EXPERIMENTAIS **75**

CAPÍTULO VI CONCLUSÕES, CONTRIBUIÇÕES E TRABALHOS FUTUROS **111**

VI.1. CONCLUSÕES **111**
VI.2. CONTRIBUIÇÕES E TRABALHOS FUTUROS **112**
VI.3. DIFICULDADES ENCONTRADAS **112**

REFERÊNCIAS BIBLIOGRÁFICAS **114**

ANEXO 1 – POLÍTICA PEDAGÓGICA (P_1) **119**
ANEXO 2 – POLÍTICA PEDAGÓGICA (P_2) **122**
ANEXO 3 – MODELO M1(RUIM) - DETERMINÍSTICO **126**
ANEXO 4 – MODELO M2(BOM) - DETERMINÍSTICO **129**
ANEXO 5 – MODELO M3(EXCELENTE) - DETERMINÍSTICO **132**
ANEXO 6 – MODELO M1(RUIM) – NÃO DETERMINÍSTICO **136**
ANEXO 7– MODELO M2(BOM) – NÃO DETERMINÍSTICO **144**
ANEXO 8– MODELO M3(EXCELENTE) – NÃO DETERMINÍSTICO **154**

Lista de Ilustrações

Figura II.1 – Arquitetura de STI's. 23
Figura II.2 – Representação Gráfica do Modelo Overlay. 25
Figura II.3 – Representação Gráfica do Modelo de Perturbação. 26
Figura II.4 – Representação Gráfica do Modelo Diferencial. 27
Figura II.5 – Representação Gráfica do Modelo Procedural 28
Figura III.1 – Modelo Dinâmico e seus Elementos 40
Figura III. 2 – Modelo de Transição PDM 43
Figura III. 3 – Um agente AR interagindo com seu ambiente. 45
Figura III. 4 – Tabela dos pares (estado - ação). 47
Figura IV. 1 – Sistema com a Técnica de AR. 56
Figura IV. 2 – Classificação do Estado Cognitivo do Aprendiz. 58
Figura IV.3 – Representação do Modelo do Aprendizado por Reforço adaptado ao Sistema de Tutoria. 59
Figura V.1. – Identificação dos módulos do Sistema de Aprendizado por reforço para Modelagem Autônoma do Aprendiz em Tutor Inteligente. 60
Figura V.18. – Gráfico do Reforço Médio do Modelo Determinístico M_1 para 500 Simulações. 75
Figura V.19. – Gráfico do Q(s,a) do Modelo Determinístico M_1 para 500 Simulações. 75
Figura V.20. – Gráficos da Média das Transições dos Estados do Modelo Determinístico (M_1) para 500 Simulações. 76
Figura V.21. – Resultados das Visitas nos Estados. 76
Figura V.22. – Gráfico do Reforço Médio do Modelo Determinístico M_1 para 1000 Simulações. 77
Figura V.23. – Gráfico do Q(s,a) do Modelo Determinístico M_1 para 1000 Simulações. 77
Figura V.24. – Gráficos da Média das Transições dos Estados do Modelo Determinístico (M_1) para 1000 Simulações. 78
Figura V.26. – Resultados das Visitas nos Estados. 78
Figura V.27 – Gráfico do Reforço Médio do Modelo Determinístico M_1 para 2500 Simulações. 79
Figura V.28 – Gráfico do Q(s,a) do Modelo Determinístico M_1 para 2500 Simulações. 79
Figura V.29 – Gráficos da Média das Transições dos Estados do Modelo Determinístico (M_2) para 2500 Simulações. 80
Figura V.30 – Resultados das Visitas nos Estados. 80
Figura V.31 – Gráfico do Reforço Médio do Modelo Determinístico M_2 para 500 Simulações. 81
Figura V.32 – Gráfico Q(s,a) do Modelo Determinístico M_2 para 500 Simulações. 81
Figura V.33– Gráficos da Média das Transições dos Estados do Modelo Determinístico (M_2) para 500 Simulações. 82
Figura V.34– Resultados das Visitas nos Estados. 82
Figura V.35 – Gráfico do Reforço Médio do Modelo Determinístico M_2 para 1000 Simulações. 83
Figura V.37 – Gráfico do Q(s,a) do Modelo Determinístico M_2 para 1000 Simulações. 83
Figura V.38 – Gráficos da Média das Transições dos Estados do Modelo Determinístico (M_2) para 1000 Simulações. 84
Figura V.39 – Resultados das Visitas nos Estados. 84
Figura V.40 – Gráfico do Reforço Médio do Modelo Determinístico M_2 para 2500 Simulações. 85
Figura V.41 – Gráfico do Q(s,a) do Modelo Determinístico M_2 para 2500 Simulações. 85
Figura V.42 – Gráficos da Média das Transições dos Estados do Modelo Determinístico (M_2) para 2500 Simulações. 86
Figura V.43 – Resultados das Visitas nos Estados. 86
Figura V.44 – Gráfico do Reforço Médio do Modelo Determinístico M_3 para 500 Simulações. 87
Figura V.45 – Gráfico Q(s,a) do Modelo Determinístico M_3 para 500 Simulações. 87
Figura V.46– Gráficos da Média das Transições dos Estados do Modelo Determinístico (M_3) para 500. 88
Figura V.47 – Resultados das Visitas nos Estados. 88
Figura V.48 – Gráfico do Reforço Médio do Modelo Determinístico M_3 para 1000 Simulações. 89
Figura V.49 – Gráfico do Q(s,a) do Modelo Determinístico M_3 para 1000 Simulações. 89
Figura V.50– Gráficos da Média das Transições dos Estados do Modelo Determinístico (M_3) para 1000 Simulações. 90
Figura V.51 – Resultados das Visitas nos Estados. 90
Figura V.52 – Gráfico do Reforço Médio do Modelo Determinístico M_3 para 2500 Simulações. 91
Figura V.53 – Gráfico do Q(s,a) do Modelo Determinístico M_3 para 2500 Simulações. 91
Figura V.54 – Gráficos da Média das Transições dos Estados do Modelo Determinístico (M_3) para 2500 Simulações. 92
Figura V.55 – Resultados das Visitas nos Estados. 92
Figura V.56– Gráfico do Reforço Médio do Modelo Não Determinístico M_1 para 500 Simulações. 93

Figura V.57 – Gráfico do Q(s,a) do Modelo Não Determinístico M_1 para 500 Simulações. 93
Figura V.40 – Gráficos da Média das Transições dos Estados do Modelo Não Determinístico (M_1) para 500 Simulações. 94
Figura V.59– Resultados das Visitas nos Estados. 94
Figura V.60– Gráfico do Reforço Médio do Modelo Não Determinístico M_1 para 1000 Simulações. 95
Figura V.61 – Gráfico do Q(s,a) do Modelo Não Determinístico M_1 para 1000 Simulações. 95
Figura V.62 – Gráficos da Média das Transições dos Estados do Modelo Não Determinístico (M_1) para 1000 Simulações. 96
Figura V.64– Resultados das Visitas nos Estados. 96
Figura V.65 – Gráfico do Reforço Médio do Modelo Não Determinístico M_1 para 2500 Simulações. 97
Figura V.66 – Gráfico do Q(s,a) do Modelo Não Determinístico M_1 para 2500 Simulações 97
Figura V.67– Gráficos da Média das Transições dos Estados do Modelo Não Determinístico (M_1) para 2500 Simulações. 98
Figura V.68 – Resultados das Visitas nos Estados. 98
Figura V.69 – Gráfico do Reforço Médio do Modelo Não Determinístico M_2 para 500 Simulações. 99
Figura V.70– Gráfico do Q(s,a)do Modelo Não Determinístico M_2 para 500 Simulações. 99
Figura V.71 – Gráficos da Média das Transições dos Estados do Modelo Não Determinístico (M_2) para 500 Simulações. 100
Figura V.72– Resultados das Visitas nos Estados. 100
Figura V.73 – Gráfico do Reforço Médio do Modelo Não Determinístico M_2 para 1000 Simulações. 101
Figura V.74 – Gráfico do Q(s,a) do Modelo Não Determinístico M_2 para 1000 Simulações. 101
Figura V.75– Gráficos da Média das Transições dos Estados do Modelo Não Determinístico (M_2) para 1000 Simulações. 102
Figura V.76– Resultados das Visitas nos Estados 102
Figura V.77 – Gráfico do Reforço Médio do Modelo Não Determinístico M_2 para 2500 Simulações. 103
Figura V.78 – Gráfico do Q(s,a) do Modelo Não Determinístico M_2 para 2500 Simulações. 103
Figura V.79 – Gráficos da Média das Transições dos Estados do Modelo Não Determinístico (M_2) para 2500 Simulações. 104
Figura V.80– Resultados das Visitas nos Estados. 104
Figura V.81– Gráfico do Reforço Médio do Modelo Não Determinístico M_3 para 500 Simulações. 105
Figura V.82– Gráfico do Q(s,a) do Modelo Não Determinístico M_3 para 500 Simulações. 105
Figura V.83– Gráficos da Média das Transições dos Estados do Modelo Não Determinístico (M_3) para 500 Simulações. 106
Figura V.84 – Resultados das Visitas nos Estados. 106
Figura V.85– Gráfico do Reforço Médio do Modelo Não Determinístico M_3 para 1000 Simulações. 107
Figura V.86 – Gráfico do Q(s,a) do Modelo Não Determinístico M_3 para 1000 Simulações. 107
Figura V.87 – Gráficos da Média das Transições dos Estados do Modelo Não Determinístico (M_3) para 1000 Simulações. 108
Figura V.88 – Resultados das Visitas nos Estados. 108
Figura V.89 – Gráfico do Reforço Médio do Modelo Não Determinístico M_3 para 2500 Simulações. 109
Figura V.90 – Gráfico do Q(s,a) do Modelo Não Determinístico M_3 para 2500 Simulações. 109
Figura V.91 – Gráficos da Média das Transições dos Estados do Modelo Não Determinístico (M_3) para 2500 Simulações. 110
Figura V.92 – Resultados das Visitas nos Estados. 110

Listas de Tabelas

Tabela II.1. – Trabalhos relevantes em STI's 22

Lista de Abreviaturas e Siglas

AR	Aprendizado por Reforço
CBR	Case-Based Reasoning
CAI	Computer-Assisted Instruction
IA	Inteligência Artificial
ICAI	Intelligence Computer-Assisted Instruction
PDM	Processo Decisório de Markov
STI's	Sistemas Tutores Inteligentes

Capítulo I

INTRODUÇÃO

I.1. O CONTEXTO DESTE TRABALHO

Esta tese situa-se na área de Inteligência Artificial, usando a técnica de Aprendizado por Reforço, aplicada a Sistemas Tutores Inteligentes -STI's-, usados na área educacional.

O contexto desta tese é amplo, devido à interdisciplinaridade, uma vez que abrange à área de programas educacionais, onde estes buscam subsídios para sustentações da base teórica na Psicologia, na Pedagogia e na Didática. Quando o processo de aprendizagem passa a ser auxiliado por computadores deve-se acrescentar à base teórica, concepções da Ciência da Cognição e na própria Ciência da Computação.

O objetivo fundamental dos STI's é proporcionar uma instrução adaptada ao aprendiz, tanto no conteúdo como na forma. Porém, o que existe de pesquisas a respeito de modelagem do aprendiz de forma computacional ainda é pouco representativo. Afirma [Giraffa 1999] que as dificuldades em implementar o que já conhecemos a respeito do processamento de informação humano, com os ambientes de que dispomos (Prolog, Lisp, Clips e outros), têm apresentado restrições que implicam, na perda do *modelo mental humano* que conhecemos em favor de um *modelo computacional*, gerando perdas significativas de qualidade. Segue [Giraffa 1999] que esta é uma questão que continuará em aberto até que surjam alternativas em nível de modelagem e implementação.

Afirma [Hendley 1992] que o uso de técnicas de Inteligência Artificial na construção de software educacionais torna-se cada vez mais importante, uma vez que seu estudo fará com que os mesmos tenham mais *qualidade*.Usando técnicas de Inteligência Artificial como recursos de hipermídia e simulação na base do domínio, para que o aprendiz manipule os conteúdos, permitindo assim, construir uma estrutura com estratégia explícita de navegação selecionada a partir do aprendiz, uma vez obtida esta estrutura, os resultados [Giraffa 1999] serão mais eficazes. As interfaces inteligentes que se adaptam às necessidades do aprendiz podem utilizar recursos de realidade virtual [Gallegari 2000]. Também tem sido proposto o uso de agentes ao qual se associam a uma entidade que funciona contínua e autonomamente

em um ambiente, no qual existem outros processos e agentes [Correa 1994]. Este trabalho também visa a utilização de técnicas de Inteligência Artificial, através do uso de métodos de Aprendizado de Máquina, tendo como conseqüência um melhoramento da qualidade do modelo do aprendiz.

A arquitetura do STI descrita nesta tese utiliza Aprendizado por Reforço, que é uma técnica de Inteligência Artificial.Usa-se o algoritmo *Q-Learning* [Watkins 1989] para executar este tipo de aprendizado. O algoritmo permite modelar autonomamente o aprendiz, representando em uma tabela de pares estado-ação seus estados cognitivos e as ações que o levaram para aquele estado.

Através do algoritmo *Q-Learning,* o maior valor de utilidade é calculado baseado na tabela de pares estado-ação. As ações que representam uma estratégia pedagógica adotada pelo tutor são atualizadas pelo módulo diagnóstico na tabela e indica(m) para o tutor a(s) ação (ões), com maior valor de utilidade para um dado estado cognitivo do aprendiz. Isto ocorre de forma dinâmica, ou seja, à medida que o aprendiz vai interagindo com o tutor estes valores vão sendo atualizados pelo módulo diagnóstico na tabela de pares estado-ação.

De acordo com [Giraffa 1999], um modelo realista do aprendiz implica em uma atualização dinâmica, enquanto o sistema avalia o desempenho do mesmo. O sistema de AR poderá ser usado nos tutores atuais para melhorar o modelamento do aprendiz, representar de uma forma dinâmica seus estados mentais e proporcionar o uso de múltiplas estratégias pedagógicas para domínios diferentes.

I.2. RAZÕES MOTIVADORAS

Quando da observação dos trabalhos sobre STI's e, especificamente, relativos ao módulo do estudante, nota-se que existe uma dificuldade em modelar o aprendiz de forma dinâmica, no momento de sua interação com o tutor, ou seja, representar de forma computacional (representação com abordagem na Ciência da Computação) seus estados mentais, já que da forma didático – pedagógico, ou seja, de maneira não algorítmica (representação com abordagem na área Pedagógica e Psicológica) existem trabalhos de grande relevância..

A maior dificuldade encontrada é definir o modelo do aprendiz que está interagindo com o tutor, pois existe insuficiência das teorias computacionais em representar estruturas mentais do ser humano frente ao conhecimento. O que realmente existe são os paradigmas de aprendizagem, ou seja, não existem teorias computacionais que possam ser descritas através de um processo formal. Segundo [Giraffa 1999], o que é considerado como teoria na área de Educação, nem sempre satisfaz os requisitos de uma teoria formal na área de Ciência da Computação. A maioria das teorias consideradas em Educação não apresentam um modelo computacional necessário sob o ponto de vista da Ciência da Computação. Este modelamento de forma quase individualizada é indispensável, devido a grande necessidade de adaptar sistemas de tutoria a um maior número de usuários. Obter este modelo, com domínios diferentes torna-se uma tarefa extremamente difícil, quase impensável formalmente.

É preciso ainda explorar os diferentes estilos individuais de aprendizagem através de várias estratégias de ensino. De acordo com [Giraffa 1999], dispomos atualmente de modelos muito mais empíricos do que realmente gostaríamos de utilizar. A modelagem do aprendiz deve proporcionar depuração (um filtro) dos seus estados mentais, que mudam conforme o conteúdo apresentado e sua evolução cognitiva, associadas com as ações do aprendiz face ao tutor. A forma de captar fidedignamente estes estados mentais ainda é uma questão em aberto.

Tendo como motivação estas dificuldades, este trabalho propõe o uso do *Aprendizado por Reforço,* que utilizando o algoritmo *Q-Learning,* vai escolher a ação de maior valor de utilidade para um dado estado cognitivo do aprendiz, de acordo com uma estratégia pedagógica. Sendo assim, obtém-se uma política de ações que leva um aprendiz a um estado cognitivo superior ao qual ele se encontra, tornando assim implícita a representação formal dos estados mentais do aprendiz. Esta depuração dar-se-á de forma autônoma e dinâmica em uma *tabela de pares estado*-ação, usada pelo algoritmo *Q-Learning*. Cabe ainda a proposta da inclusão de um módulo de diagnóstico na arquitetura tradicional dos Tutores. Este módulo

atualiza a *tabela de pares estado-ação,* calcula os reforços para um dado estado cognitivo do aprendiz e ainda indica para o módulo de tutoria qual a melhor ação (maior valor de utilidade) a ser escolhida, de acordo com a estratégia pedagógica. Com isso, seria possível trabalhar com múltiplas estratégias pedagógicas de forma conjunta e com domínios que sejam correlatos . Explica-se esta autonomia e flexibilidade, devido ao uso da técnica de AR, que obtém on-line (sem necessidade de um treinamento anterior) as ações com maior valor de utilidade como conseqüência, garantindo um modelo do aprendiz.

I.3. OBJETIVOS

I.3.1. Geral

Contribuir para um maior entendimento teórico e prático do problema da utilização do sistema de aprendizado por reforço usado para modelar autonomamente o aprendiz em um tutor inteligente, como também determinar dinâmica das estratégias instrucionais através da política de ações, que é obtida no final pelo algoritmo Q-Learning.

I.3.2. Específicos

O atual estado da arte dos Sistemas Tutores Inteligentes permite que uma arquitetura não usual, ou seja, uma arquitetura não clássica possa ser proposta e modelada a partir do conhecimento que se tem da arquitetura clássica.

O modelo de arquitetura dos Sistemas Tutores Inteligentes, concebido de acordo com a idéia anterior, destina-se a alcançar os objetivos do problema motivador e resolvê-lo. Este modelo apresentar-se-á com as seguintes capacidades específicas:

- Inserir nos tutores uma classificação do estado cognitivo do aprendiz, para que a cada novo estado mental este possa estar dentro da escala de valores relativos, facilitando assim a classificação inicial (feita por questionário) e as demais.
- Criar um log de registro para cada aprendiz onde este armazenaria seu último estado cognitivo evitando assim a repetição do questionário inicial. Este log armazenaria também o último conteúdo apresentado ao aprendiz, possibilitando o recomeço do ponto no qual ele havia parado.
- Incorporar ao modelo técnicas de Aprendizado por Reforço usando o algoritmo Q-Learning para, através deste, formalizar o uso de técnicas computacionais na modelagem do aprendiz.
- Incluir na arquitetura tradicional dos STI's o Módulo de Diagnóstico que tem como objetivo atualizar os valores da tabela de pares estado-ação e indicar para o módulo tutoria a ação com maior valor de utilidade.
- Possibilitar o uso de múltiplas estratégias pedagógicas, já que o modelamento do aprendiz acontece através de processo autônomo e dinâmico.

- Usar domínios correlatos, já que existem múltiplas estratégias e um modelamento do aprendiz de forma autônoma e dinâmica.

1.4 CONTRIBUIÇÕES

Este trabalho não pretende ser exaustivo a ponto de suprir todas as lacunas na área de modelagem autônoma do aprendiz em STI's, porém deve se levar em consideração sua contribuição em associar uma técnica de IA (Aprendizado por Reforço) à modelagem autônoma e dinâmica do aprendiz em Tutores Inteligentes.

Segundo autores citados neste trabalho [Marietto 2000], [Leite 1999], [Giraffa 1999], [Viccari 1990], [Correa 1994], [Hendley 1992], existe uma falta de formalismo nas bases teóricas dos STIs, dificultando a representação computacional de estados mentais do aprendiz. Quando se faz esta representação com base no formalismo da Ciência da Computação, há perda da representação do modelo mental humano. Considerando este problema, busca-se no trabalho uma contribuição formal e prática, para uma melhor representatividade do modelo mental humano, preservando com isso, as bases teóricas.

I.5. ESTRUTURA DA TESE

Esta tese compõe-se de seis capítulos, sendo o primeiro constituído da presente introdução. Em cada capítulo serão abordados os seguintes assuntos:

- Capítulo 2: Apresenta uma introdução aos Sistemas Tutores Inteligentes, mostrando sua evolução histórica e sua arquitetura clássica. São mostrados os 4 módulos com suas subdivisões, e discutidos conceitos. Aborda ainda a função dos STI's como ferramentas educacionais e finaliza a modelagem do aprendiz nos tutores atuais.

- Capítulo 3: Apresenta a técnica de Aprendizado por Reforço. Apresenta a teoria do processo decisório de Markov e o papel do crítico no PDM. São vistos ainda critérios de desempenhos para utilidades e maximizações dos estados e o uso do algoritmo Q-Learning na técnica de Aprendizado por Reforço.

- Capítulo 4: Propõe o uso da técnica de Aprendizado por Reforço em um tutor inteligente para a definição da estratégia de ensino a ser usada pelo tutor. Propõe também a inclusão na arquitetura tradicional dos STI's de um Módulo Diagnóstico. Com o uso do módulo diagnóstico, obtém-se um modelo do aprendiz. Com o uso

das técnicas de AR, transfere-se parâmetros para o tutor, para que este possa adotar uma estratégia, baseada no modelo do aprendiz, afim de que se obtenha o maior ganho para seu estado cognitivo atual.

- Capítulo 5: Apresenta os resultados das simulações, realizados sobre um protótipo de sistema de tutoria inteligente, usando o módulo diagnóstico e o Algoritmo de Aprendizado por Reforço (*Q-Learning*). Descreve cada modelo simulado e os resultados obtidos. Por questões evolutivas demonstra as etapas do experimento até a obtenção dos resultados finais. Aborda ainda, as questões de limitação do projeto, como projeções para trabalhos futuros.
- Capítulo 6: apresenta conclusões, contribuições e trabalhos futuros.

Capítulo II

SISTEMAS TUTORES INTELIGENTES

II.1. INTRODUCÃO

Sistemas Tutores Inteligentes modelam a representação do conhecimento humano. Os STI's são uma evolução do CAI (*Computer-Assisted Instruction*) que, por sua vez, evoluíram com as técnicas de Inteligência Artificial (IA), aumentando o grau de "Inteligência " dos sistemas educacionais (ICAI). Hoje, qualquer sistema que tenha o objetivo de ensinar e incorpore princípios (técnicas) de IA é denominado Sistema Tutor Inteligente.

Segundo [Giraffa 1999] os STI's são uma modalidade dos ICAI e modela o aprendiz de forma individualizada. Quando esse modelo é "fraco",ou seja, o perfil do aprendiz não esta bem modelado, fazendo com que seus estados cognitivos não tem uma representação fidedigna, têm-se os STIs, como assistentes. Quando é "forte",ou seja, uma representação exata dos seus estados mentais, tem-se como tutores. Os STI's possibilitam ao aprendiz a capacidade de aprender com o tutor, servindo este como guia no processo; ele deve se adaptar ao aprendiz, e não o contrário, como acontece no método tradicional. Com isso, é necessário um modelamento do aprendiz, para que os STI's possam saber o que ensinar, a quem ensinar e como ensinar. Ele deve ser capaz de mudar o nível de entendimento para responder às entradas do aprendiz, em vários níveis, podendo mudar as estratégias pedagógicas, adaptando-se de forma individualizada, de acordo com o ritmo e as características de cada aprendiz.

Nos STI's o ensino é construído sobre uma base de conhecimento (domínio), criado por um especialista que interage com o aprendiz.

Os STI's são programas que, interagindo com os aprendizes: modificam suas bases de conhecimento, percebem suas intervenções, possuem a capacidade de aprender e adaptar-se às estratégias de ensino, de acordo com o desenrolar do diálogo com o aprendiz [Viccari 1990]. Caracterizam-se principalmente, por construir representação cognitiva do aprendiz, através da interação, formulação e comprovação de hipóteses sobre o estilo cognitivo do aprendiz; seu procedimento, o seu nível de conhecimento do assunto e suas estratégias de aprendizagem,

adequam-se à situação do ambiente. Porém, em detrimento de um formalismo da Ciência da Computação, perde-se a representação do estado cognitivo do aprendiz.

Para ser inteligente, um tutor deve ser flexível, isto é, deve ter capacidade para aprender com o meio ambiente e atualizar seus conhecimentos [Viccari 1990].

II.1.1. Evolução Histórica dos STI's

Autores como Carbonell [Carbonell 1970] e Sleeman [Sleeman 1982] observaram que os programas de CAI possuíam uma estrutura de transmissão de conhecimento seqüencial, determinada previamente por conseqüência da incapacidade de adaptação individual à realidade do aprendiz. Em meados dos anos 80, pesquisas concentraram-se nas questões do aspecto pedagógico, ou seja, valor instrucional. Já nos anos 90, os pesquisadores começaram a explorar os ambientes e variáveis instrucionais específicas.

A primeira geração de Sistemas de Tutores Inteligentes surgiu no inicio dos anos 70 até meados dos anos 80, com o objetivo de explorar as técnicas de IA, enfatizando a representação do conhecimento e sua classificação cognitiva. Sua característica era trabalhar com Sistemas ilustrativos do funcionamento de técnicas de IA. SCHOLAR, WHY, GUIDON são exemplos de sistemas de primeira geração.

A segunda geração dos STI's surgiu no final dos anos 80 e início dos anos 90, com o objetivo de aplicação e teste dos STI's, enfatizando os aspectos pedagógicos, e tendo como principal característica um escopo maior do domínio. GEOMETRY TUTOR e LISP TUTOR são exemplos de sistemas da segunda geração.

A terceira geração dos STI's surgiu em meados dos anos 90 e predomina até hoje. Seu maior objetivo é a representação, comunicação e conhecimento do aprendiz, enfatizando as tecnologias hipermídia. Suas características são as equipes multidisciplinares. Segundo [Giraffa 1999] a terceira geração explora ambientes e variáveis instrucionais, usando equipes interdisciplinares para realizar atividades multidisciplinares, ou seja, dando ênfase à importância da aplicação de IA aos vários contextos de ciências afins. Essa geração [Giraffa 1999] dos STI's deve contribuir para que novas teorias e estratégias pedagógicas sejam colocadas nos STI's, pois as já existentes são fixas e em números reduzidos, restringindo-se apenas a uma. [Burns, Parllet & Redfield 1991] salientam também que as pesquisas dos STI's, principalmente as de ensino-aprendizagem, devem enfocar as estratégias de ensino, em particular, dando ênfase aos estados cognitivos de cada aprendiz.

Na terceira geração, os modelos baseados em IA representam melhor o meio de comunicação do conhecimento, porque apresentam uma capacidade de modelagem cognitiva, facilitando as decisões educacionais à medida que o aprendiz utiliza o sistema. Dentro desta perspectiva, o processo de aprendizagem pode ser concebido como o mapeamento do conhecimento do tema a ser ensinado.Nas pesquisas em qualquer tipo de sistema educacional envolvendo princípios de IA, o propósito principal é captar o conhecimento necessário que permita aos especialistas, compor uma interação educacional. De acordo com [Leite 1999] os Tutores Inteligentes devem incorporar os conhecimentos didático/pedagógicos dos professores, adquiridos informalmente para que componham o modelo produzido em IA. Portanto, também é responsabilidade dos programas compor interações educacionais dinamicamente, ou seja, seguindo [Leite 1999] pode-se estabelecer uma escala informal do conhecimento até a representação formal.

Na tabela II.1 são mostrados projetos envolvendo STI's de relevância histórica, para entender a evolução das pesquisas na área.

Tabela II.1. – Trabalhos relevantes em STI's

Nome	***Ano***	***Domínio***	***Características***
SCHOLAR	1970	Geografia	Jogo de perguntas e respostas.
SOPHIE	1973	Eletrônica	Baseado em um simulador de circuitos
GUIDON	1972-80	Medicina	Interface gráfica para utilização do NEOMICYN, sistema especialista baseado em regras.
PIXIE	1987	Álgebra	Shell para desenvolvimento de tutores.
WHY	1988	Clima	Perguntas e Respostas.
ITSSOL	1988	Cont. de Submarino	Ambiente de simulação de painel de comandos.
PROUST	1988	Pascal	Detecta erros de programadores na linguagem PASCAL por reconhecimento de planos.
BUGGY	1988	Aritmética	Diagnóstico de erros aritméticos elementares através de redes procedurais.
ELECTRE	1988	Eletricidade	Arquitetura Baseada em regras.
TUTOR PROLOG	1989	Prolog	Ambiente para ensino de programação Prolog.
ELETRO TUTOR	1992	Eletricidade	Sistema em Prolog.
CARNEY	1994	Telecomunicação	Arquitetura baseada em regras.
CALAT	1995	Telecomunicação	Arquitetura distribuída sobre INTERNET.
ARESFED	1996	Administração	Arquitetura distribuída baseada em sist. de trabalho cooperativo.

II.2. ARUITETURA CLÁSSICA DO STI

Os Sistemas Tutores Inteligentes apresentam uma arquitetura clássica que é composta de 4 módulos(conforme ilustrado na Figura II.1):

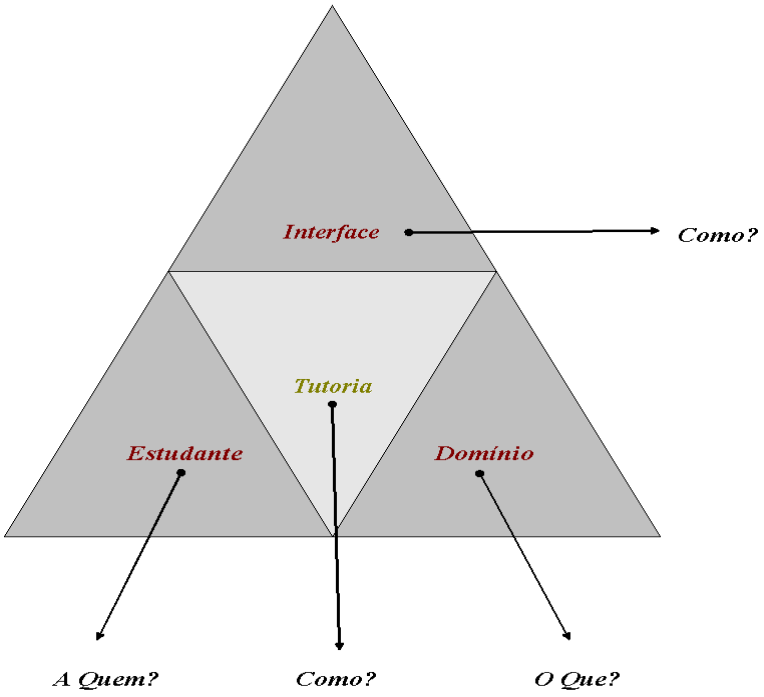

Figura II.1 – Arquitetura de STI's.

São os módulos:

- Módulo do Estudante;
- Módulo de Tutoria;
- Módulo de Domínio;
- Módulo de Interface.

Segundo [Marietto 2000] os módulos têm como objetivo separar especificações sobre o conhecimento de questões didático-pedagógicas, para criar arquiteturas operacionais para sistemas educacionais. Cada um desses módulos tem uma característica e relacionamento próprios entre eles, mas depende de cada pesquisador o enfoque e a maneira em que este enxerga o processo de ensino-aprendizagem. Neste trabalho, o maior enfoque será dado ao módulo do estudante, pois será com os resultados obtidos com uso da técnica de Aprendizado por Reforço, que por conseqüência teremos delineado o estado cognitivo de cada aprendiz, ao

qual se disponha a interagir com o tutor. Importante salientar o ganho para o módulo de tutoria, que poderá usar uma base de dados, e através desta, armazenará múltiplas estratégias pedagógicas, para serem usadas conforme a evolução do estado cognitivo do aprendiz. Outro ganho significativo é o do domínio, ao qual poderá ser usada qualquer área do saber, já que o modelo proposto neste trabalho não determina quais ações específicas devam ser tomadas pelo tutor, e sim aquelas com maior valor de utilidade para um dado estado cognitivo do aprendiz. Trata-se, neste trabalho, das ações que compõem o domínio, como ações generalistas, ao qual pode se adaptar qualquer conteúdo ou múltiplos conteúdos desde que esses sejam correlatos, já que este, vai estar ligado a múltiplas estratégias pedagógicas, bastando apenas que elas existam para serem aplicadas, ou seja, a técnica de AR indica a melhor estratégia pedagógica para um dado estado cognitivo do aprendiz e como conseqüência acha de forma dinâmica (on-line) o modelo deste aprendiz.

II.2.1.Módulo do Estudante

Busca captar o atual estado cognitivo do aprendiz, individualizando o ensino, representando aspectos do comportamento e conhecimento do aprendiz. Deve ser capaz de detectar erros cometidos pelos aprendizes e verificar mudanças no perfil do aprendiz, gerando processo de diagnóstico.

A complexidade do projeto está na modelagem do modelo do aprendiz, que vai medir o estado cognitivo do mesmo a partir do comportamento observável e, com isso, poder indicar uma estratégia de ensino a ser utilizada.

Esta estrutura pode ser estática ou dinâmica [Viccari 1990]. Para melhor representar o estado cognitivo do aprendiz, a estrutura ideal seria a dinâmica, face ao uso que o aprendiz faz do sistema em forma diversificada, adaptando-se ao mesmo; de acordo com o seu desempenho perante os questionários e modificando a base de dados para representar novos conhecimentos, que o tutor poderia aprender com o aprendiz.

Decisões pedagógicas razoáveis são fatores de desafio para os sistemas computacionais, já que as pessoas, nem sempre conseguem representar seus próprios processos mentais e, algumas vezes, ficam confusas.

As técnicas usadas para construir um modelo do aprendiz devem se deter em um log de respostas dadas por ele, anteriormente. Os pontos em comum do aprendiz com o especialista levam em consideração sua área de interesse e seus objetivos.

O modelo do aprendiz pode ser representado tomando como base o modelo de descrição [VanLenh 1988]:

- Modelo de Overlay;
- Modelo de Perturbação;
- Modelo Diferencial;
- Modelo Procedural.

II.2.1.1. Modelo de Overlay

É a equiparação entre a representação do conhecimento no modelo do aprendiz e a base do domínio. Este modelo assume que "incoerências" do aprendiz segundo[VanLenh 1988] são concepções ausentes, ou seja, são devidas à falta de informação na base do domínio, e não levam em consideração as possíveis "confusões" no processo mental do aprendiz que de acordo com [VanLenh 1988] são concepções incorretas.

De acordo com a Figura II.2 o modelo do domínio é representado pelo retângulo e o modelo do aprendiz representado pela elipse. A sobreposição das figuras indica que o modelo do aprendiz é um subconjunto do modelo do domínio,isto faz com que este modelo seja pouco abrangente não tratando das informações que estão fora do modelo do domínio, ou seja, não considerando confusões mentais do aprendiz, resumindo-se apenas ao que foi colocado neste modelo pelo especialista.

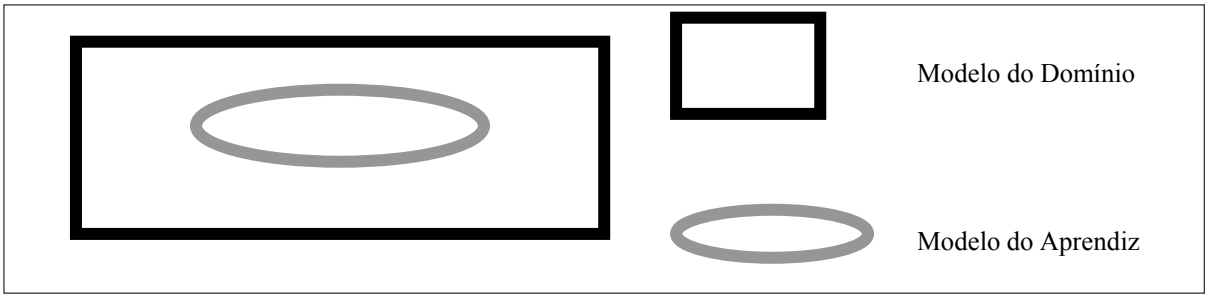

Figura II.2 – Representação Gráfica do Modelo Overlay.

II.2.1.2. Modelo de Perturbação

Este modelo também se baseia na equiparação do modelo aprendiz e base do domínio, porém, ele parte da premissa da criação de uma biblioteca de erros (*Bugs*), proposta inicialmente por Brown e Burton [Brow & Burton 1982] e recebeu o nome de modelo BUGGY. Essa biblioteca de erros típicos do aprendiz na base do conhecimento, pode-se assim dizer que um erro do aprendiz seria a falta de conhecimento de algum conceito ou uma concepção errada do mesmo. Este é ainda, um modelo limitado, pois toma como base do conhecimento apenas uma biblioteca de erros , que seria incapaz de representar toda a base do conhecimento.

Na Figura II.3 também o modelo do domínio é representado pelo retângulo, o modelo doaprendiz, pela elipse e, existe o quadrado que representa a biblioteca de bugs. Esta área , ao qual denominou-se de biblioteca de bugs que inclui possíveis erros ou falsas concepções do aprendiz.

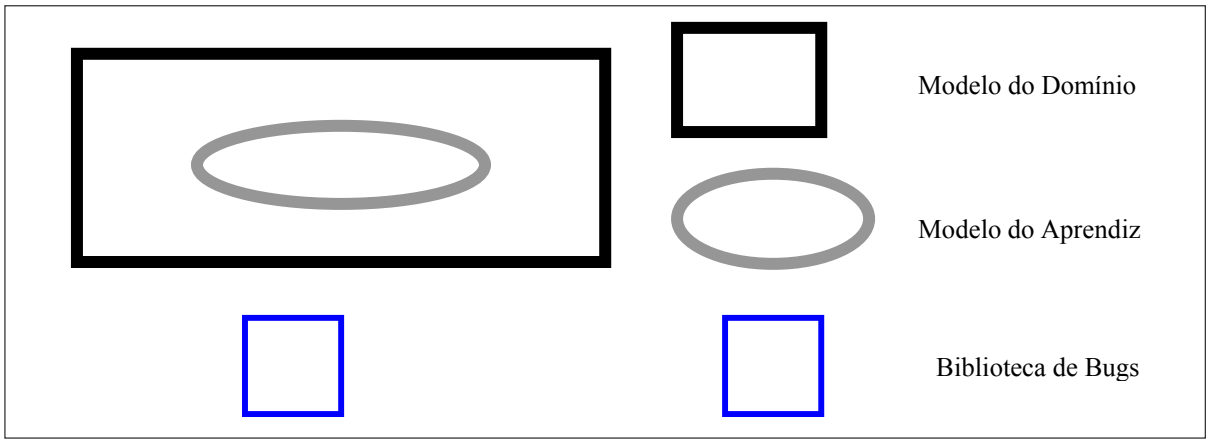

Figura II.3 – Representação Gráfica do Modelo de Perturbação.

II.2.1.3- Modelo Diferencial

É o que compara a resposta do aprendiz com a base de conhecimento, estabelecendo assim, uma comparação entre o modelo do aprendiz e o especialista. Esta modelagem, segundo [Damico 1995], se compara a performance do especialista com a do aprendiz e não o conhecimento deles. Neste caso, teríamos o modelo do aprendiz baseado na capacidade de modelagem do especialista, em representar o conhecimento, presente ou ausente no aprendiz.

A Figura II.4 também o modelo do domínio é representado pelo retângulo, o modelo do aprendiz representado pela elipse e através de um quadrado é a representação do conhecimento esperado do aprendiz,ou seja, a divisão do conhecimento em 2 categorias

– o que o aprendiz deveria saber;

– o que não se pode esperar que ele conheça.

Desta forma o modelo do aprendiz torna-se muito restrito aos conhecimentos do especialista.

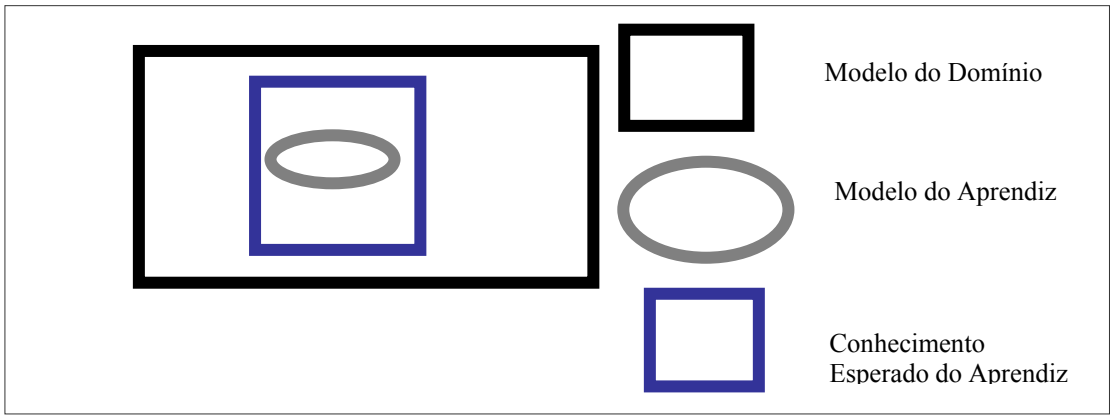

Figura II.4 – Representação Gráfica do Modelo Diferencial.

II.2.1.4- Modelo Procedural

Este modelo reflete um processo incremental; um conjunto de etapas da qual existem outros mecanismos para armazenar o conhecimento do aprendiz, como redes bayesianas, que representam através dos seus nós, estados probabilísticos de conhecimento do aprendiz. Uma outra forma de modelar os estados mentais do aprendiz é através das crenças. Estas, não terão uma influência direta no sistema mas ajudarão a compor um modelo do aprendiz, em particular, considerando seus desejos, intenções, expectativas e seus objetivos pessoais.

A figura II.5 mostra o retângulo representando o modelo de domínio, a elipse representando o modelo do aprendiz, o cubo representando o modelo de tutoria e o quadrado alguma Técnica de IA utilizando o Aprendizado.

A ligação que existe entre o modelo do aprendiz e o modelo de domínio, apenas uma representação, ou seja, os modelos são independentes havendo apenas uma adaptação do modelo de domínio ao perfil do aprendiz. Já as ligações existentes entre o modelo de tutoria, modelo de domínio e Técnica de IA representam a interação entres os modelos e o aprendizado, ocorrendo de forma incremental.

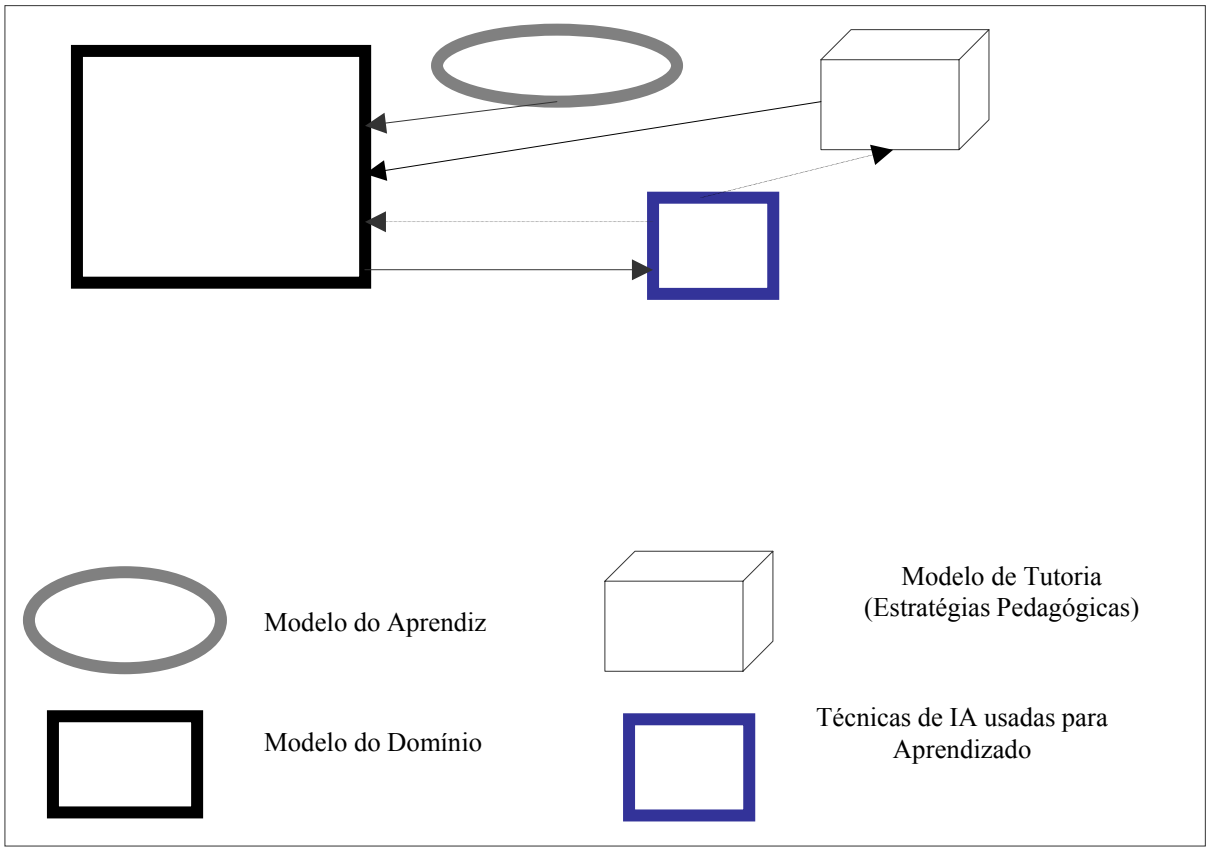

Figura II.5 – Representação Gráfica do Modelo Procedural

II.2.2.Módulo de Tutoria

É o módulo que centraliza as ações dos outros módulos. É neste que encontra-se as estratégias didáticas e pedagógicas adotadas por cada STI. Essas estratégias são responsáveis pelo planejamento instrucional, que objetiva apresentar um conteúdo ao aprendiz de acordo com um diagnóstico de suas deficiências e eficiência de conhecimento em um dado domínio.

O problema se constitui em estabelecer uma estratégia de ensino que adote uma abordagem junto ao aprendiz, de forma que algumas questões sejam estabelecidas, tais como:

- Em que momento se deve intervir?
- Que razões levaram a esta interrupção?
- Como deve abordar tal erro?
- Como se deve avaliar e quais os critérios envolvidos nesta avaliação?
- Como se deve apresentar esta solução, e de que forma será esta seqüência de instrução?
- De que maneira se deve responder aos questionamentos do aprendiz em relação ao conteúdo apresentado?
- Deve-se mudar ou sugerir alterações no modelo do aprendiz ou, até mesmo, de forma dinâmica, sugerir alterações nas estratégias de ensino?
- Como dizer? O que apresentar ao aprendiz?

Para estas questões não existem soluções gerais e, a maioria dos autores, indica a falta de utilização de teorias pedagógicas que especifiquem claramente tais questionamentos. Aliada a essas questões pode-se citar também a dificuldade de modelar os estados mentais do aprendiz.

Observa-se com freqüência que os STI's têm como estratégia pedagógica o controle do comportamento do aprendiz adotando uma abordagem instrucionista, ou seja, não permitindo que este tenha qualquer tipo de ação própria no aprendizado, restringindo suas ações fazendo, desta forma, com que o aprendiz não tenha liberdade em suas respostas. Este tipo de estratégia foi baseada no modelo proposto por Skinner [Skinner 1950], onde o conteúdo a ser ensinado deve ser subdividido em módulos, estruturados de forma lógica, de acordo com perspectiva pedagógica de quem planejou a elaboração do material instrucional. Ao final de cada módulo, o aprendiz deve responder a uma pergunta, cuja resposta correta leva ao módulo seguinte. Caso a resposta do aprendiz não seja correta, o aprendiz retorna aos módulos anteriores, até obter sucesso. Enquanto as interações dos programas instrucionistas enfatizam o software e o hardware [Almeida 2000] com o objetivo de ensinar o aprendiz, uma outra abordagem faz com que o software construído pelas interações com aprendiz tenham como características, a *individualidade* ou *cooperativismo,* que são características do construtivismo, que tem como objetivo *pensamento*, *criação*, *desafio*, *conflito* e a *descoberta.*

Desta forma, a abordagem pedagógica com a utilização de computadores se realiza sob abordagens que se situam e "oscilam entre dois grandes pólos" - instrucionista e construtivista [Valente 1993]. Sendo assim cita Bustamante [Bustamante 1996]. "De um lado, a riqueza de imagens e as múltiplas opções; de outro, o programa sem nada, a não ser o desafio a explorar, descobrir e demonstrar. A interação grupal, a troca. A conclusão extraída a partir do desafio".

A estratégia pedagógica adotada em cada STI vai caracterizar como trata-se as respostas do aprendiz, seus erros e a forma de modelar o domínio do conhecimento. As estratégias pedagógicas devem incluir funções de orientação no desempenho das atividades, explanações dos fenômenos e processos e correções, bem como de adaptações específicas e individuais, gerando desafios, explicações, exemplos e/ou contra-exemplos no decorrer das interações. A partir de uma abordagem de aprendizagem assume-se um plano de ações didáticas, constituindo-se em *táticas* sobre *como*, *quando* e *o quê* trabalhar com o aprendiz. Para que surja o processo de ensino-aprendizagem, isto é, as técnicas de *instrução*. Instrução, de acordo [Halff 1988], é a forma da apresentação do conteúdo a ser ensinado ou aprendido peloaprendiz. *Currículo* é a seleção e o seqüenciamento do conteúdo a ser apresentado ao aprendiz. Dependendo do método de ensino que é adotado, tem-se maior ou menor esforço na seleção e seqüenciamento do conteúdo (currículo) ou na instrução.

Existem diversas estratégias pedagógicas empregadas nos STI's como:

- Modelo de Ensino Socrático ou através de diálogos exploratórios;
- Modelo de Ensino por Treinamento (Coaching);
- Modelo de Ensino baseado em Casos (Case-based teaching);
- Modelo de Ensino baseado em ambientes exploratórios.

Segue abaixo, uma explicação destas estratégias pedagógicas sendo aplicadas ao ambiente computacional não entrando no mérito da questão didático/pedagógica.

O modelo de ensino socrático é conduzido pelo tutor, a partir de um tópico no qual o aprendiz demonstre já possuir algum conhecimento. Assim, o tutor aprofunda um conteúdo do domínio que o aprendiz tenha citado ou que o aprendiz, constatadamente, já domine. Entretanto, os primeiros tutores que trabalhavam com uma abordagem socrática, apresentavam limitações que se tentava amenizar acrescentando regras associadas a condições, especialmente em resolução de conflitos (quando muitas ações são associadas à uma condição). Outra maneira, é apoiar o ensino socrático de uma hierarquia de dificuldades

crescentes, pré-estabelecidas como descrito em [Viccari 1990]. Neste método, o tutor ensina o aprendiz através de uma abordagem de exposição indireta, que consiste em oferecer ao aprendiz uma sucessão de questões que visam formular princípios gerais baseados em casos particulares para ele poder analisar hipóteses, descobrir contradições e finalmente fazer inferências corretas.

Em um STI's, que adota o modelo de *coaching*, o sistema monitora as atividades do aprendiz, reconhece o comportamento não ótimo e interrompe para dar conselhos. O modelo *coaching* pode ser utilizado em resolução de problemas, sendo neste caso uma poderosa aplicação de um dos tipos da aprendizagem cognitiva. Para a implementação desta estratégia, um tutor e um aprendiz resolvem problemas em conjunto. Algumas vezes, o aprendiz lidera o processo de resolução e o tutor somente indica sua concordância ou não em cada etapa. Em outras ocasiões, o tutor orienta o aprendiz através das partes mais difíceis ou as quais o aprendiz não consegue resolver sozinho. Entretanto, todo o tempo, eles estão trabalhando na modalidade de resolução de problemas. Diz-se que um treinador está modelando os processos cognitivos, quando o tutor está dirigindo a resolução do problema. Um tutor-treinador que pouco interrompe o processo de resolução de problemas está esmorecendo sua plataforma de ajuda.

O modelo baseado em casos, afirma que a maior tarefa da ação educacional é ensinar o conhecimento (conceitos e procedimentos) sobre o domínio e a forma de acessar este conhecimento. A fundamentação é a adoção do raciocínio baseado em casos (*Case-based reasoning*-CBR), que permite que o processo de ensino-aprendizagem aconteça através de problemas colocados e resolvidos pela adaptação de soluções previamente apresentadas de problemas similares; situações são entendidas pela adaptação de interpretações previamente mostradas de situações similares, bem como resolução de problemas através da exploração de soluções, de forma dirigida, indiretamente. Praticamente, raciocínio baseado em casos é uma técnica de IA que procura em uma base de casos por um caso mais próximo (análogo) para aplicar sua solução, no novo caso.

No modelo baseado em ambientes exploratórios estão todos os STI que desenvolvem suas técnicas instrucionais em ambientes de simulação de situações, de experiências e ambientes exploratórios. As atividades de modelagem e simulação estão entre as formas mais criativas de aplicações na Educação. A razão principal é que a construção e simulação de modelos formais têm muito a ver com o modo como o próprio conhecimento científico é gerado, através do ciclo: hipóteses, observação, experimentação, e formação de leis e teorias, em aproximações sucessivas [Sabbatini 1984]. Em segundo lugar, os sistemas computacionais

fornecem meios de cálculo e representação que tornam possível o uso didático da simulação, sem necessidade de habilidades especiais em matemática, etc. O computador agrega ainda um elemento de realismo que motiva o aprendiz. Os recursos como multimídias permitem fundir as tradições analíticas e empíricas na produção de materiais didáticos nos novos sistemas educacionais.

A modelagem e simulação podem ser usadas em uma grande variedade de contextos no processo ensino-aprendizagem. Segundo [Sabbatini 1984], para todos os níveis, mais especificamente e para aprendiz de menor idade, é comum o professor colocar programas prontos de simulação (levando em conta o fator tempo) de modelos relativos a fenômenos de interesse (por exemplo, em Física, Biologia,...). O aprendiz deve experimentar com o modelo, de modo a adquirir mais conhecimentos sobre o fenômeno em si. Neste caso, os níveis de conhecimentos prévios do aprendiz, a complexidade do modelo e os recursos de simulação determinam a melhor adequação da atividade ao currículo específico. A modelagem, que consiste em levar o aprendiz a construir total ou parcialmente o modelo subjacente, pode ser combinada com a simulação, apresentando um valor didático muito poderoso.

A técnica de simulação é muito usada em treinamento de profissionais para contextos e situações que acarretam grandes custos e/ou perigos para o desenvolvimento das habilidades necessárias; como por exemplo, os simuladores de tratamento de pacientes com risco de vida em Unidades de Tratamento Intensivo, simuladores de vôo, para pilotos de aeronaves tanto militares como da aviação civil, treinamento de cosmonautas, etc.

Os ambientes exploratórios ou ambientes de descobertas baseadas em simulações se diferenciam dos demais ambientes de aprendizagem na natureza do modelo do domínio, pois, na maioria das vezes, o domínio é modelado em termos quantitativos, para eficientemente executar cálculos das variáveis de simulações envolvidas, como também, normalmente, não utiliza características instrucionais.

Portanto, grande parte dos sistemas, só implementa uma única estratégia pedagógica , o que faz com que os STI's não tenham uma diversificação, tornando-se limitados neste aspecto.

Independente da estratégia pedagógica a ser adotada, o módulo de tutoria deve escolher a melhor ação, baseada em fatores do ambiente de aprendizado, natureza do conteúdo, conhecimento e vivência do aprendiz; para que com isso, o tutor não destrua a motivação pessoal do aprendiz ou seu senso de exploração.

II.2.3.Módulo de Domínio

Contém o conhecimento do especialista, que será mostrado ao aprendiz e atuará como base do conhecimento para questões, respostas e tarefas; e também como padrão para avaliação do desempenho do aprendiz.

Para o sucesso dos STI's seu domínio não deve apenas representar o "certo/errado" mas demonstrar o caminho até a sua resposta, de forma natural, como seria o processo compreendido pelo ser humano.

Várias abordagens para modelar o domínio e representar o conhecimento têm sido usadas, como por exemplo [Marietto 2000]:

- Modelo de Caixa Preta;
- Modelo Caixa de Vidro;
- Modelo Cognitivo.

II.2.3.1.Modelo de Caixa Preta

Este modelo de domínio representa as soluções através de modelos matemáticos. Não representa o raciocínio do especialista; com isso, não consegue explicar suas conclusões; apenas diz se os resultados estão "certos" ou "errados", e organiza as ações seguintes. Os STI's baseados no modelo caixa preta aplicam a tutoria tomando como base "as questões", que são respondidas pelo aprendiz oferecendo o retorno imediato.

II.2.3.2.Modelo Caixa de Vidro

Segundo [Marietto 2000] este modelo refere-se aos sistemas especialistas que são usados no Módulo do Domínio, ou seja, o modelo em questão originou-se de um sistema especialista que foi adaptado ao domínio a ser explorado. O modelo caixa de vidro comparando com o processo de busca gerado por estes sistemas são de extrema simplicidade carecendo de ajustes estruturais, sendo assim este modelo gera uma saída para o sistema, identifica como chegou àquela solução.

II.2.3.3.Modelo Cognitivo

Através deste modelo, pretende-se simular o processo cognitivo humano, modelando com mais exatidão os processos dinâmicos; saindo do foco de sistema especialista, que utiliza regras de inferência, para uma abordagem cognitiva, com 3 níveis de conhecimentos [Anderson 1988].

- Conhecimento declarativo;
- Conhecimento procedural;
- Conhecimento qualitativo.

O ***conhecimento declarativo*** é a forma de representar a maior parte do conhecimento do cotidiano das pessoas pois representa afirmações ou fatos sobre um mundo real. Existe uma forma de representar o conhecimento declarativo, que é através de *frames;* onde o conhecimento sobre um objetivo ou evento é armazenado de forma conjunta. Por fim, uma outra maneira de representar o conhecimento declarativo é através de *redes semânticas*, onde os objetos são representados pelos nodos, em grafos, e suas relações por arcos rotulados.

O ***conhecimento procedural*** é a representação do conhecimento em passos, a fim de diagnosticar problemas ou encontrar soluções, e isto é representado em forma de regras ou árvores de decisões.

O ***conhecimento qualitativo*** é o modelo usado para "pensar" sobre um determinado comportamento.

Na representação do conhecimento humano, a forma correta a ser empregada, dependerá do próprio conhecimento a ser representado, e de sua utilização pretendida.

II.2.4.Módulo de Interface

Nos sistemas interativos é de fundamental importância a definição de uma boa interface. Em STI's sua significância é muito maior, já que ela é o elo entre o material instrucional e o acompanhamento da evolução do aprendiz, através de suas respostas.

Algumas características da interface podem ser citadas:

- Evitar a desmotivação do aprendiz, através de recursos de mídia sofisticada para apresentação do material instrucional;
- Importância da bidirecionalidade na comunicação entre o tutor e o aprendiz;
- Tempo de resposta acessível ;
- Existência de um *log* para reconhecimento do aprendiz, evitando com isso questionários excessivos, para o conhecimento do *background* do aprendiz.

Boas interfaces de STI's devem evidenciar objetivos, entendimento e conceitos que são importantes para o aprendiz e o domínio. Por este fato, há necessidade em novamente salientar o estado mental do aprendiz. A interface passa a atuar como "ponte" do conhecimento cognitivo e o domínio da aplicação.Vale ressaltar a importância da aparência externa para o processo, já que a motivação do aprendiz pode ser aumentada com uma interface bem elaborada. Como objetivo na construção da interface devemos ressaltar os aspectos do *dinamismo* e da *inteligência* deste módulo, ou seja, a questão da adaptação deste módulo com o aprendiz. Já sua facilidade de interação constitui fator preponderante para o sucesso do diálogo entre o tutor e o aprendiz.

II.3. STI's COMO FERRAMENTAS EDUCACIONAIS E SUAS ESTRATÉGIAS DE ENSINO

Segundo [Leite 1999] do ponto de vista dos professores, um STI deverá servir de auxiliar para o docente, para que este possa acumular idéias pedagógicas e didáticas que serão experimentadas por seus aprendizes. Para o aprendiz ele será um parceiro competente, sendo assim os STI's envolverão professores e aprendizes em um processo de sócio-aprendizagem, ou seja, em um verdadeiro processo construtivista e cooperativista, onde o professor crie oportunidade de aprendizado e não apenas ensine. Já o aprendiz, consiga adquirir conhecimento (aprendizagem) aprendendo com o tutor e o computador, ou seja, um elemento agregador desses papéis e ações.

Os STI's criados com abordagem construtivista não pretendem dar liberdade total ao aprendiz, nem privá-lo de qualquer tipo de assistência diz [Giraffa 1999], fazendo com que os STI's oriente-o, mas sem desconsiderar seu background durante a interação. A forma que os STI's conduzirão estas intervenções caracterizam as estratégias pedagógicas, que visam a garantir a qualidade pedagógica adotada nos STI's, mas o uso de apenas uma estratégia não garante o sucesso para todos os aprendizes, já que cada estratégia tem vantagem específica e o uso da mesma depende de vários fatores, daí a importância da diversificação destas estratégias.

Segundo [Giraffa 1999] a seleção do conjunto de estratégias de ensino que constituirão o STI's são muito importantes para garantir a qualidade pedagógica do ambiente. Estratégias de ensino constituem as funções políticas de um sistema tutorial, pois dependendo da ordem e da maneira nos quais os tópicos de conteúdo são tratados e expostos, pode-se produzir experiências diferentes de aprendizado. Conforme foi mencionado por [Marietto 2000] o módulo tutor representa o conhecimento didático-pedagógico do sistema, conhecimento este que independe do domínio. No modelo proposto por [Leite 1999] muitos dos procedimentos ou estratégias de ensino estão inseridos no modelo de representação de conhecimento, que está alocado no módulo de domínio, cabendo assim ao módulo de tutoria apenas estratégias de cunho genérico, que podem ser habilitadas ou desabilitadas pelo especialista dependendo do uso a ser dado aos STI's.

Estratégias de ensino são implementadas como planos de como se deve processar as formas do ensino e da aprendizagem do aprendiz. Na formação de um plano de como ensinar determinado conteúdo do domínio, o tutor necessita do conhecimento das informações oriundas do diagnóstico, da monitorização sobre o comportamento, da interação com o aprendiz, e das suas características cognitivas, de modo a gerar uma seqüência de táticas capazes de conduzir a uma interação de ensino/aprendizagem, com relativo sucesso.

Conforme [Giraffa 1999] a seleção de uma estratégia depende de vários fatores, tais como: o nível de conhecimento do aprendiz, o domínio, a motivação e as características afetivas do mesmo. A utilização da mesma estratégia não produz um efeito satisfatório para todos os aprendizes.

Segundo [Marietto 2000] o módulo de tutoria leva o aprendiz a alcançar metas e para cumpri-las deve-se seguir objetivos específicos como:

- Estabelecer forma e seqüência de instrução;
- Determinar onde, quando e como deve intervir;
- Responder às questões feitas pelos aprendizes sobre o conteúdo apresentado;
- Escolher critérios de avaliação do desempenho do aprendiz;
- Realizar avaliações;
- Solicitar que seja alterado o modelo do , caso seja necessário.

O módulo de tutoria determina como e o que apresenta ao estudante, este solicita segundo [Marietto 2000] ao módulo do estudante que faça um diagnóstico do tutoriado e, com estas informações procure adaptar as instruções do módulo de tutoria.

Afirma [Frasson & Mengelle & Aimeur 1997] que cada estratégia tem vantagens específicas. É muito útil saber qual a estratégia que fortalecerá adequadamente o processo de aquisição para um determinado aprendiz. Professores competentes apresentam o mesmo material de modos diferentes. As situações em que isso pode ocorrer não são previsíveis (até mesmo se o domínio for previssível). O que se pode fazer é procurar diversificar o auxílio oferecido ao aprendiz, levando em consideração os indicadores retirados do próprio trabalho pregresso doaprendiz.

As estratégias de ensino devem incluir funções de orientação no desempenho das atividades, explanações dos fenômenos e processos, correções, bem como de adaptações específicas e individuais, gerando desafios, explicações, exemplos e/ou contra-exemplos no decorrer das interações.

A partir de um paradigma ou teoria de ensino/aprendizagem explícita ou implicitamente assumida, um plano de ações didáticas é determinado, constituindo-se em táticas sobre como, quando e o quê trabalhar com o aprendiz para que surja o processo de ensino/aprendizagem, isto é, as técnicas de instrução.

Um STI pode se utilizar de diferentes técnicas instrucionais ou de ensino, porém deve exibir sempre no mínimo três características [Halff 1988]:

1.ter controle sobre o currículo;

2.ser capaz de responder às questões do aprendiz referentes ao conteúdo em estudo;

3.ser capaz de determinar as necessidades de auxílio do aprendiz.

Muitos dos sistemas citados na literatura apresentam mais de uma característica de classificação. Isto acontece porque os projetistas dispõem de várias metas para construir o sistema. Eles consideram os princípios instrucionais diversificados, os métodos diferentes para estruturação do conhecimento, os vários formatos instrucionais para áreas de assuntos diferentes.

II.4. COMO MODELAR O APRENDIZ NOS TUTORES ATUAIS ?

Como modelar o aprendiz nos tutores atuais é a parte crucial na construção de um tutor, pois é através da modelagem forte ou fraca segundo [Giraffa 1999] que se define a taxionomia dos STI's, ou seja, se o modelo do aprendiz for fraco, os STI's são assistentes caso contrário, os STI's são tutores. Existe ainda uma importância muito maior em modelar corretamente um aprendiz, no que diz respeito ao uso da estratégia pedagógica a ser adotada, já que esta é individualizada e dinâmica, pois depende do diálogo do aprendiz com os STI's.

Modelar o aprendiz significa representar seus conhecimentos e suas habilidades cognitivas no momento da interação com os STI's, e é através deste modelamento e do conteúdo a ser aprendido que se escolhe a melhor estratégia pedagógica a ser utilizada em seguida pelos STI's. É através desta dinâmica que o tutor vai avaliar o desempenho do aprendiz e interpretar seu modelo. [Giraffa 1999] expõe algumas técnicas para construir o modelo do aprendiz tais como:

1. *incluir um reconhecimento de padrões aplicados ao histórico das respostas fornecidas por ele;*

2. *comparar a conduta do aprendiz com a de um especialista e verificar os pontos em comum;*
3. *acrescentar as preferências do aprendiz;*
4. *incluir seus objetivos particulares;*
5. *observar as coisas que o aprendiz sempre costuma esquecer quando interage com o tutor.*

Alguns modelos do aprendiz (os mais tradicionais) já foram apresentados neste trabalho (item II.2.1), porém, é importante salientar como serão armazenados dentro do modelo do aprendiz,ou seja, quais as variáveis do estado mental de cada aprendiz para que, com isso, seja possível moldar de forma personalizada cada instrução. Esta adaptação se baseia na determinação do atual estado de conhecimento do aprendiz sobre o domínio a ser tutoriado, representando assim os conteúdos e habilidades do aprendiz para formar o estado mental do mesmo. Além dos fatores citados anteriormente, diz [Marietto 2000] que o estado mental é aplicado também a seus traços de personalidade. [Correa 1994] apresenta uma sociedade de agentes que capta a crença, desejo e expectativa do aprendiz. Sendo assim a modelagem pode ser vista como o componente do STI que acompanha o comportamento do aprendiz e faz inferências sobre o seu conhecimento a respeito de determinado assunto (domínio) e como ele está ligado ao processo instrucional. Estas informações são usadas, posteriormente, para auxiliar o tutor em suas tomadas de decisões.

Daí pode se afirmar que o modelamento do aprendiz deve ser capaz de procurar dinamicamente o processo de aquisição do conhecimento por parte do aprendiz, ou seja, identificar seu estado mental atual. Desta forma [Marietto 2000] afirma que pela ausência de uma teoria capaz de explicar como se dá a aprendizagem humana, na prática a modelagem do aprendiz está muito distante de cumprir o papel de modelar os estados mentais do aprendiz.

Este trabalho visa, através do Aprendizado por Reforço, a abordar a modelagem do aprendiz com o objetivo de fornecer subsídios para o módulo de tutoria, para definição da estratégia pedagógica mais adequada a cada perfil do aprendiz. Com isso, deve-se refletir sobre o modelo do aprendiz, que para obter resultados não precisa ser explícito. O que deve estar bem definido é a estratégia, se esta é boa para um dado perfil do aprendiz, ou seja, que leva o aprendiz a estados cognitivos mais significativos, a técnica de AR conseguirá encontrar a política pedagógica como conseqüência, e indicará para o módulo de tutoria.

Capítulo III

APRENDIZAGEM AUTÔNOMA

III.1. APRENDIZAGEM AUTÔNOMA

Aprendizagem Autônoma é empregada em sistemas em que um processo dinâmico a ser controlado sofre modificações estruturais ao longo do tempo, ou quando a modelagem do processo é difícil. A Aprendizagem Autônoma tem capacidade de adaptação a situações novas, para as quais não foi fornecido todo o conhecimento necessário com antecedência, realizando assim uma aprendizagem a partir das próprias experiências. Um modelo de interação com o processo dinâmico externo (ambiente) para o problema de Aprendizagem Autônoma está ilustrado na Figura III.1.

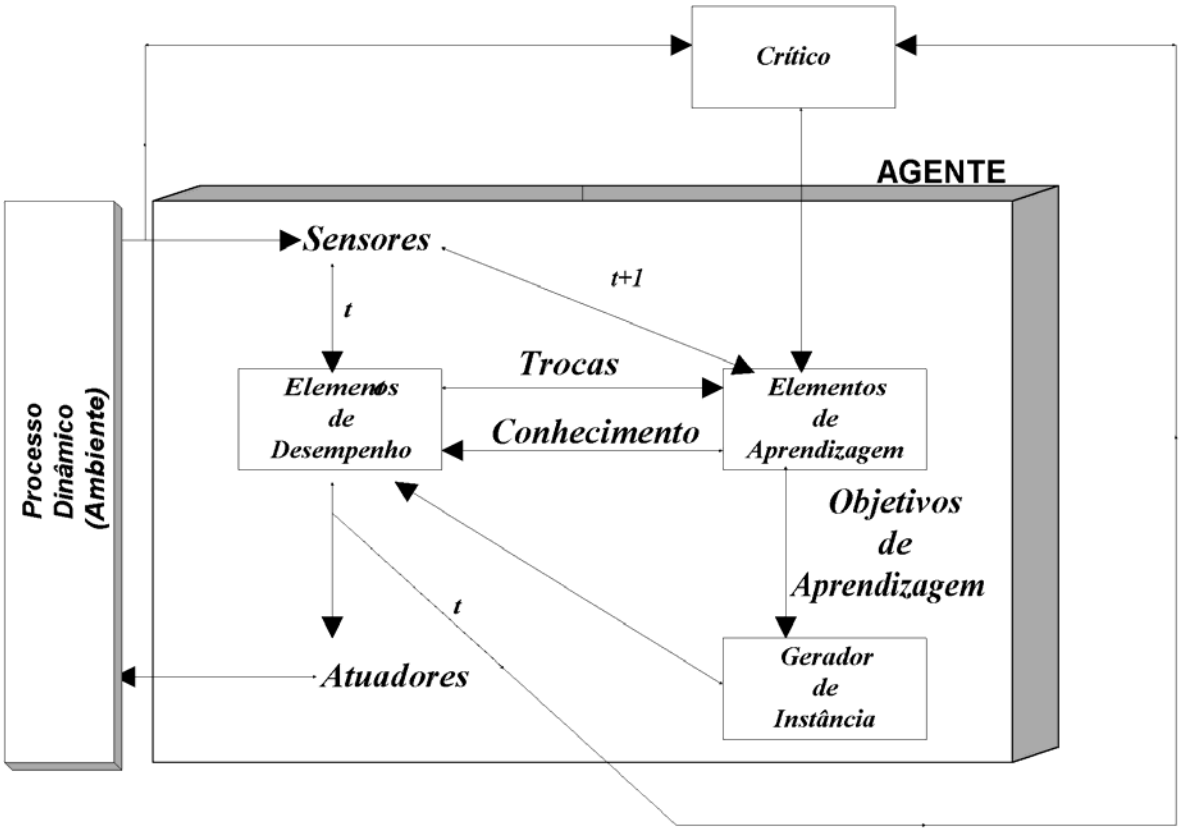

Figura III.1 – Modelo Dinâmico e seus Elementos

O modelo contém os seguintes elementos:

O **agente** relaciona-se com o ambiente via percepção (**Sensores**) e ação. Em cada passo da interação o agente recebe indicações do estado atual do ambiente através do crítico e escolhe uma ação para gerar como saída (**Atuadores**).

O **crítico** comunica ao elemento de aprendizado quão bem ou mal o agente está operando de acordo com algum critério. Isto é necessário porque observações sensoriais, em si, não são indicativos de desempenho.

Elemento de aprendizado é o processo pelo qual parâmetros do agente (armazenados no elemento de desempenho) são modificados. O elemento de aprendizado armazena instruções sobre como a modificação dos parâmetros deve ser feita. Essas modificações de parâmetros têm como objetivo melhorar o desempenho do sistema, tal como medido pelo crítico.

Elemento de desempenho executa as ações do agente, passa parâmetros do agente para o elemento de aprendizagem, recebe estes parâmetros dos sensores do gerador de instância e passa as ações para o agente através dos atuadores.

O **Gerador de instâncias** sugere ações alternativas (são passadas pelo elemento de aprendizado) que podem ser tomadas pelo agente, no qual são passadas para o elemento de desempenho.

Sensores possibilitam que o agente observe o comportamento do processo. Estas observações são transformadas em representações inteligíveis para o agente e podem corresponder a estados completos (seria ideal) ou parciais.

Atuadores são mecanismos pelos quais o agente atuará no mundo exterior. Os parâmetros recebidos pelos atuadores são passados pelo elemento de desempenho que modifica o estado do processo.

Uma função que especifica as ações combinadas do elemento de desempenho e do gerador de instância, em termos de observações ou tempo, é uma *política de ações.*

No processo dinâmico o agente tem como objetivo mapear uma política de ações que lhe permita obter a maior quantidade possível de recompensa segundo algum critério de comportamento ótimo. Desta forma, o projetista do sistema deve obter uma maximização ou minimização dos reforços obtidos pelo agente a longo prazo. Dependendo da qualidade de informação(Reforço) fornecida pelo crítico, pode-se ter alguns tipos de aprendizado, como por exemplo: Aprendizado Supervisionado, Aprendizado Não-Supervisionado e Aprendizado por Reforço. De acordo com as definições.

- na **Aprendizagem Supervisionada**, a informação de retorno (*feedback*) especifica a ação desejada em relação ao "aluno" e o objetivo da aprendizagem é aproximar-se desta ação desejada o máximo possível;
- na **Aprendizagem Não-Supervisionada**, nenhum *feedback* explícito é fornecido e o objetivo é encontrar ações úteis e desejadas através de um processo de auto organização;
- na **Aprendizagem por Reforço**, o *feedback* especifica apenas o reforço associado à ação realizada, e o objetivo é maximizar uma função dos reforços obtidos ao longo do tempo..

Na aprendizagem autônoma associa-se um reforço às condições críticas do sistema. Define-se um algoritmo (*Q-Learning* é o algoritmo de aprendizado autônomo que será usado neste trabalho, descrito no item III.1.2.1) capaz de propagar reforços ao longo do tempo, e executa-se este algoritmo várias vezes sobre o sistema. O processo dinâmico pode ter características estocásticas, que forçará a adoção de um enfoque probabilístico para o estudo do algoritmo de aprendizado.

III.1.1.Processos Decisórios de Markov -PDM

Na propriedade de Markov o estado atual representa uma sumarização dos estados passados. Segundo Sutton e Barto[Sutton & Barto 1998], pode-se predizer qual será o próximo estado e a próxima recompensa esperada dado o estado e ação atuais. A observabilidade perfeita destes estados garante que o processo sempre será de Markov.

Um Processo Decisório de Markov (PDM)é definido por uma quádrupla {S,A,P,R}.
Sendo assim um PDM consegue modelar o ambiente e mostrar a formalização do problema de Aprendizado Autônomo onde, contém os seguintes elementos:
Um conjunto finito de estados **S**.
Um conjunto finito de ações **A**.

Uma função de recompensa **R** esperada ao mover-se para **s'** é $\mathbf{R^a_{ss'}}$ **,** onde **Novo Estado –** $\mathbf{E\{r_{t+1}| s_t = s, a_t=a, s_{t+1}=s'\}}$**.**
Uma função de probabilidade do novo estado **s'** é $\mathbf{P^a_{ss'}}$, onde **Probabilidade-Pr**$\mathbf{\{s_{t+1} =s'| s_t = s, a_t=a\}}$.

A figura III.2 mostra a transição no Processo Decisório de Markov com seus elementos.

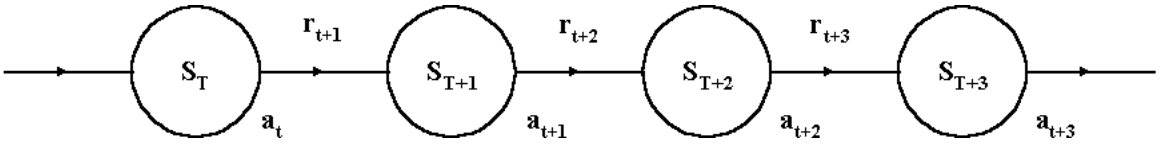

Figura III. 2 – Modelo de Transição PDM

III.1.1.1. Papel do Crítico no Processo Decisório de Markov -PDM

A função do crítico é produzir, para cada par (estado, ação) do PDM, um valor real (reforço $r(s_t, a_t)$), que é um indicativo do resultado instantâneo da evolução do PDM e pode ser não-determinístico. O reforço produzido é instantâneo (pouco significativo, pois precisam de um horizonte mais a longo prazo para avaliar) e pouco informativo de caráter estocástico, ou seja, enfoques probabilísticos, estando normalmente associados a uma condição dramática do processo. Conforme mostra a Figura III.1 o crítico não faz parte do PDM, ele é um crítico externo ao processo, pois pode ser projetado de forma independente, pelo projetista, desde que este não escolha valores com reforços infinitos, sendo assim, o critério de observação será do agente para as políticas estacionárias μ.

III.1.1.2. Critérios de Desempenho para Utilidade de Estado

A equação de Bellman (3.1) [Bellman 1957] fornece um método, para calcular as utilidades para cada um dos estados, sem precisar calcular as utilidades como reforços associados ao sair do estado inicial até chegar ao estado objetivo. Com isso, têm-se a soma descontada. A utilidade que está sendo calculada, embora a equação relacione utilidade com estado e com a utilidade do estado sucessor. O $\mathbf{V^{\mu}}$ **(s)** (para qualquer estado) é o reforço obtido numa política μ mais o reforço obtido do sucessor de **(s)** seguindo a mesma política, obtendo assim, a seguinte equação (3.2).

$$V^{*}(s) = E\{ r(s,\mu^{*}(s)) + \gamma \quad V^{*}(s')\} \quad (3.1)$$

$$V^{\mu}(s) = E_{\mu}\{R_t | s_t = s\} \qquad (3.2)$$

Para cada possível política μ que o agente pode adotar, define-se uma função de avaliação sobre os estados, onde r_t, r_{t+1},... são gerados seguindo-se a política μ, começando-se pelo estado **s**.

III.1.1.3. Maximização da Utilidade dos Estados

O agente precisa estabelecer uma política de ações que maximize o critério de desempenho da utilidade dos estados, ou seja, uma política ótima. A política é ótima quando consegue produzir o *maior valor de utilidade* possível para todos os possíveis estados de (***s***).

Formalmente, esta política ótima μ^* deve ser:

$$V^{\mu^*}(s) = V^* = \max_{\mu} V^{\mu}(s) \quad (3.3)$$

A política ótima μ^* é a melhor política que o agente pode seguir, para a utilidade de um estado ***s*** qualquer, seguindo esta política μ^*, é a maior utilidade entre todos os V(s) para todas as possíveis políticas μ.

III.1.2. Técnica de Aprendizado por Reforço

Afirma [Hendley1992] que o uso de técnicas de Inteligência Artificial na construção de software educacionais torna-se cada vez mais importante, uma vez que seu estudo fará com que os mesmos tenham mais qualidade.A arquitetura descrita neste trabalho utiliza uma técnica de Inteligência Artificial que é o Aprendizado por Reforço (AR), para modelar o aprendiz de forma dinâmica e autônoma. AR permite ao agente adquirir uma capacidade de conhecimento do ambiente que não estava disponível em tempo de projeto [Sutton & Barto 1998]. AR é baseada na existência de um crítico externo ao ambiente, que avalia a ação tomada, mas sem indicar explicitamente a ação correta.

Formalmente, AR utiliza uma estrutura composta de estados, ações e reforços conforme mostra a Figura III.3.

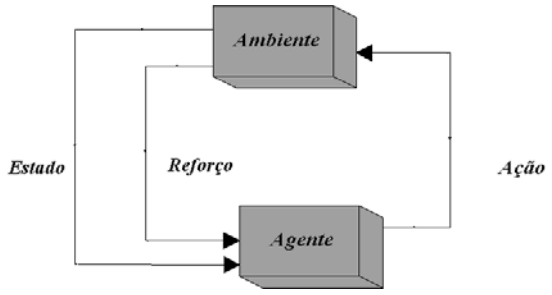

Figura III. 3 – Um agente AR interagindo com seu ambiente.

O agente existe em um ambiente descrito por um conjunto de possíveis estados e pode executar, para cada estado, uma ação dentro de um conjunto de ações possíveis, recebendo um valor de reforços a cada vez que executa uma ação. Este reforço indica o valor imediato desta transição estado-ação. Isto produz uma seqüência de pares estado-ação, e seus respectivos valores de reforço. O objetivo do agente é aprender uma política que maximize a soma esperada destes reforços a longo prazo.

Uma consideração importante a ser feita quando se lida com AR é a questão da exploração. O agente deve lidar com o problema de haver um compromisso em escolher a exploração de estados e ações desconhecidas, de modo a coletar nova informação, ou a explotação de estados e ações que já foram aprendidos e que irão gerar altas recompensas, de modo a maximizar seus reforços acumulados [Bellman 1957]. Sendo assim, de um lado, o agente deve aprender quais ações maximizam os valores das recompensas obtidas no tempo, e, de outro, deve agir de forma a atingir esta maximização, explorando ações ainda não executadas ou regiões pouco visitadas do espaço de estados. É importante, portanto, estabelecer uma política mista de explotação e exploração *(ε-greedy*), que inclua a escolha intencional de se executar uma ação que não é considerada a melhor no estado atual, com o objetivo de se adquirir conhecimento de estados ainda desconhecido ou pouco visitados. A escolha desta ação ocorre de forma randômica. Em uma política de explotação pura (*greedy*) escolhem-se as ações que se julguem (talvez erroneamente, caso o algoritmo de AR ainda esteja longe da convergência) serem as melhores para resolver o problema.

No contexto deste trabalho, a técnica de AR terá como função modificar parâmetros e armazená-los em "esquemas de planos", que definem a forma de apresentar o material instrucional ao aprendiz. O grande desafio, neste caso, é escolher a melhor ação que - baseada

no estado do aprendiz - possa mudar a estratégia de ensino, para obter resultados significativos, fornecedores de parâmetros indicativos de quão boa ou ruim está sendo uma determinada estratégia. O algoritmo utilizado (*Q-learning*) é baseado em estimativas de utilidades de pares estado-ação, e com estas estimativas pode-se gerar alternativas de novas estratégias pedagógicas. Neste caso, a aprendizagem das estratégias mais adequadas, que indicam a ação a escolher para cada estado do aprendiz, dar-se-á por reforço, através de uma estrutura de parâmetros adaptativos sobre os quais o algoritmo opera.

AR procura aproximar uma função que define a utilidade relativa dos pares estado-ação. Esta função de utilidade fornece indicações estimativas, mapeando os pares estado-ação em uma medida baseada na soma dos reforços esperados a longo prazo.

Para cada par estado-ação (s, a) é definido o reforço r(s, a), indicando uma conseqüência imediata da execução da ação a no estado s. O problema em AR é achar uma política ótima de ações (μ^*), ou seja, um conjunto de ações que maximizem, para cada estado s, os valores de utilidade Q(s, a). Com base nesses valores, o algoritmo de AR estima a ação de maior valor de utilidade – a ação que deverá ser executada pelo agente tutor.

III.1.2.1. Algoritmo Q-Learning

O algoritmo Q-Learning foi proposto em 1989 [Watkins 1989], para definir a escolha da melhor ação, no Aprendizado por Reforço. No Q-Learning a escolha de uma ação é baseada em uma função de utilidade que mapeia estados e ações a um valor numérico.

Neste trabalho, o valor de utilidade **Q(s, a)** de um par (estado**(s)**, ação**(a)**), é calculado a partir de reforços medidos pela qualidade do estado cognitivo do aprendiz (módulo de diagnóstico, que será proposto formalmente no Capítulo IV). Portanto, o principal objetivo do algoritmo Q-Learning é estimar autonomamente, em cada estado que o aprendiz se encontra, a ação com maior valor de utilidade. Como conseqüência, conseguirá estimar o estado cognitivo do aprendiz.

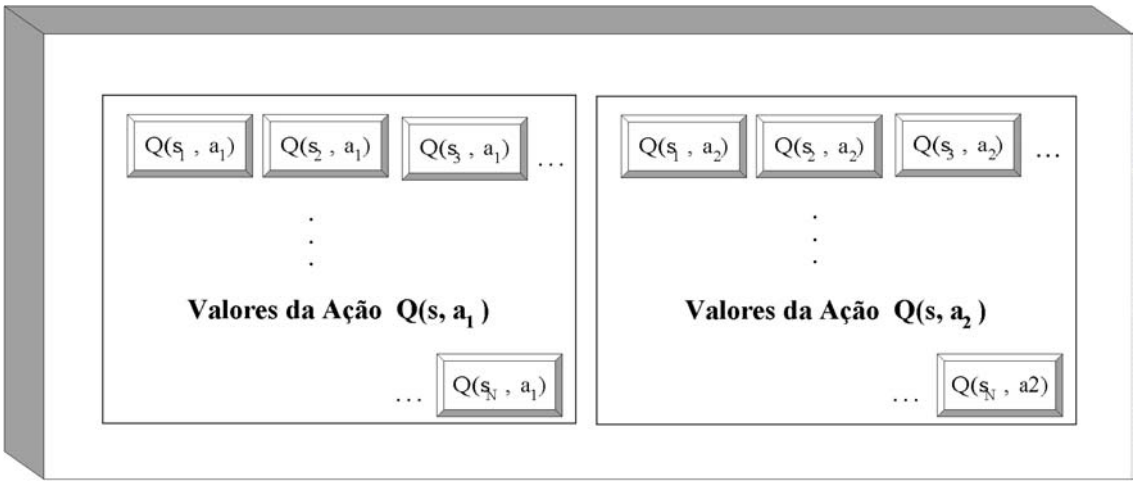

Figura III. 4 – Tabela dos pares (estado - ação).

O Q-Learning produz uma atualização dos valores **Q** na direção $\mathbf{max_a Q^{\mu}_t(s_{t+1}, a)}$, originando o seguinte **algoritmo**:

Inicialize **Q(s, a)**.

Para cada instante **t** repita:

1- Observe estado $\mathbf{s_t}$ e escolha ação $\mathbf{a_t}$ de acordo com a política de ações **(μ)**;

2- Observe o estado $\mathbf{s_{t+1}}$ e atualize $\mathbf{Q^{\mu}_t(s_t, a_t)}$ de acordo com

$$\mathbf{Q^{\mu}_{t+1}(s_t,a_t) = Q^{\mu}_t(s_t,a_t) + \alpha_t[r(s_t) + \gamma max_a Q^{\mu}_t(s_{t+1},a) - Q^{\mu}_t(s_t,a_t)];}$$

até **t** igual a limite de passos.

Onde podemos definir:

- $\mathbf{Q^{\mu}_{t+1}(s_t,a_t)}$**-** É o valor (qualidade) da ação $\mathbf{a_t}$ no estado $\mathbf{s_t}$, seguindo a política de ações **(μ).**
- $\mathbf{r(s_t)}$**-** É o reforço imediato recebido no estado $\mathbf{s_t}$.
- α**-** É a taxa de aprendizagem.
- γ**-** É a taxa de desconto.
- **t-** É uma seqüência discreta de passos no tempo, ou seja , t=0,1,2,3,...
- $\mathbf{max_a Q^{\mu}_t(s_{t+1},a)}$**-** Política de maximização, que escolhe na tabela (Figura III.4) a ação com maior valor de utilidade no estado futuro.
- O fator γ**(entre 0 e 1)**- quanto mais perto de 1**,** mais importância é dada aos reforços mais distantes no tempo.

No início, o algoritmo vai estar muito longe da utilidade ótima associada ao estado s, mas com o passar do tempo as estimativas de Q melhoram. De fato, os valores Q atualizados pelo algoritmo *Q-Learning* convergem para os corretos, desde que os pares (s,a) tenham sido visitados um número infinito de vezes [Watkins and Dayan 1992]. Na prática, convergência para políticas de ação adequadas é obtida apenas com exploração do espaço de estados, durante um número razoável de iterações. A política de ações durante a execução do algoritmo nas fases iniciais de treinamento deve, portanto, garantir a exploração do espaço de estados (política ε-*greedy*). Além do *Q-learning*, existem vários outros métodos [Dietterich 1996][Kaelbling 1995] como:

- Average Reward Reinforcement Learning
- Actual Return Reiforcement Learning
- Feudal Q-Learning
- Compositional Q-Learning
- Hierarchical Q-Learning
- Recurrent Q-Learning

A razão pela qual o *Q-Learning* é um dos mais usados e pela qual justifica sua utilização neste trabalho é porque, para aprender a utilidade dos estados, seria necessário uma descrição bastante detalhada do ambiente, o que nem sempre é possível (neste trabalho seria o estado cognitivo de cada aprendiz a ser tutoriado). Para este tipo de caso considera-se o *Q-Learning* um método mais "econômico", porque baseia-se apenas em valores de qualidade para cada ação em cada estado, muitas vezes mais fáceis de se determinar.

III.2. Estudo comparativo de métodos de modelagem do aprendiz em STI

O módulo do estudante não mais relaciona somente as informações sobre a análise das interações do aprendiz com o domínio, mas busca uma contextualização maior destas interações em função das ações do aprendiz, o contexto em que elas ocorrem e a estrutura cognitiva do aprendiz naquele momento. Alguns sistemas podem monitorar as atividades dos aprendiz de perto, adaptando suas ações as respostas dos aprendiz mas nunca cedendo o controle da interação.

O Tutor Espertinho de Konzen [Konzen & Frozza 2000], tem como primeiro passo na utilização do sistema a determinação do perfil do aluno. Este perfil influenciará na forma de exposição do conteúdo e é determinado através de um questionário que determinam o nome, preferência de cores e características de interação (se o aprendiz possui um perfil visual ou auditivo). Estas informações restringem a adaptação do tutor à determinação de fatores sobre a interface, e que por sua vez são estáticos, já que não se alteram durante a sessão de trabalho.

Observa-se que os STI clássicos, de uma forma geral, se caracterizam por utilizar - sem proporcionar uma adaptação dinâmica (em nível de estratégias) - as características individuais de aprendizagem de cada aluno.

Em função do desenvolvimento da IA distribuída, e a crescente pesquisa em Sistemas Multiagentes - SMA, a tecnologia de agentes passou a ser adotada como uma metodologia alternativa para modelar o aprendiz nos STI's. Segundo [Giraffa 1999], a razão fundamental para modelar STI com uma arquitetura multiagente é sua capacidade de comunicação e interação. Agentes podem se adaptar e aprender durante uma sessão. Segue abaixo a exemplificação de modelagem do aprendiz em alguns STI usando SMA.

O sistema LeCS [Rosatelli 1999] tem como objetivo dar suporte à aprendizagem colaborativa através da Internet utilizando o método de Estudos de Casos. Este método tem como objetivo ensinar aos alunos conteúdos baseados em situações complexas e reais. São apresentadas situações problema onde o aprendiz deverá buscar uma solução. A falta de uma estrutura definida no domínio permite que o aprendiz desenvolva uma flexibilidade cognitiva para que possa lidar com tais situações. O sistema LeCS é composto por uma sociedade de agentes, composta por: agentes de interface, agente de informação, agente de aconselhamento e um agente facilitador, cabendo ao agente facilitador a função de modelagem do aprendiz, onde este armazena em uma base de crenças informações sobre aprendiz individualmente: o passo da metodologia em que cada aprendiz está trabalhando, a resposta conjunta do grupo para caso (esta informação é armazenada em apenas um agente de um grupo de aprendiz). O

agente prepara e ordena as requisições do aprendiz em uma estrutura de dados interna, monitorando suas atividades.

O sistema ITStrategic de Marietto[Marietto 2000] tem como objetivo definir as estratégias instrucionais do STI de forma dinâmica, visando aumentar a adaptabilidade do sistema ao perfil do aluno O ITStratégic é composto por cinco módulos, o módulo percepção, módulo estudante, o módulo tutoria e o módulo entrega. O módulo estudante é formado pelo agente Guia, que é responsável pela ativação ou criação de um modelo de aprendiz, bem como atualizar e disponibilizar este modelo à sociedade de agentes.

O sistema LAHYSTOTRAIN [Oswaldo & Angélica 2000] tem como objetivo o treinamento cirurgiões por meio de um Ambiente com Realidade Virtual (utilizando personagens), Treinamento Teórico e tutoria para resolução de exercícios. O sistema é composto dois sistemas, um simulador VR (Realidade Virtual) e o ATS (*Advanced Traning System*). O ATS é divido em seis subsistemas: Assistente, Tutor, Cirurgião assistente, Enfermeira, Anestesista, Estudante e a Interface de usuário. Há também uma base de dados associada ao log de uma sessão. Cabe salientar que o subsistema Estudante não é considerado um agente mas uma interface com usuário. O modelo de usuário (Interface do usuário) é divido em estático e dinâmico. O estático contém informações sobre preferências, informações pessoais, nível de experiência e o conhecimento teórico prévio do aprendiz. O modelo dinâmico inclui o desempenho do aprendiz, exercícios realizados, erros cometidos e o conhecimento sobre patologias e instrumentos.

No Explanation Agent Zouaq [Zouaq & Frasson & Rouane 2000] propõe o Agente de Esclarecimentos (Explanation Agent) que tem como principal objetivo prover repostas ou explicações sobre o conteúdo com maior qualidade, identificando problemas que possam ocorrer durante o processo de explicação ou resolução de problemas. O agente possui as características de reatividade, autonomia, aprendizagem, mobilidade, dedução e capacidade de compreensão. A mobilidade permite que o agente no início de uma sessão se mova do *servidor de cursos* para máquina do aprendiz, o que torna o código móvel do agente o mais atualizado possível. A capacidade de aprendizagem e dedução permite que ele identifique interpretações errôneas, padrões de comportamento relacionados ao perfil do aprendiz que são relativos a determinados tipos de erros.

Framework JADE [Silveira & Bica & Viccari] propõe uma metodologia e um *framework* numa arquitetura multiagente para criação de ambientes educacionais distribuídos, JADE (*Java Agents for Distance Education Framework*). O *framework* Jade disponibiliza os recursos necessários para o projeto e construção de STI. Sua arquitetura é baseada em uma

sociedade de agentes onde existe o agente Gerenciador de modelos de aprendiz: responsável pela construção e manutenção do conhecimento sobre cada aprendiz conectado ao sistema; A arquitetura interna de cada agente do *framework* é composta por uma estrutura cognitiva (baseada em estados mentais e compõe a base de conhecimento do agente) e habilidades (particulares do agente e de comunicação).

Em AME-A **(**Ambiente Multiagente de Ensino-Aprendizagem) o modelo do aluno é representado através de 16 perfis definidos, onde um agente identifica (no início do curso através de um questionário, e durante analisando as interações do aprendiz). Desta forma o perfil do aprendiz pode ser alterado dinamicamente. O agente Seleciona estratégia analisa as mudanças de perfil e seleciona a estratégia mais adequada.

O sistema MarCo (Mediator Artificial de Conflitos) por Tedesco [Tedesco & Self 2000] é um ambiente colaborativo para resolução de problemas baseado em um modelo computacional de resolução de conflitos metacognitivo. Este modelo permite que o ambiente infira a respeito dos conflitos do grupo de aprendiz e interfira na busca de melhores interações.O sistema é composto por: uma interface distribuída, pela qual os aprendizes interagem entre si, elaboram uma solução para um dado problema e recebem o auxílio do sistema; Servidor Prolog, que processa as informações oriundas da interface.

A tarefa de monitorar o ambiente (sistema) e o(s) aprendiz(es) durante as interações, podem trazer uma sobrecarga ao sistema como um todo, e comprometer sua performance (desempenho). Porém este conjunto de informações é fundamental para o auxílio personalizado ao(s) aprendiz(es).

Capítulo IV

USO DA TÉCNICA DE APRENDIZADO POR REFORÇO EM UM TUTOR INTELIGENTE

IV.1. INTRODUÇÃO AO APRENDIZADO DE MÁQUINA

Aprendizado de Máquina - AM - é uma sub-área de pesquisa muito importante em Inteligência Artificial - IA -, pois a capacidade de aprender é essencial para um comportamento inteligente. AM estuda métodos computacionais para adquirir novos conhecimentos, novas habilidades e novos meios de organizar o conhecimento já existente. O estudo de técnicas de aprendizado baseado em computador também pode fornecer um melhor entendimento de nosso próprio processo de raciocínio.

Pode-se dizer que uma máquina aprende sempre que se modifica a sua estrutura, o programa ou os dados de tal maneira que o desempenho futuro esperado da máquina melhore. Algumas dessas modificações, como a adição de um registro em uma base de dados, não são necessariamente um aprendizado. Mas, por exemplo, quando uma máquina de reconhecimento de voz progride depois de ouvir várias amostras da fala de uma pessoa, trata-se de um caso em que a máquina está aprendendo.

O AM é importante pelos seguintes motivos :

- Há tarefas que só podem ser definidas por exemplos, isto é, somente é possível especificar entradas e saídas, não sendo definido a relação entre a entrada e a saída.

- É possível existir relações e correlações escondidas em uma grande pilha de dados. Os métodos de aprendizado de máquina podem ser usados para extrair estas relações.

- Projetistas freqüentemente produzem máquinas que não funcionam como desejado no ambiente de trabalho. De fato, certas características do ambiente não poderiam ser completamente conhecidas em tempo de projeto. Os métodos de aprendizado de máquina podem ser usados para que existam projetos de máquina que evoluam progressivamente no trabalho.

- A quantidade de conhecimento avaliado em certas tarefas pode ser muito grande para ser codificado explicitamente. Máquinas que aprendem este conhecimento gradualmente podem ser capazes de capturar mais do que os humanos podem escrever.

- Os ambientes modificam-se todo o tempo. Máquinas que podem se adaptar com a modificação do ambiente, reduzem a necessidade de refazer os projetos.

No AM pode ser distinguidas algumas fases:

Treinamento: um conjunto de exemplos de condutas corretas (ou indicativos de conduta) é analisado, e alguma representação do conhecimento recentemente aprendido é armazenado,freqüentemente sob forma de regras ou alteração de parâmetros.

Validação: As regras são checadas e, se necessário, um treinamento adicional é dado.

Aplicação: As regras são usadas em respostas para alguma situação nova.

Estas fases podem não ser distintas. Por exemplo, podem não ter uma fase de validação explícita, uma vez que o algoritmo de aprendizado garante alguma forma de corretude. Também em alguma circunstância, o sistema aprende sobre o trabalho, que é um treinamento e aplicação, uma sobreposição das fases.

Atualmente já foram propostos diversos paradigmas de AM. Esta seção procura apresentar brevemente alguns desses paradigmas, tais como o paradigma simbólico, estatístico, Instance-based, conexionista e genético

Os sistemas de aprendizado simbólico buscam aprender construindo representações simbólicas de um conceito através da análise de exemplos e contra-exemplos desse conceito. As representações simbólicas estão tipicamente na forma de alguma expressão lógica, árvore de decisão, regras de produção ou rede semântica. Atualmente, entre as representações simbólicas mais estudadas estão as árvores e regras de decisão. Desenvolvimento original do programa para a indução de árvores de decisão é atribuído Morgan [Morgan & Messenger 1973]. Já os trabalhos com indução de regras de decisão surgiram com a simples tradução das árvores de decisão para regras, com a poda realizada sobre as regras, tal abordagem surgiu com Quinlan [Quinlan 1987].

No paradigma estatístico, técnicas estatísticas tendem a focar tarefas em que todos os atributos têm valores contínuos ou ordinais. Muitos deles também são paramétricos, assumindo alguma forma de modelo, e então encontrando valores apropriados para os parâmetros do modelo a partir de dados. Os classificadores estatísticos freqüentemente assumem que valores de atributos estão normalmente distribuídos, e então usam os dados fornecidos para determinar média, variância e co-variância da distribuição. Os classificadores estatísticos freqüentemente assumem que valores de atributos estão normamente distribuídos, e então usam os dados fornecidos para determinar média, variância e co-variância da distribuição.

No paradigma Instance-based,uma forma de classificar um caso é lembrar de um caso similar cuja classe é conhecida e assumir que o novo caso terá a mesma classe [Quinlan 1988].

No paradigma de conexionismo as redes neurais são construções matemáticas que foram inspiradas no modelo biológico do sistema nervoso. Sua representação envolve unidades altamente interconectadas, no qual o nome *conexionismo* é utilizado para descrever a área de estudo. As pesquisas em redes neurais foram iniciadas com o trabalho pioneiro de McCulloch e Pitts [Mcculloch & Pitts 1943]. Rosenblatt [Rosenblatt 1958] apresentou o Perceptron, modelo fundamental de rede neural artificial que originou várias variantes. Minsky e Papert [Minsky & Papert 1969] demonstraram a existência de limites fundamentais nos perceptrons de uma camada. Hopfield [Hopfield 1982] utilizou a idéia de uma função de energia para formular uma nova forma de compreender os cálculos realizados em redes recorrentes com conexões sinápticas simétricas.

O paradigma genético é um formalismo de classificação derivado do modelo evolucionário de aprendizado [Holland 1986]. Um classificador genético consiste de uma população de elementos de classificação que competem para fazer a predição. Elementos que possuem uma performance fraca são descartados, enquanto os elementos mais fortes proliferam, produzindo variações de si mesmos. Este paradigma possue uma analogia direta com a teoria de Darwin, onde sobrevivem os mais bem adaptados ao ambiente.

Devido a complexidade dos problemas reais, que normalmente são não lineares e dinâmicos, a combinação de paradigmas tem se mostrado forma comum de solução de problemas [Langley 1996].

IV.2. INTRODUÇÃO AO FUNCIONAMENTO de STI's COM AR

Foi vista no capítulo II a arquitetura dos Sistemas Tutores Inteligentes (STI's) como também suas aplicações e limitações. Já no capítulo III foi vista a técnica de Aprendizado por Reforço (AR) com uso do algoritmo Q-Learning. Neste capítulo abordará como é o funcionamento do sistema AR com STI's. O objetivo deste capítulo é apresentar as definições adotadas no texto, no que tange aos conceitos relacionados com o sistema de tutores inteligentes com aprendizado por reforço desenvolvido neste trabalho e por fim apresentar uma descrição geral do ambiente utilizando os estudos preliminares realizados para compor a arquitetura. Será apresentada em detalhes a arquitetura proposta para a modelagem de um STI e o uso da técnica de Aprendizado por Reforço.

IV.3. ESTRUTURA DO SISTEMA

A estruturação do sistema tem como objetivo facilitar o entendimento do funcionamento do sistema de AR no tutor inteligente desenvolvido. Este sistema irá modelar de forma dinâmica e autônoma o aprendiz, fazendo com que cada ação e o estado mental de cada aprendiz sejam representados em uma tabela de par estado-ação. A estrutura do sistema mostra também algumas partes que já são exploradas por tutores atuais, porém cabe, novamente, representá-las para que se entenda a dinâmica e o funcionamento do sistema por completo.

Salienta-se algumas mudanças em seu funcionamento como forma de melhoria e uma adaptação coesa ao sistema de AR. Estas modificações são propostas para obter uma maior eficiência do tutor em suas funções, criando assim algumas alternativas para os módulos do tutor; como por exemplo, possibilidades de adaptar múltiplas estratégias pedagógicas no módulo de tutoria; possibilita ainda uma maior adaptação do tutor com cada aprendiz devido à classificação e armazenamento do status do mesmo em associação com um *login*, para identificar o acesso de forma personalizada ao tutor. Com isso possibilita o retorno do aprendiz ao ponto ao qual parou, dando assim continuidade às ações interrompidas. Desta forma, o tutor terá a possibilidade de interagir com vários usuários, sem que seja necessário criar generalização de faixas, ou seja, classificar o aprendiz de forma genérica, sem levar em consideração seus estados mentais específicos.

A estrutura, com o uso do AR, mostra ainda como funcionará o módulo de diagnóstico ao qual este trabalho pretende propor a inclusão na arquitetura dos STI's, tornando assim o tutor

mais dinâmico, adaptável ao aprendiz e gerando possibilidades de alternativa para o módulo de tutoria .

A Figura 4.1 ilustra a estrutura de funcionamento da técnica de Aprendizado por Reforço para um STI. Todos os itens que fazem parte do sistema serão detalhados a seguir.

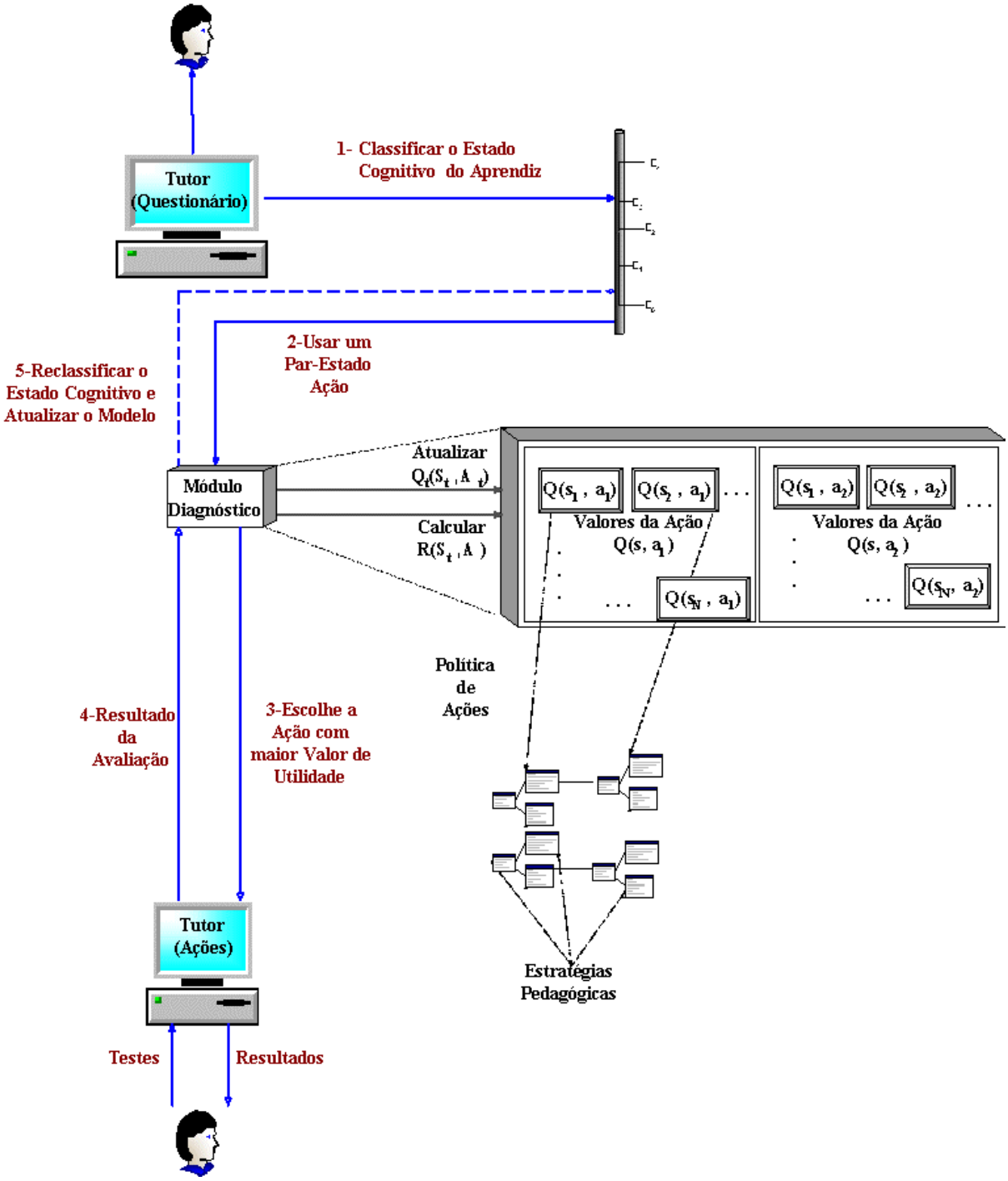

Figura IV. 1 – Sistema com a Técnica de AR.

IV.3.1. Inclusão na Arquitetura Tradicional dos STI's do Módulo Diagnóstico usando Técnica de AR

A proposta de inserção da técnica de AR no módulo diagnóstico é de suma importância pois ela definirá, através da qualidade de sua observação, como se dará o processo de ensino aprendizagem.

Este módulo diagnóstico produzirá - de acordo com a política de ações estabelecida – uma indicação de quão bem ou mal está sendo o desempenho do aprendiz em relação às ações produzidas pelo tutor. Ele enviará reforços, indicando o desempenho, de modo intermitente, produzindo para cada par (estado, ação) um valor $r(s_t, a)$ correspondendo a reforços positivos quando se produzem resultados favoráveis, de acordo com a política de ações. Os valores de utilidade $Q_t(s_t, a_t)$ são atualizados na tabela de valores Q mostrada na Figura III. 4 para que, com base nesses valores, o tutor possa reclassificar o aprendiz.

A complexidade de estabelecer um perfil está na modelagem do aprendiz. O perfil vai indicar o estado cognitivo do mesmo a partir do comportamento observável e, com isso, poder indicar uma estratégia de ensino a ser utilizada.Decisões pedagógicas razoáveis são fatores de desafio para os tutores, já que as pessoas nem sempre conseguem representar seus próprios processos mentais e, algumas vezes, ficam confusas.

A primeira interação com o tutor dar-se-á através de questionários, onde será conhecido o perfil do aprendiz, através da classificação do mesmo em faixas distintas de conhecimentos, como mostra a Figura IV.2. Com as atualizações ocorrendo na tabela par (estado-ação) o sistema irá reclassificar o aprendiz a cada novo ciclo, porém vai estar embasado nas ações que estão produzindo o maior valor,ou seja, ações que dado um estado cognitivo do aprendiz possa levá-lo para um estado cognitivo melhor que o anterior. É claro que no início estes parâmetros serão insuficientes para gerar boas ações, pois a regra principal para um sistema de AR é que haja um número grande de visitas a estes pares estado-ação. Quando isto ocorrer, o sistema estará indicando para o tutor a melhor ação para um dado estado cognitivo do aprendiz.

Como complementação, para construir um perfil do aprendiz deve-se manter um *log* de status do aprendiz, ou seja, do último estado cognitivo e da última ação realizada pelo tutor, para que ao retornar seja desnecessária a repetição de tais questionários e possibilite a continuidade do processo do ponto ao qual parou.

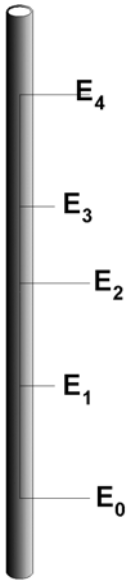

Figura IV. 2 – Classificação do Estado Cognitivo do Aprendiz.

IV.3.2. Funcionamento do Tutor com o Módulo Diagnóstico

Para melhor entendimento do sistema, abaixo é mostrado como funciona passo a passo:

1. O sistema classifica o estado cognitivo do aprendiz através do resultado do questionário, usando uma escala de classificação, Figura IV.2 e obtendo seu perfil.

2. Com o perfil definido, o sistema usa uma tabela de pares estado-ação. Nesta, as ações representam estratégias pedagógicas adotadas pelo tutor para guiar o aprendiz.

3. O sistema escolhe a ação com o maior valor de utilidade baseado nos resultados do algoritmo Q-Learning.

4. O tutor executa a ação junto ao aprendiz e obtém resultados.

5. Estes resultados são encaminhados para o Módulo Diagnóstico, que calcula reforços positivos ou negativos para o par estado-ação $r(s_t, a)$ e atualiza na tabela o valor de utilidade $Q_t(s_t, a_t)$.

6. Depois de atualizado o valor de utilidade na tabela, o sistema volta a reclassificar o aprendiz na escala e retorna ao passo 2.

A figura IV.3 mostra uma adaptação da estrutura formal da aprendizagem por reforço, para definir a interação entre o agente (Tutor) e seu ambiente (Aprendiz) com o uso do Módulo Diagnóstico que seria o elo entre o tutor(agente) e o seu ambiente(aprendiz).

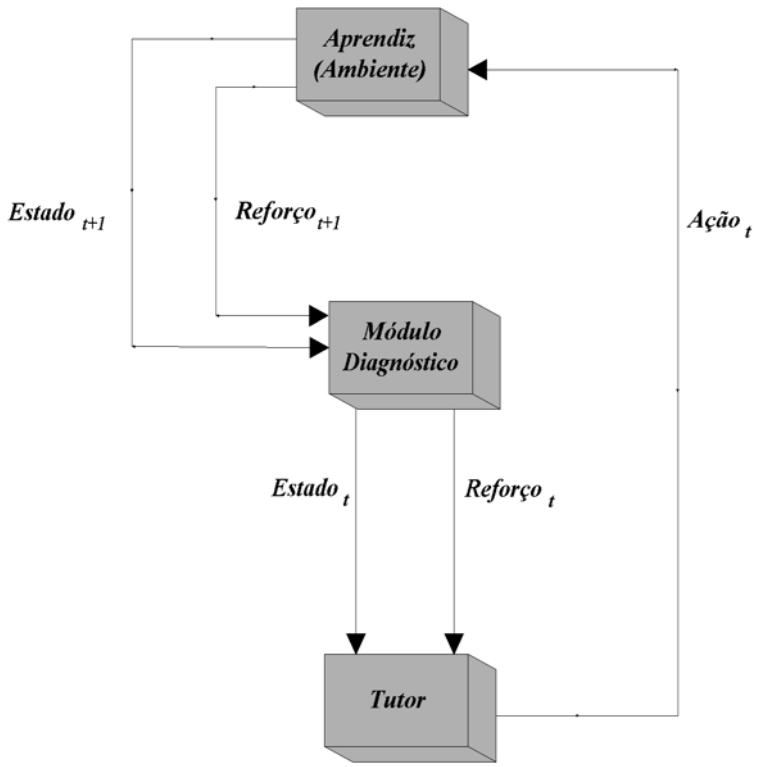

Figura IV.3 – Representação do Modelo do Aprendizado por Reforço adaptado ao Sistema de Tutoria.

Capítulo V

RESULTADOS E CONCLUSÕES

V.1.IMPLEMENTAÇÃO DO PROTÓTIPO

Para implementação do protótipo do modelo do sistema AR representado na Figura V.1, adotou-se a linguagem C e programação modular, que tem como principal objetivo simplificar o desenvolvimento da programação através da sua divisão em módulos. Cada um dos módulos pode ser mais facilmente entendido, programado, testado e modificado, pois o problema que está sendo resolvido torna-se menor.

O ponto básico para a programação modular é a abordagem de resolução de programas, através de sua decomposição, ou seja, adotando uma política de dividir para simplificar.

Assim, o problema original é dividido em vários subproblemas. O resultado final dar-se-á através da combinação das soluções dos subproblemas.

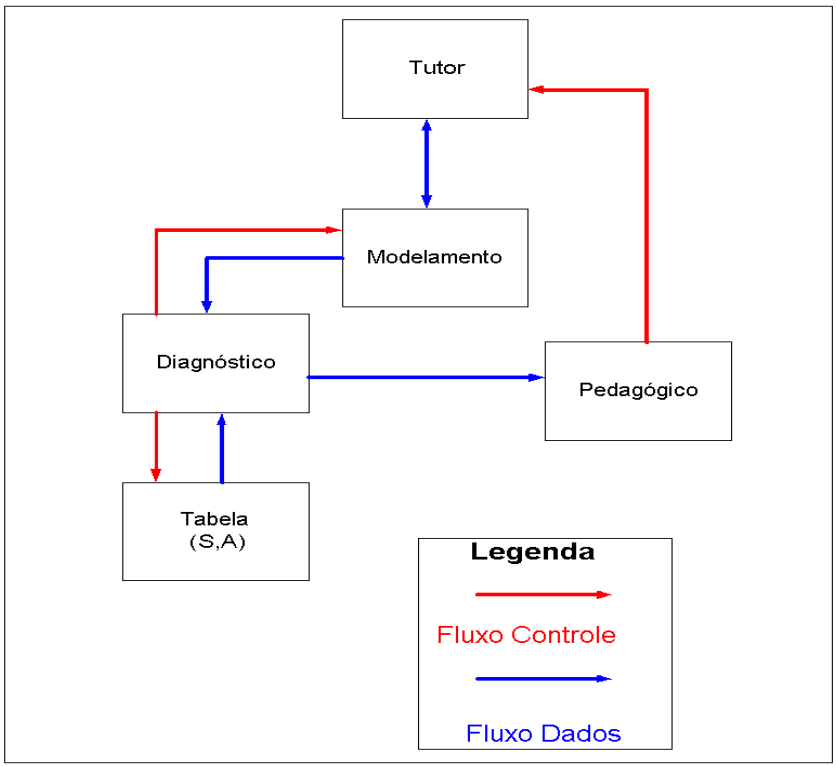

Figura V.1. – Identificação dos módulos do Sistema de Aprendizado por reforço para Modelagem Autônoma do Aprendiz em Tutor Inteligente.

V.1.1. Módulo do Tutor

Este módulo foi desenvolvido para se adaptar ao conteúdo pedagógico, tornando-se um tutor mais versátil. Para que isto possa ocorrer, são implementadas funções que possibilitem alterações dos conteúdos apresentados pelo tutor, no que visa a garantir sua manutenção, como também sua adaptabilidade às estratégias pedagógicas.

V.1.2.Módulo de Modelamento

O tutor, ao iniciar sua tutoria, realiza um questionário para avaliar o estado cognitivo do aprendiz sobre o conteúdo; do contrário, ele associa a senha com o último estado cognitivo do aprendiz (antes de sair tutor) e posiciona-o no conteúdo adequado. Durante a interação, o módulo diagnóstico fornece parâmetros para o módulo tabela (S,A), indicando o estado cognitivo do aprendiz, e este será reclassificado dentro de parâmetros previamente definidos, indicando para o módulo pedagógico o estado cognitivo do aprendiz.

V.1.3.Módulo Pedagógico

Este, tornou-se bastante flexível, ou seja, pode ser alterado ou acrescido de várias estratégias pedagógicas, tornando o tutor mais adaptável aos diversos estados cognitivos do aprendiz. No início, este módulo assumirá ações aleatórias, mas com o decorrer das interações, o módulo diagnóstico indicará as ações com maior valor de utilidade.

V.1.4. Módulo Tabela (S,A)

Receberá parâmetros do módulo diagnóstico e do módulo de modelamento que serão armazenados em uma estrutura de matriz.

V.1.5. Módulo Diagnóstico

É responsável pelo cálculo das utilidades Qs e atualização da Tabela $Q(s,a)$.

V.2. EXPERIMENTOS REALIZADOS COM O PROTÓTIPO

Os experimentos foram realizados sobre um protótipo de tutor, em um ambiente com uma matriz 5x10 mapeando estados e ações (5 estados e 10 ações) com elementos descritos a seguir.

- Um conjunto de estados S={E_0, E_1, E_2, E_3, E_4}, onde cada um representa um possível estado cognitivo do aprendiz, em face da interação com o tutor, ou seja, é o resultado obtido pelo tutor ao aplicar uma ação A_i em um dado instante de tempo $_i$. Com isso, estima-se que esse aprendiz tenha um grau cognitivo baseado nesses estados, onde:

E_0=> [0,2],
E_1=>]2,4],
E_2=>]4,6],
E_3=>]6,8],
E_4=>]8,10].

Os valores de 0 a 10 divididos nos cinco conjuntos em cada estado representam uma métrica usada para diferenciar os estados, esses valores representam o grau cognitivo do aprendiz, face as ações aplicadas a este, seguindo para isso uma escala crescente do grau evolutivo do aprendiz na interação com o tutor (Figura IV. 2).

- Um conjunto de ações A={A_0, A_1, A_2, A_3, A_4, A_5, A_6, A_7, A_8, A_9} que podem ser escolhidas pelo tutor. Cada ação pode corresponder à aplicação de provas, exercícios, questionários, perguntas, trabalhos, testes, etc, ou combinações destes e outros dispositivos de avaliação, usados pelo tutor, seguindo as estratégias pedagógicas estabelecidas.
- Um conjunto de Reforços Instantâneos associados a cada estado visitado, ou seja,

E_0=>R=1-Ruim;
E_1=>R=3-Regular;
E_2=>R=5-Bom;
E_3=>R=7-Muito Bom;
E_4=>R=10-Excelente

Foi definida uma metodologia de teste na qual foram criados três modelos determinísticos e não determinísticos. Segue abaixo as representações de cada um dos modelos:

a) Modelo M_1 (Ruim) deterministico, definido por um conjunto de estados e ações de acordo com o diagrama de transição de estados da Figura V.2.

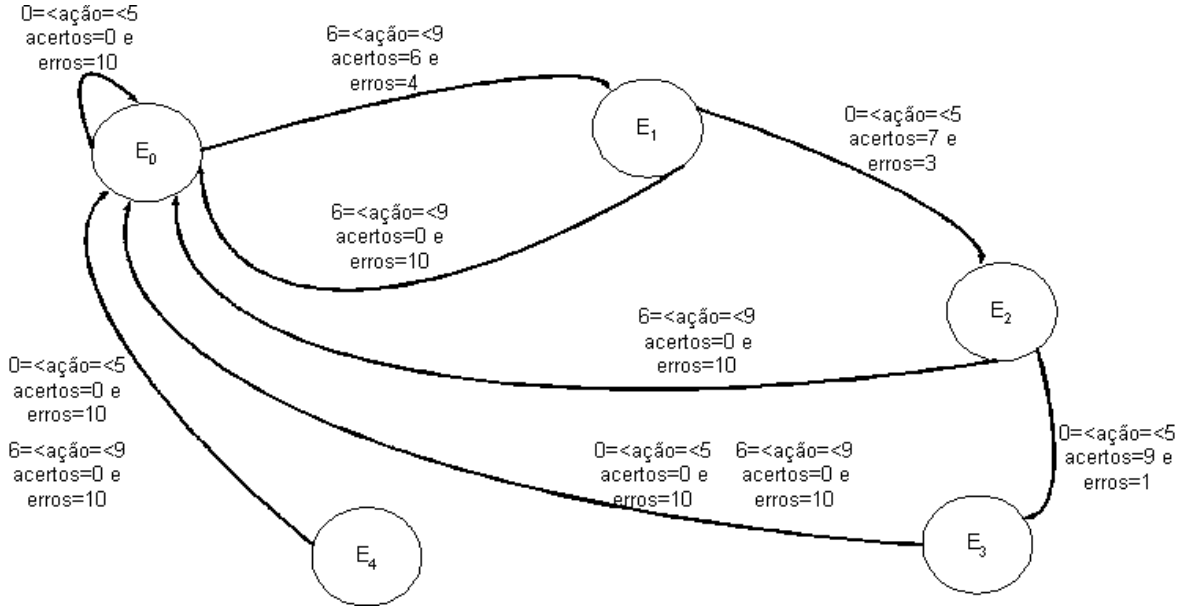

Figura V.2. – Modelo M_1 (Ruim) deterministico.

b) Modelo M_2 (Bom) deterministico, definido por um conjunto de estados e ações de acordo com o diagrama de transição de estados da Figura V.3.

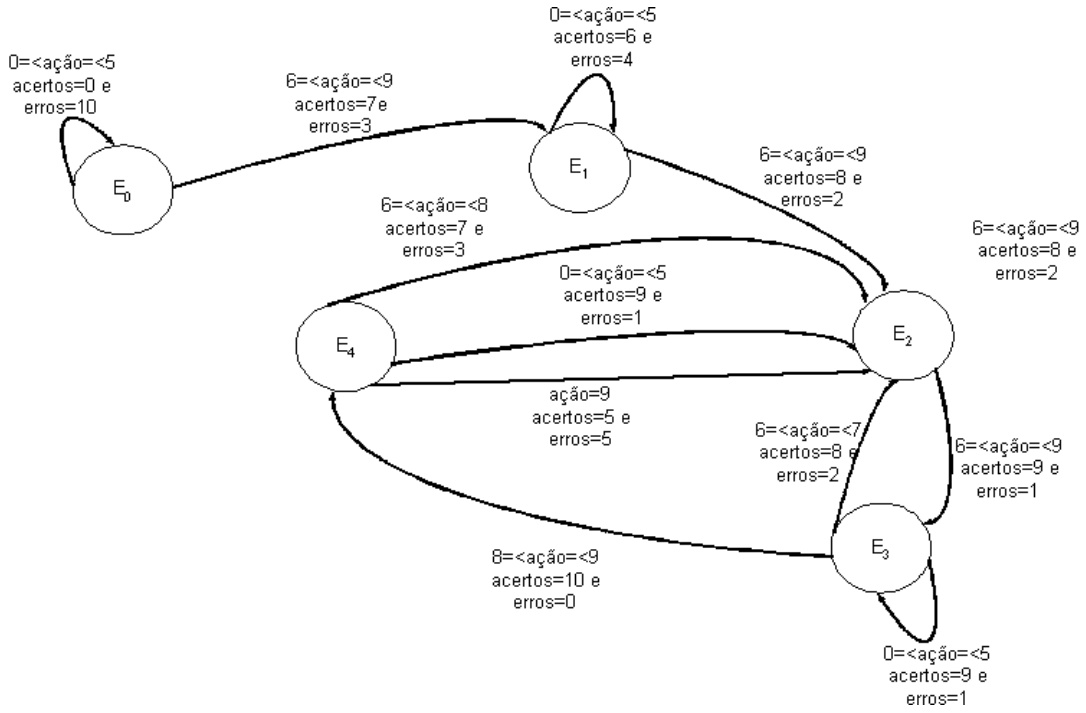

Figura V.3. – Modelo M_2 (Bom) deterministico

c) Modelo M_3 (Excelente) deterministico, definido por um conjunto de estados e ações de acordo com o diagrama de transição de estados da Figura V.4.

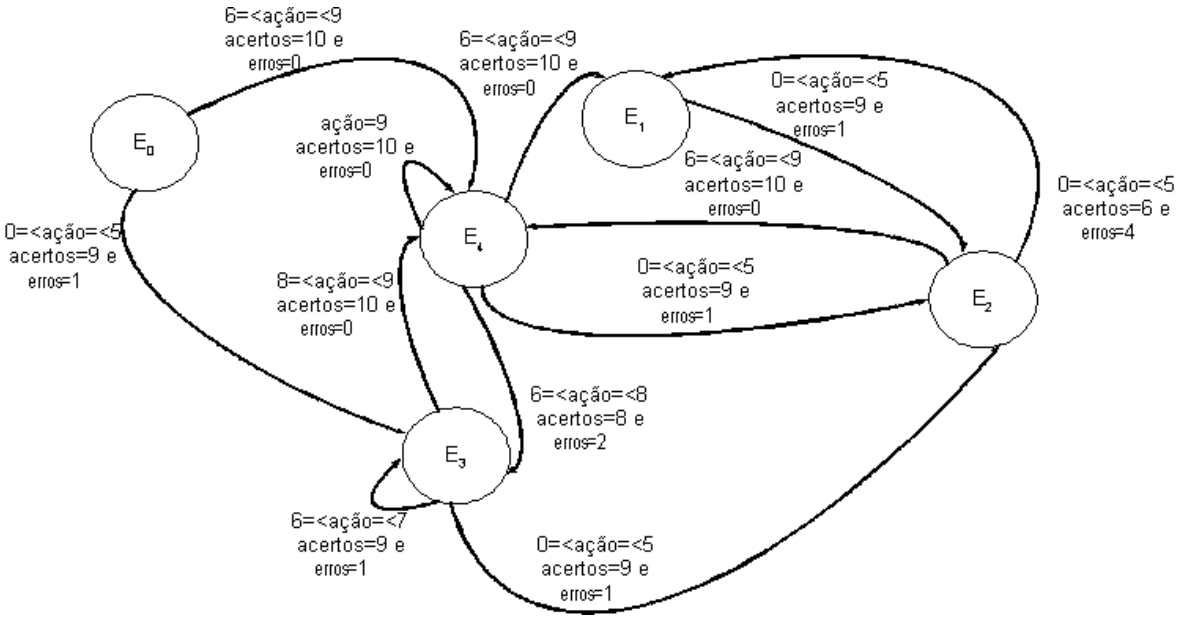

Figura V.4. – Modelo M_3 (Excelente) deterministico.

d) Modelo M_1 (Ruim) não-deterministico, definido por um conjunto de estados e ações de acordo com o diagrama de transição de estados da Figura V.5.

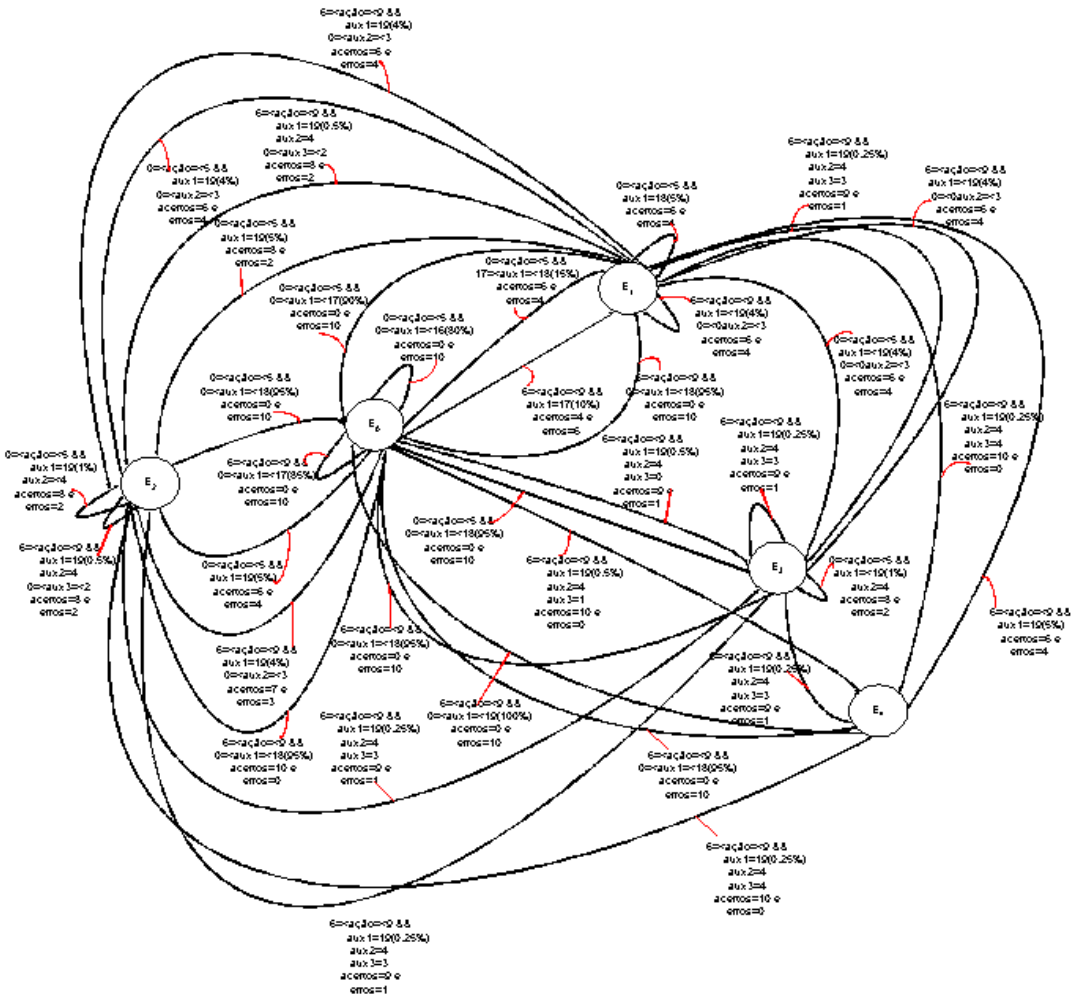

Figura V.5. – Modelo M_1 (Ruim) Não deterministico.

e) Modelo M_2 (Bom) não-deterministico, definido por um conjunto de estados e ações de acordo com o diagrama de transição de estados da Figura V.6.

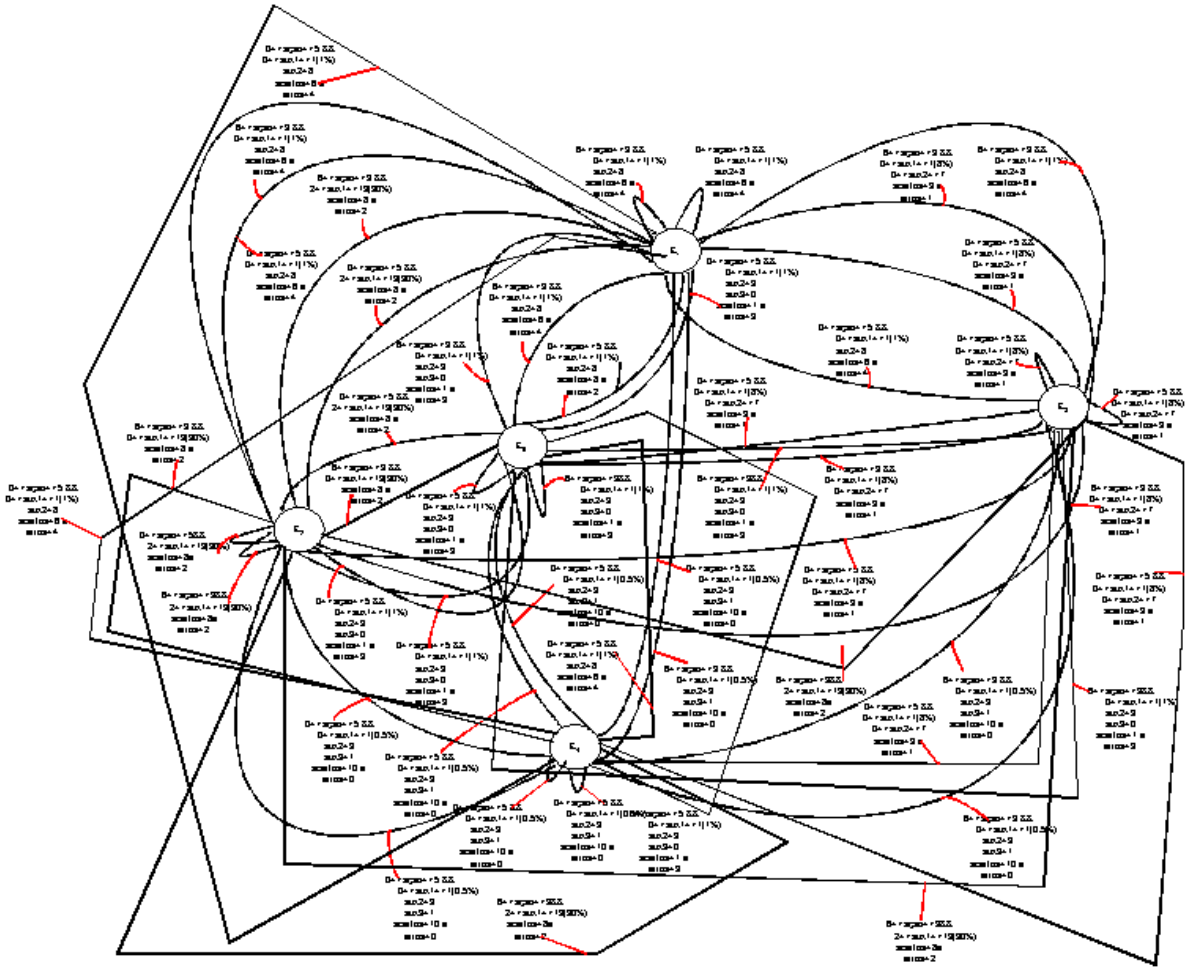

Figura V.6. – Modelo M_2 (Bom) Não deterministico.

f) Modelo M_3 (Excelente) não-deterministico, definido por um conjunto de estados e ações de acordo com o diagrama de transição de estados da Figura V.7.

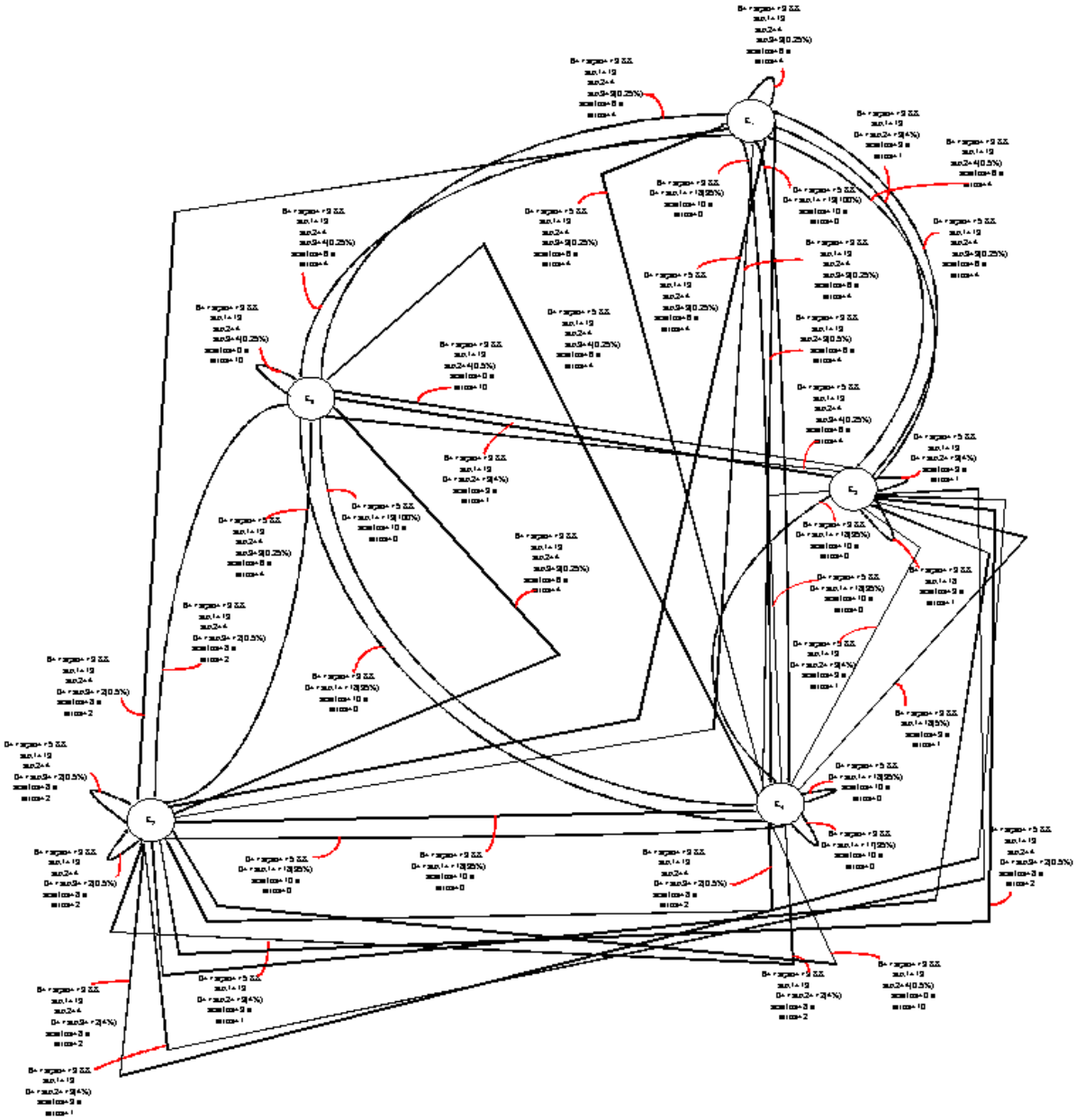

Figura V.7. – Modelo M_3 (Excelente) Não deterministico.

Finalmente, foram criadas duas políticas pedagógicas P_1 e P_2, onde P_2 é uma política mais restritiva do que P_1 em relação ao modelo, já que os intervalos entre as ações são menores, possibilita desta forma uma depuração maior das ações em relação aos estados, sendo assim haverá um número maior de decisões para cada estado com o uso da política P_2..

Com objetivo de mostrar como foram montadas estas políticas, são mostrados a seguir os diagramas de transição de estados correspondentes.

Para a política P_1 temos:

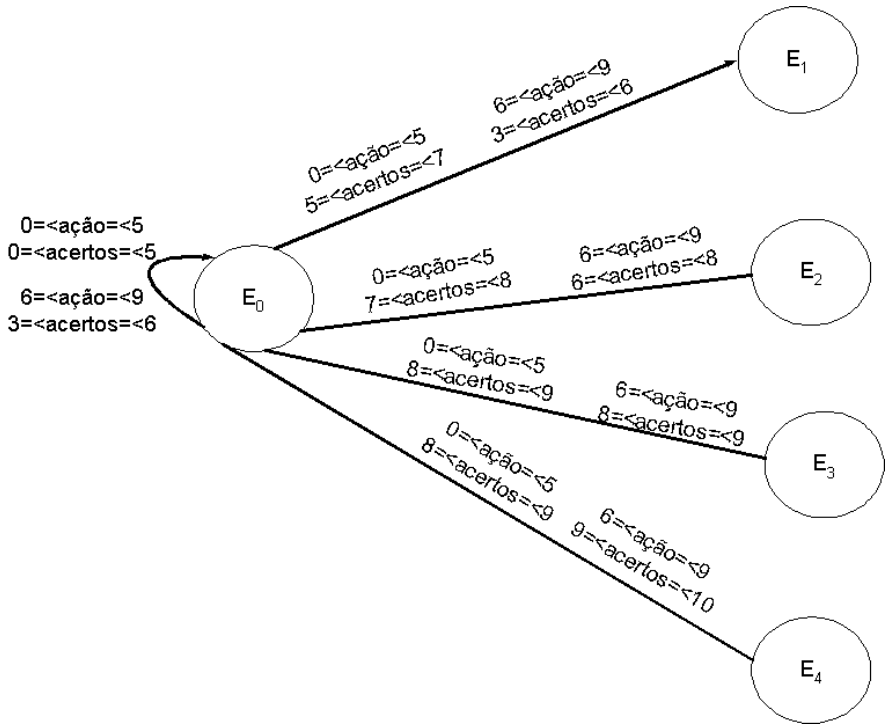

Figura V.8. – Política Pedagógica P_1 para o Estado E_0.

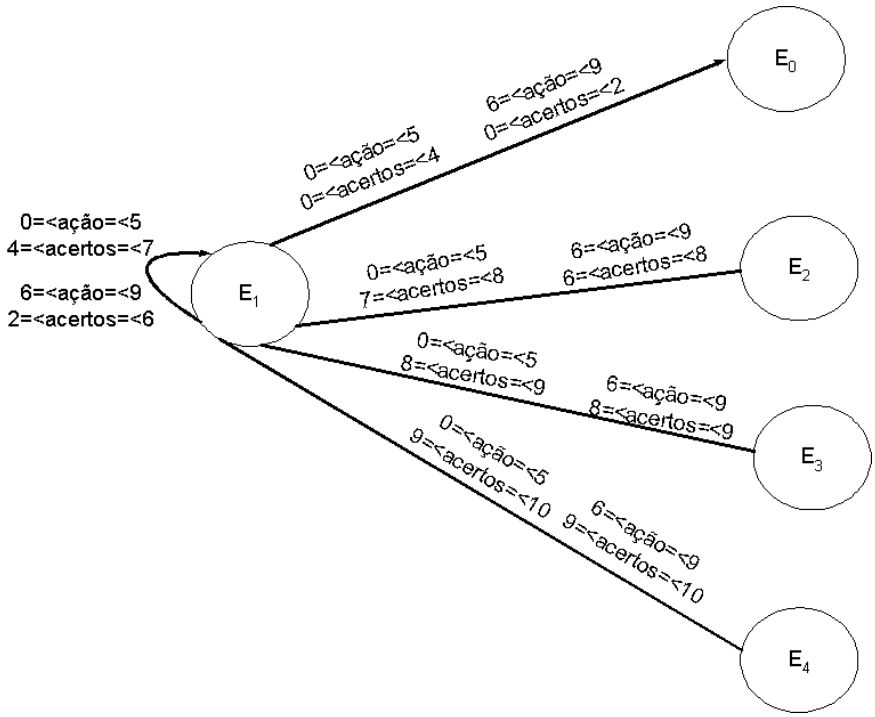

Figura V. 9. – Política Pedagógica P_1 para o Estado E_1.

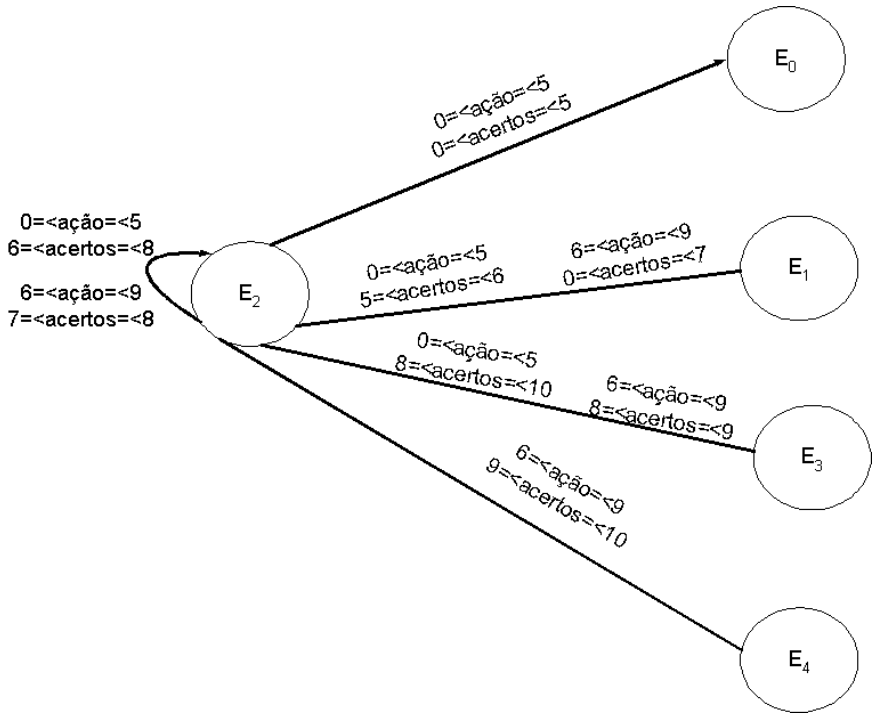

Figura V.10. – Política Pedagógica P_1 para o Estado E_2.

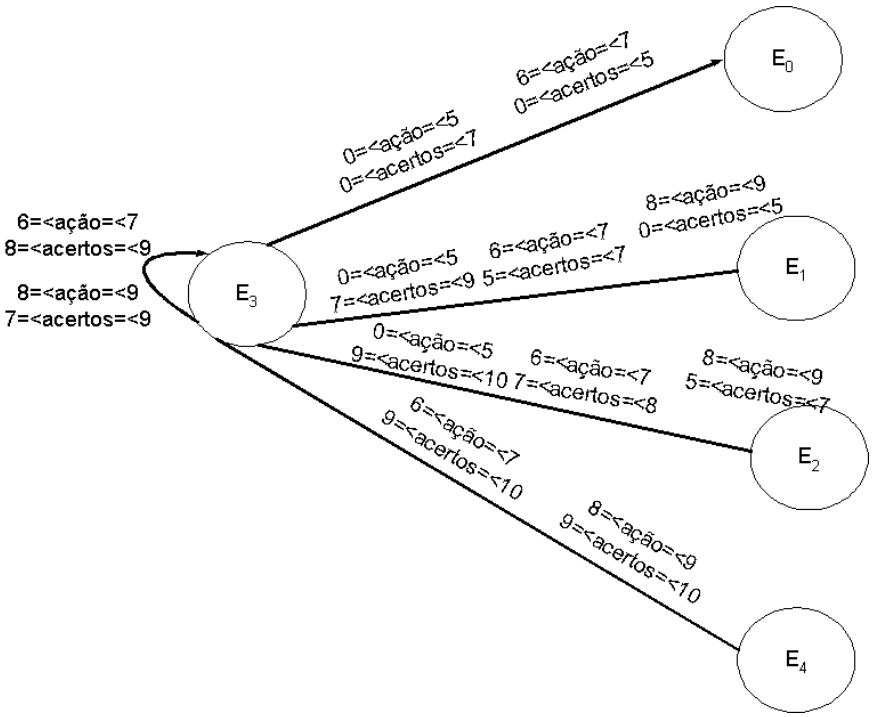

Figura V.11. – Política Pedagógica P_1 para o Estado E_3.

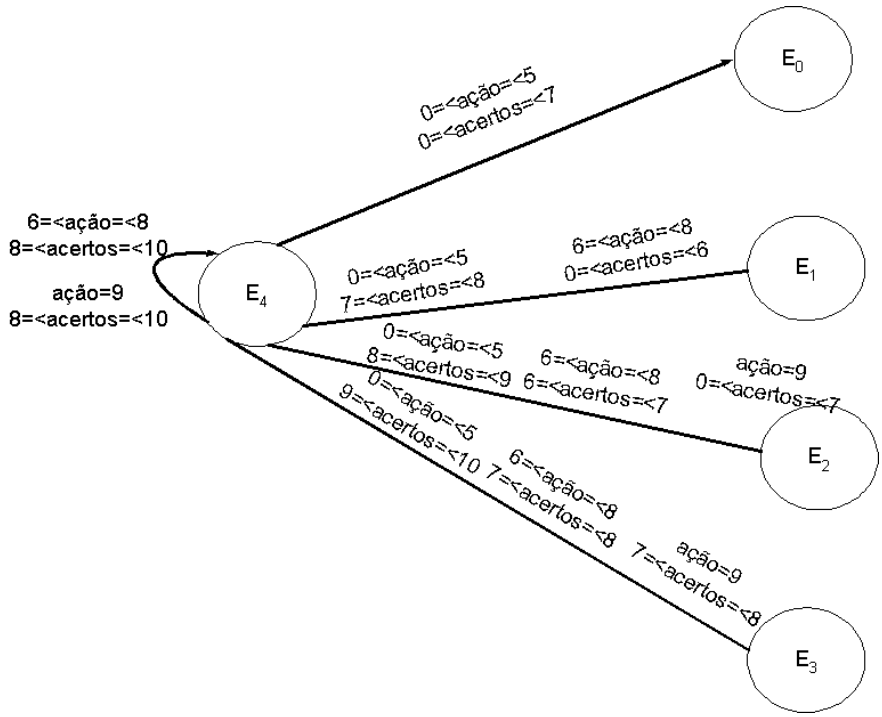

Figura V.12. – Política Pedagógica P_1 para o Estado E_4.

Para política P_2 temos:

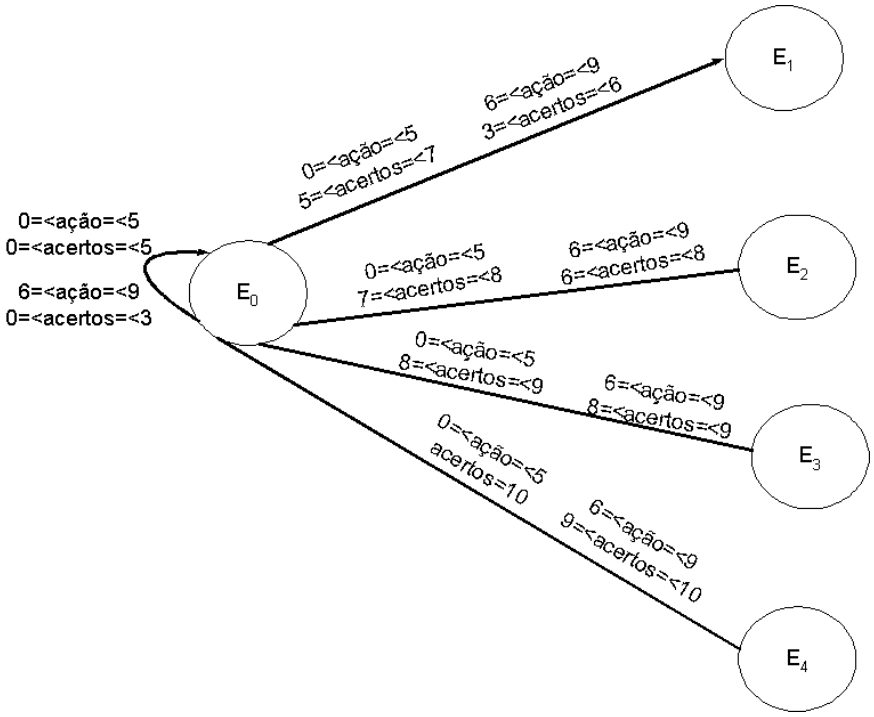

Figura V.13. – Política Pedagógica P_2 para o Estado E_0.

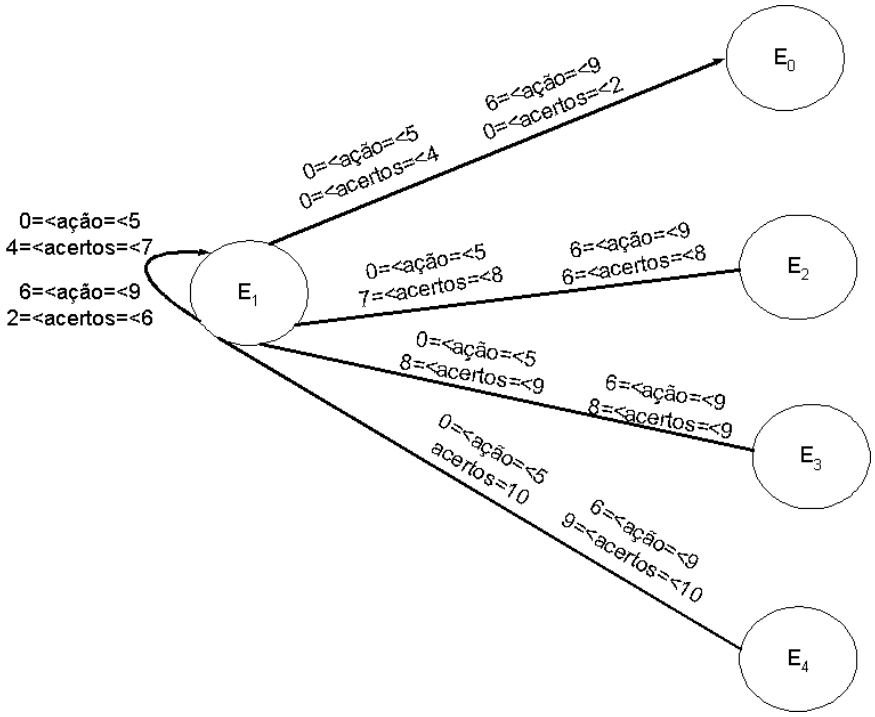

Figura V.14. – Política Pedagógica P_2 para o Estado E_1.

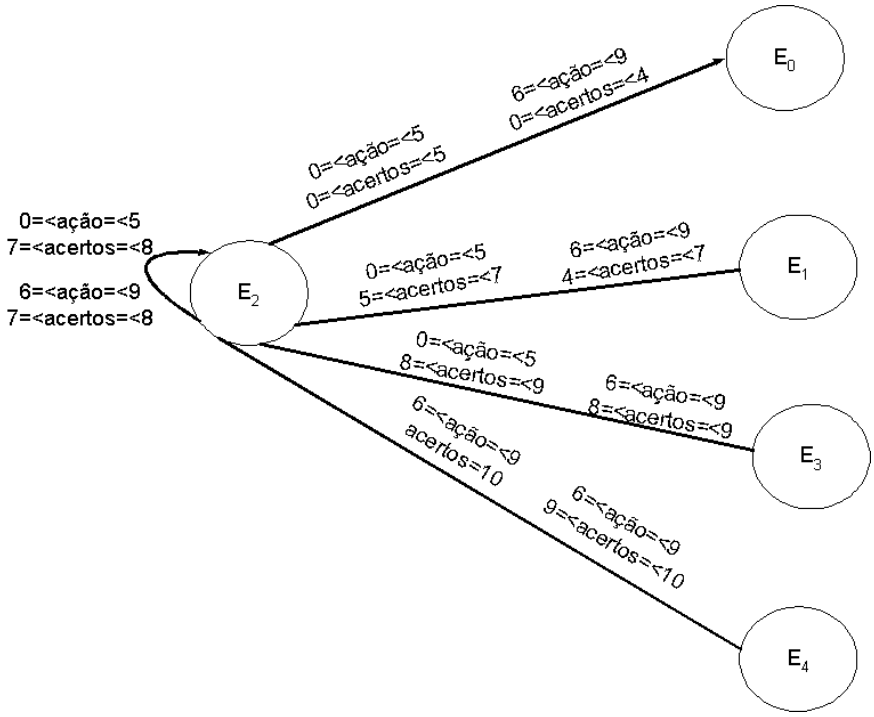

Figura V.15. – Política Pedagógica P_2 para o Estado E_2.

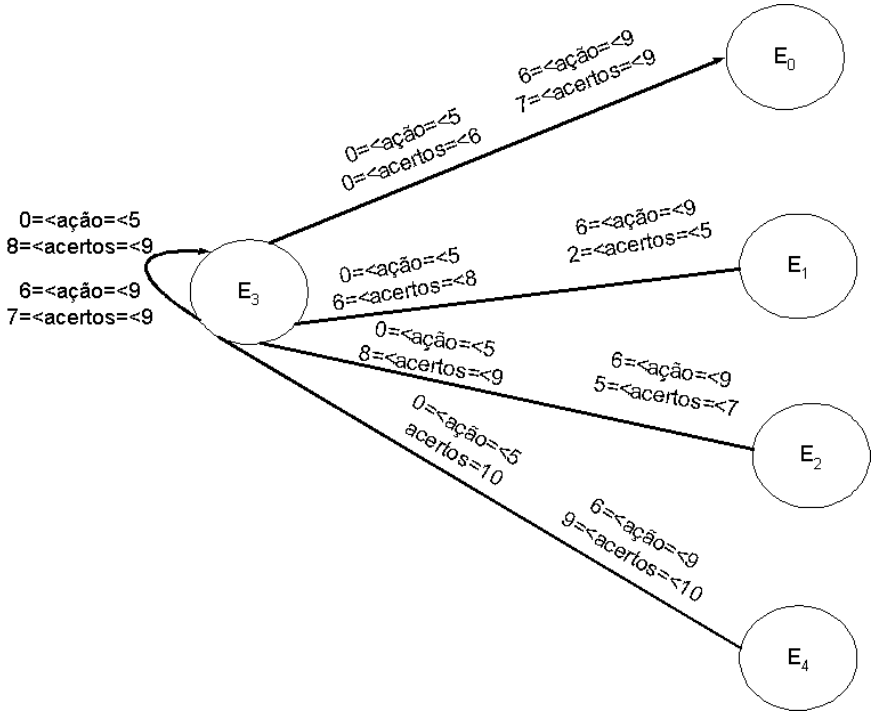

Figura V.16. – Política Pedagógica P_2 para o Estado E_3.

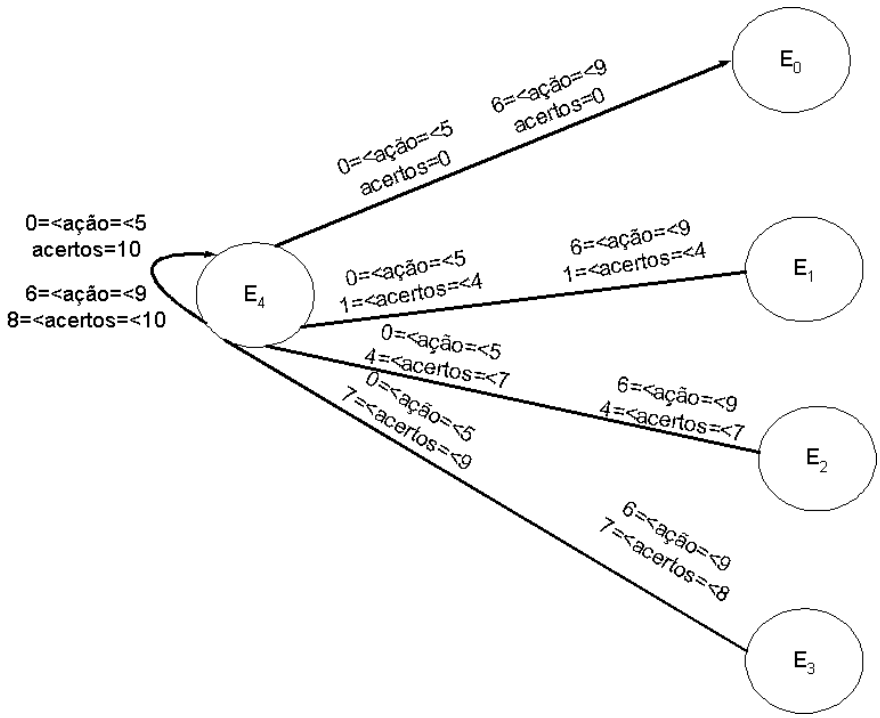

Figura V. 17. – Política Pedagógica P_2 para o Estado E_3.

Os modelos foram submetidos a 500, 1000 e 2500 passos, e de cada conjunto de simulação foi calculada uma média sobre vinte simulações, para obtenção dos dados finais.

V.3. HISTÓRICO E ETAPAS

O protótipo do tutor foi construído em módulos conforme explicado anteriormente (seção V.1). Foi definido nesta etapa qual seria o ambiente que o protótipo de tutor simularia, chegando assim à matriz 5x10 definida na seção V.2. Foram criados ainda os estados e as ações conforme o seção V.2. Após essa etapa, iniciaram-se as simulações.

Em uma primeira etapa da simulação foi criado apenas um modelo M_2 (Anexo 4) determinístico e uma política pedagógica (P_1- Anexo1). Com os resultados obtidos foram criados dois outros modelos (M_1-Anexo 3 e M_3-Anexo 5), também caracterizados por serem modelos determinísticos, ou seja, tinham suas transições para os estados bem definidas. Estes dois modelos, assim como o primeiro, usavam a política pedagógica P_1 (Anexo1).. Os resultados foram satisfatórios, já que dado um modelo, através da política pedagógica P_1 o algoritmo convergiu e como conseqüência resultou no modelo esperado.

Na segunda etapa da simulação foram criados três modelos(M_1-Anexo 6, M_2- Anexo7 e M_3-Anexo 8) que tinham como característica o não-determinismo, ou seja, tinham transições não-determinísticas para os estados. Para estes modelos, foi utilizada a política pedagógica P_2

(Anexo 2). Esta, ao contrário de P_1, era mais restritiva (filtro maior).Os resultados também foram satisfatórios, pois assim como na primeira etapa da simulação, o algoritmo convergiu e como conseqüência também resultou no modelo esperado.

A terceira etapa da simulação consistiu na definição da política de exploração de estados e ações desconhecidos, de modo a coletar nova informação, ou a explotação de estados e ações que já foram aprendidos e que irão gerar altas recompensas, de modo a maximizar seus reforços acumulados. Desta forma a implementação foi feita para que ε-greedy tivesse 20% contemplando a política de exploração e 80% a política de explotação.

Por questões metodológicas, a quarta etapa consistiu na criação de um conjunto de simulações, onde as políticas e os modelos foram submetidos a 500, 1000 e 2500 passos, e foram calculadas as médias sobre vinte simulações cada, para obtenção dos dados finais a serem mostrados (item V.4.).

Na quinta etapa da simulação foram definidos os valores de γ como 0.9 e α como 0.9, estes valores foram definidos empiricamente, visando convergência mais rápida do algoritmo. Neste trabalho não foi contemplado este aspecto, mas esta possível lentidão poderia ser compensada pelo uso de variantes baseadas em aproximações compactas [Tsitsiklis & Bertsekas 1996], estratégias de generalização da experiência [Ribeiro 1998], ou planos de ação obtidos em simulação [Sutton & Barto, 1998].

V.4. RESULTADOS EXPERIMENTAIS

São apresentados os resultados da Política Pedagógica **(P_1)** com o modelo determinístico **M_1 (Ruim)**, seguindo uma simulação com 500 passos, cujo γ é 0.9 e o α é 0.9. Os resultados obtidos na Figura V.18 mostram as médias dos reforços obtidos durante a simulação para o modelo **M_1**. Esses resultados demonstram que o algoritmo Q-Learning seguindo uma política de ações **(μ)** conseguiu convergir para o modelo **M_1 (Ruim)**.

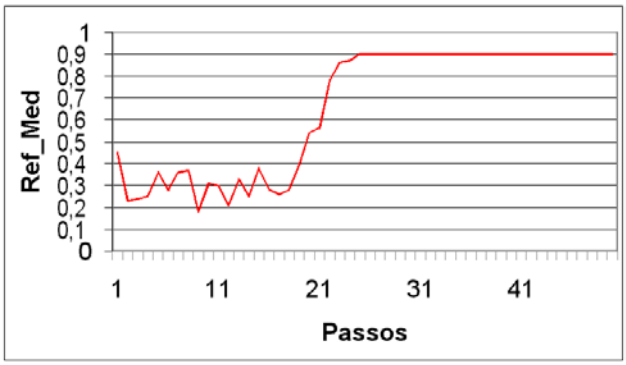

Figura V.18. – Gráfico do Reforço Médio do Modelo Determinístico M_1 para 500 Simulações.

No gráfico da Figura V.19 usa-se a mesma política **(P_1)**, com valores de γ e α igual a 0.9. A Figura V.19. mostra o resultado das médias dos valores de utilidade Q(s,a). Com base nesses valores, o algoritmo consegue definir (através da maximização sobre Q(s,a)) a ação de maior valor de utilidade no estado **s**. Ao longo do tempo, ocorreu convergência para uma política ótima de ações **(μ)** no modelo simulado.

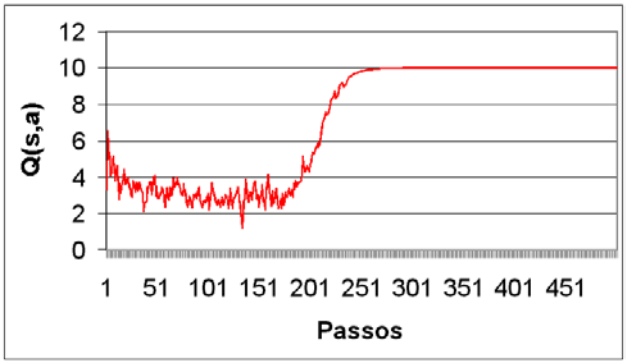

Figura V.19. – Gráfico do Q(s,a) do Modelo Determinístico M_1 para 500 Simulações.

A Figura 20. representa as médias das transições entre os estados cognitivos do modelo simulado. Importante observar que, para obter estes resultados, o protótipo do STI não conhece o modelo: os resultados da Figura V.20 são conseqüências da convergência do Q-Learning para uma política ótima de ações **(μ)**.

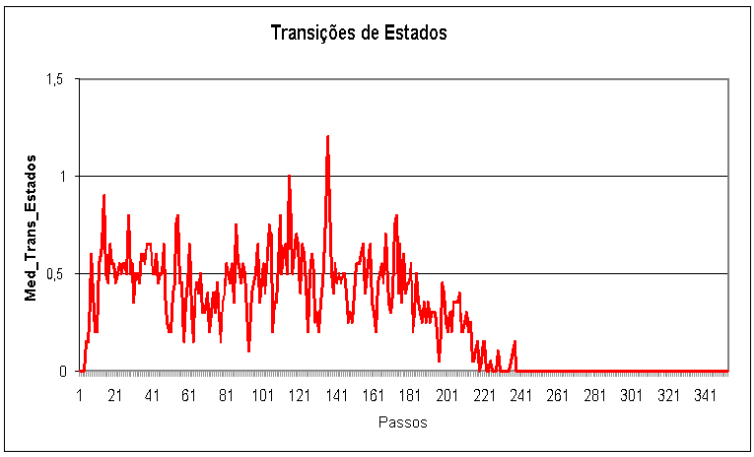

Figura V.20. – Gráficos da Média das Transições dos Estados do Modelo Determinístico (M_1) para 500 Simulações.

No gráfico da Figura V.21 observa-se que o algoritmo Q-Learning consegue, através da política ótima de ações **(μ)**, visitar mais o $\mathbf{E_0}$, modelo $\mathbf{M_1}$ **(Ruim)**, perfazendo um total de 85 % de visitas para este estado (que representa o modelo o qual se esperava obter com as simulações).

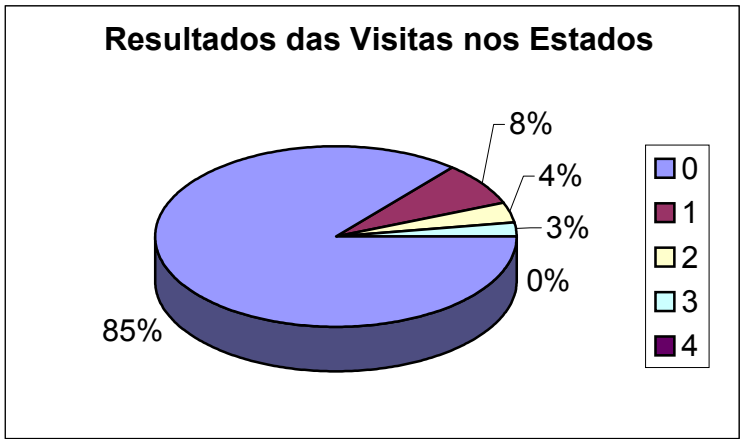

Figura V.21. – Resultados das Visitas nos Estados.

São apresentados os resultados da Política Pedagógica **(P_1)** com o modelo determinístico **M_1 (Ruim)**, seguindo uma simulação com 1000 passos, cujo γ é 0.9 e o α é 0.9. Os resultados obtidos na Figura V.22 mostram as médias dos reforços obtidos durante a simulação para o modelo **M_1**. Esses resultados demonstram que o algoritmo Q-Learning seguindo uma política de ações **(μ)** conseguiu convergir para o modelo **M_1 (Ruim)**

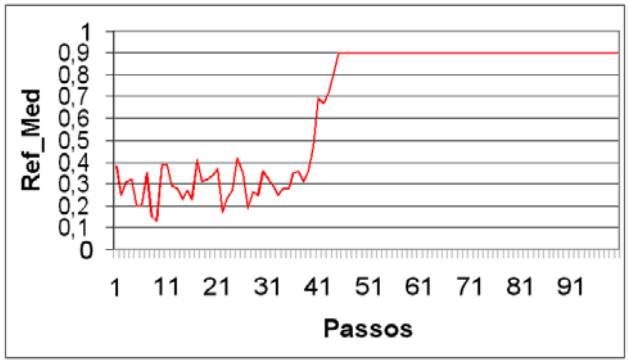

Figura V.22. – Gráfico do Reforço Médio do Modelo Determinístico M_1 para 1000 Simulações.

No gráfico da Figura V.23 usa-se a mesma política **(P_1)**, com valores de γ e α igual a 0.9. A Figura V.23 mostra o resultado das médias dos valores de utilidade Q(s,a). Com base nesses valores, o algoritmo consegue definir (através da maximização sobre Q(s,a)) a ação de maior valor de utilidade no estado **s**. Ao longo do tempo, ocorreu convergência para uma política ótima de ações **(μ)** no modelo simulado.

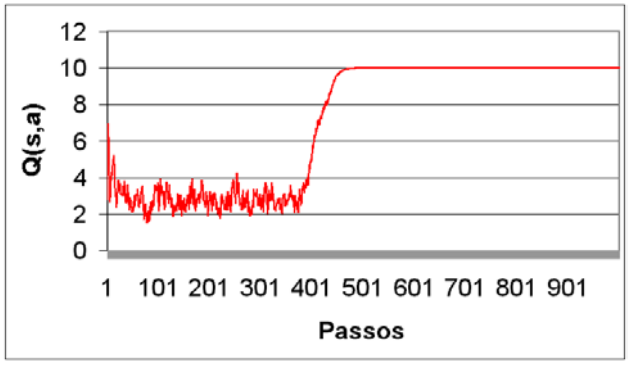

Figura V.23. – Gráfico do Q(s,a) do Modelo Determinístico M_1 para 1000 Simulações.

A Figura V.24 representa as médias das transições entre os estados cognitivos do modelo simulado. Importante observar que, para obter estes resultados, o protótipo do STI não conhece o modelo: os resultados da Figura V.24 são conseqüências da convergência do Q-Learning para uma política ótima de ações **(μ)**.

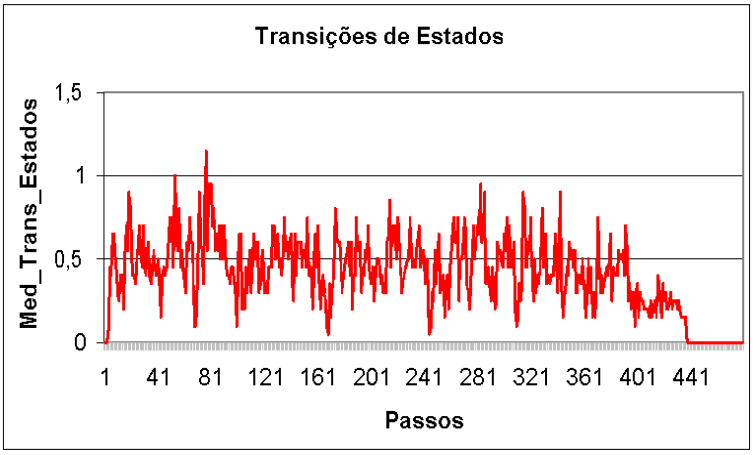

Figura V.24. – Gráficos da Média das Transições dos Estados do Modelo Determinístico (M_1) para 1000 Simulações.

No gráfico da Figura V.25 observa-se que o algoritmo Q-Learning consegue, através da política ótima de ações **(μ)**, visitar mais o $\mathbf{E_0}$, modelo $\mathbf{M_1}$ **(Ruim)**, perfazendo um total de 90 % de visitas para este estado (que representa o modelo o qual se esperava obter com as simulações).

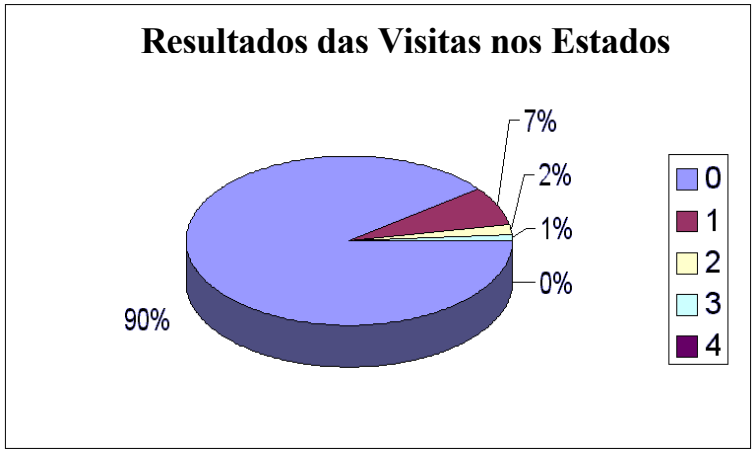

Figura V.26. – Resultados das Visitas nos Estados.

São apresentados os resultados da Política Pedagógica **(P_1)** com o modelo determinístico **M_1 (Ruim)**, seguindo uma simulação com 2500 passos, cujo γ é 0.9 e o α é 0.9. Os resultados obtidos na Figura V.27 mostram as médias dos reforços obtidos durante a simulação para o modelo **M_1**.Esses resultados demonstram que o algoritmo Q-Learning seguindo uma política de ações **(μ)** conseguiu convergir para o modelo **M_1 (Ruim)**.

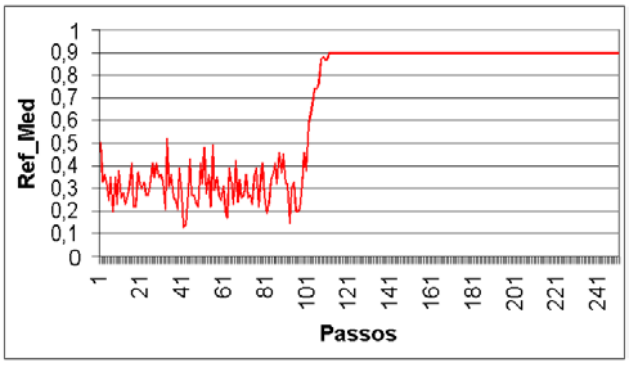

Figura V.27 – Gráfico do Reforço Médio do Modelo Determinístico M_1 para 2500 Simulações.

No gráfico da Figura V.28 usa-se a mesma política **(P_1)**, com valores de γ e α igual a 0.9. A Figura V.28 mostra o resultado das médias dos valores de utilidade Q(s,a). Com base nesses valores, o algoritmo consegue definir (através da maximização sobre Q(s,a)) a ação de maior valor de utilidade no estado **s**. Ao longo do tempo, ocorreu convergência para uma política ótima de ações **(μ)** no modelo simulado.

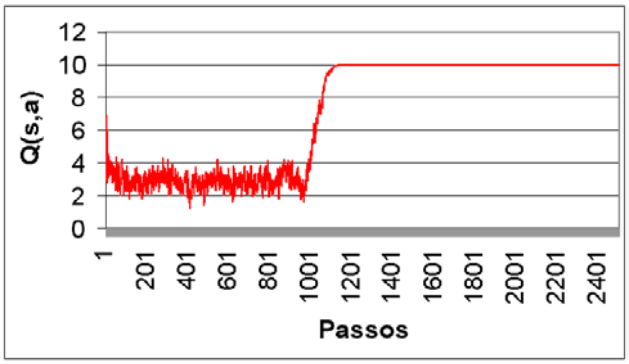

Figura V.28 – Gráfico do Q(s,a) do Modelo Determinístico M_1 para 2500 Simulações.

A Figura V.29 representa as médias das transições entre os estados cognitivos do modelo simulado. Importante observar que, para obter estes resultados, o protótipo do STI não conhece o modelo: os resultados da Figura V.29 são conseqüências da convergência do Q-Learning para uma política ótima de ações **(μ)**.

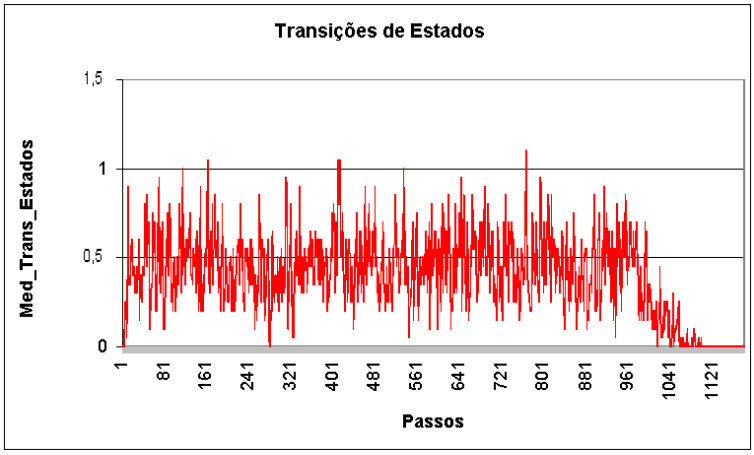

Figura V.29 – Gráficos da Média das Transições dos Estados do Modelo Determinístico (M_2) para 2500 Simulações.

No Gráfico da Figura V.30 observa-se que o algoritmo Q-Learning consegue, através da política ótima de ações **(μ)**, visitar mais o $\mathbf{E_0}$, modelo $\mathbf{M_1}$ **(Ruim)**, perfazendo um total de 88 % de visitas para este estado (que representa o modelo o qual se esperava obter com as simulações).

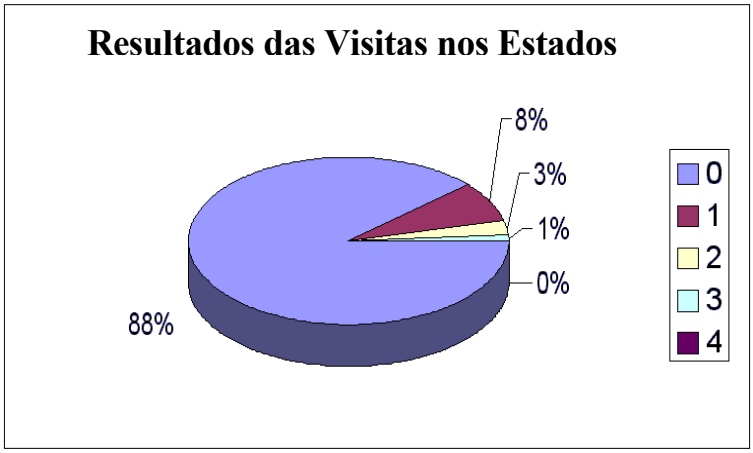

Figura V.30 – Resultados das Visitas nos Estados.

São apresentados os resultados da Política Pedagógica **(P_1)** com o modelo determinístico **M_2 (Bom)**, seguindo uma simulação com 500 passos, cujo γ é 0.9 e o α é 0.9. Os Resultados obtidos na Figura V.31 mostram as médias dos reforços obtidos durante a simulação para o modelo **M_2**. Esses resultados demonstram que o algoritmo Q-Learning seguindo uma política de ações **(μ)** conseguiu convergir para o modelo e **M_2 (Bom)**.

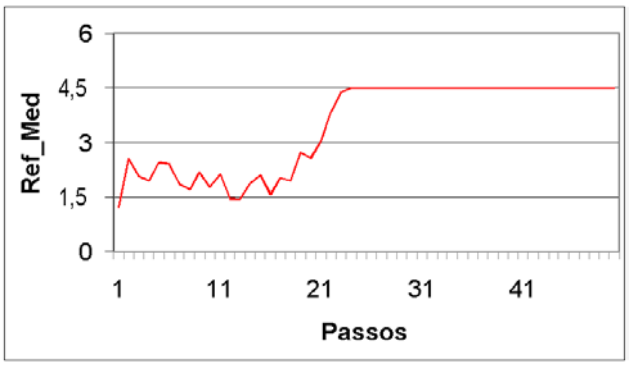

Figura V.31 – Gráfico do Reforço Médio do Modelo Determinístico M_2 para 500 Simulações.

No gráfico da Figura V.32 usa-se a mesma política **(P_1)**, com valores de γ e α igual a 0.9. A Figura V.32 mostra o resultado das médias dos valores de utilidade Q(s,a). Com base nesses valores, o algoritmo consegue definir (através da maximização sobre Q(s,a)) a ação de maior valor de utilidade no estado **s**. Ao longo do tempo, ocorreu convergência para uma política ótima de ações **(μ)** no modelo simulado.

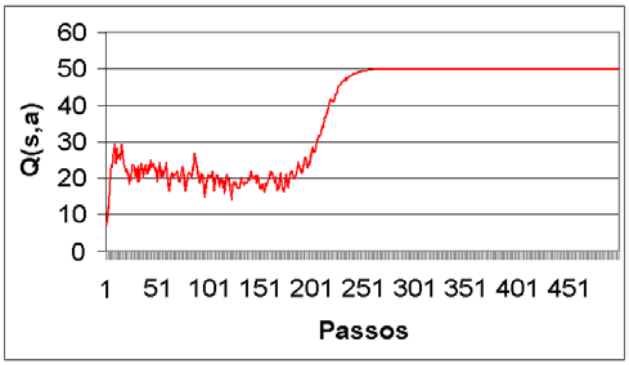

Figura V.32 – Gráfico Q(s,a) do Modelo Determinístico M_2 para 500 Simulações.

A Figura V.33 representa as médias das transições entre os estados cognitivos do modelo simulado. Importante observar que, para obter estes resultados, o protótipo do STI não conhece o modelo: os resultados da Figura V.33 são conseqüências da convergência do Q-Learning para uma política ótima de ações **(μ)**.

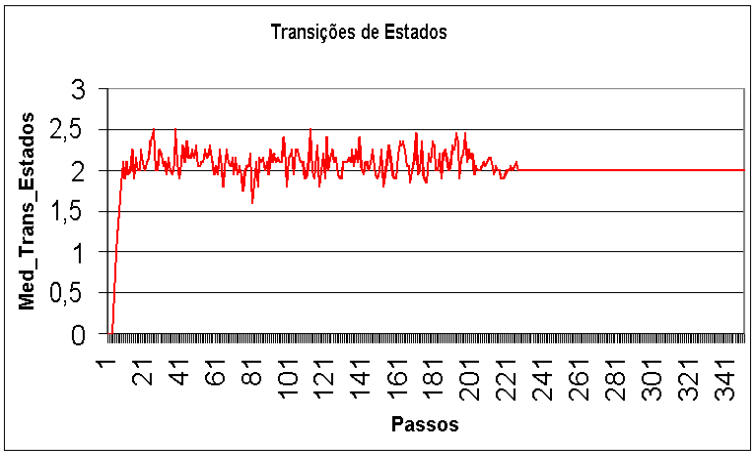

Figura V.33– Gráficos da Média das Transições dos Estados do Modelo Determinístico (M_2) para 500 Simulações.

No gráfico da Figura V.34observa-se que o algoritmo Q-Learning consegue, através da política ótima de ações **(μ)**, visitar mais o $\mathbf{E_2}$, modelo $\mathbf{M_2}$ **(Bom)**, , perfazendo um total de 86 % de visitas para este estado (que representa o modelo o qual se esperava obter com as simulações).

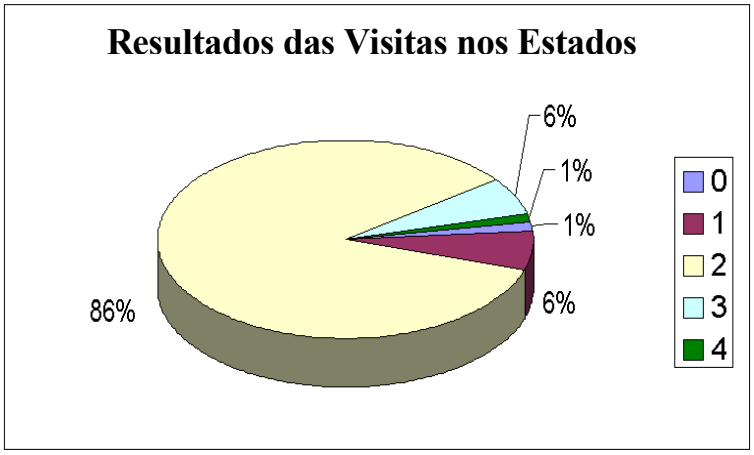

Figura V.34– Resultados das Visitas nos Estados.

São apresentados os resultados da Política Pedagógica **(P_1)** com o modelo determinístico **M_2 (Bom)**, seguindo uma simulação com 1000 passos, cujo γ é 0.9 e o α é 0.9. Os resultados obtidos na Figura V.35 mostram as médias dos reforços obtidos durante a simulação para o modelo **M_2**. Esses resultados demonstram que o algoritmo Q-Learning seguindo uma política de ações **(μ)** conseguiu convergir para o modelo **M_2 (Bom)**.

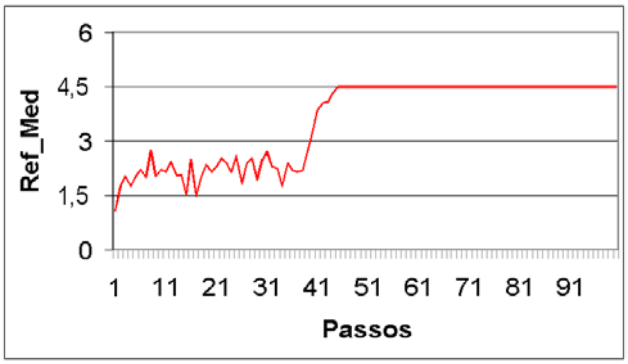

Figura V.35 – Gráfico do Reforço Médio do Modelo Determinístico M_2 para 1000 Simulações.

No gráfico da Figura V.36 usa-se a mesma política **(P_1)**, com valores de γ e α igual a 0.9. A Figura V.37 mostra o resultado das médias dos valores de utilidade Q(s,a). Com base nesses valores, o algoritmo consegue definir (através da maximização sobre Q(s,a)) a ação de maior valor de utilidade no estado **s**. Ao longo do tempo, ocorreu convergência para uma política ótima de ações **(μ)** no modelo simulado.

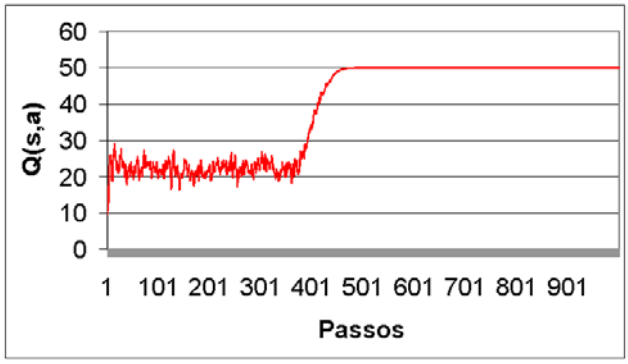

Figura V.37 – Gráfico do Q(s,a) do Modelo Determinístico M_2 para 1000 Simulações.

A Figura V.38 representa as médias das transições entre os estados cognitivos do modelo simulado. Importante observar que, para obter estes resultados, o protótipo do STI não conhece o modelo: os resultados da Figura V.38 são conseqüências da convergência do Q-Learning para uma política ótima de ações **(μ)**.

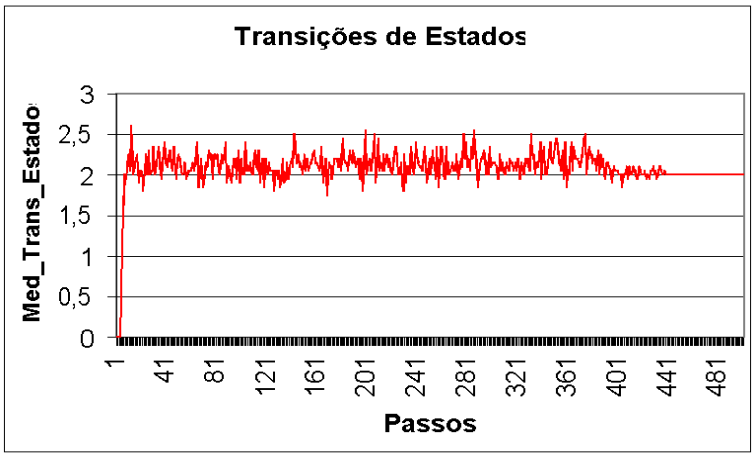

Figura V.38 – Gráficos da Média das Transições dos Estados do Modelo Determinístico (M_2) para 1000 Simulações.

No gráfico da Figura V.39observa-se que o algoritmo Q-Learning consegue, através da política ótima de ações **(μ)**, visitar mais o $\mathbf{E_2}$, modelo $\mathbf{M_2}$ **(Bom)**, perfazendo um total de 87 % de visitas para este estado (que representa o modelo o qual se esperava obter com as simulações).

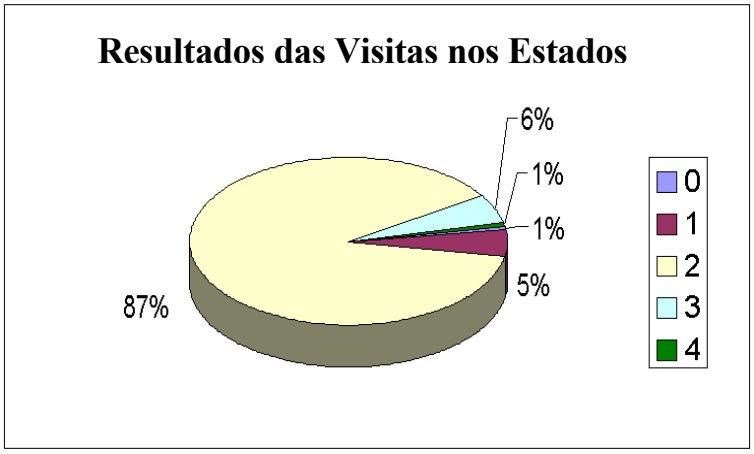

Figura V.39 – Resultados das Visitas nos Estados.

São apresentados os resultados da Política Pedagógica **(P_1)** com o modelo determinístico **M_2 (Bom)**, seguindo uma simulação com 2500 passos, cujo γ é 0.9 e o α é 0.9. Os resultados obtidos na Figura V.40 mostram as médias dos reforços obtidos durante a simulação para o modelo **M_2**. Esses resultados demonstram que o algoritmo Q-Learning seguindo uma política de ações **(μ)** conseguiu convergir para o modelo **M_2 (Bom)**.

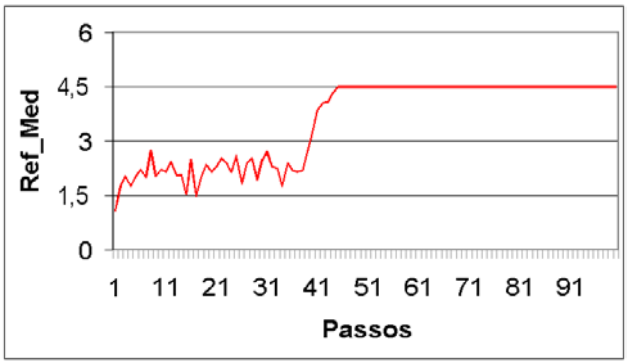

Figura V.40 – Gráfico do Reforço Médio do Modelo Determinístico M_2 para 2500 Simulações.

No gráfico da Figura V.41 usa-se a mesma política **(P_1)**, com valores de γ e α igual a 0.9. A Figura V.41 mostra o resultado das médias dos valores de utilidade Q(s,a). Com base nesses valores, o algoritmo consegue definir (através da maximização sobre Q(s,a)) a ação de maior valor de utilidade no estado **s**. Ao longo do tempo, ocorreu convergência para uma política ótima de ações **(μ)** no modelo simulado.

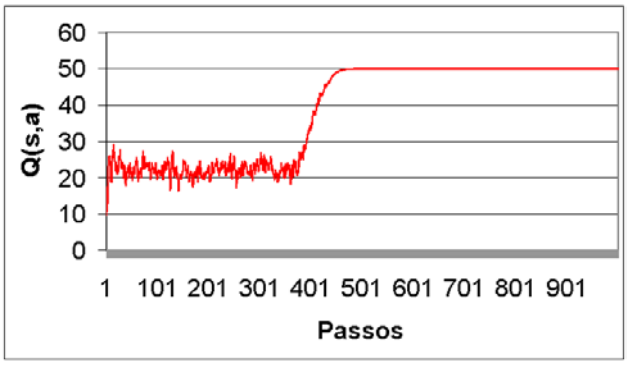

Figura V.41 – Gráfico do Q(s,a) do Modelo Determinístico M_2 para 2500 Simulações.

A Figura V.42 representa as médias das transições entre os estados cognitivos do modelo simulado. Importante observar que, para obter estes resultados, o protótipo do STI não conhece o modelo: os resultados da Figura V.42 são conseqüências da convergência do Q-Learning para uma política ótima de ações **(μ)**.

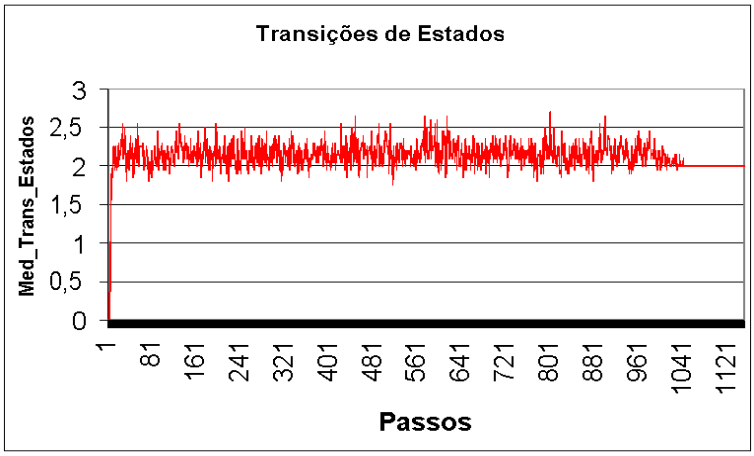

Figura V.42 – Gráficos da Média das Transições dos Estados do Modelo Determinístico (M_2) para 2500 Simulações.

No gráfico da Figura V.43 observa-se que o algoritmo Q-Learning consegue, através da política ótima de ações **(μ)**, visitar mais o $\mathbf{E_2}$, modelo $\mathbf{M_2}$ **(Bom)**, perfazendo um total de 87 % de visitas para este estado (que representa o modelo o qual se esperava obter com as simulações).

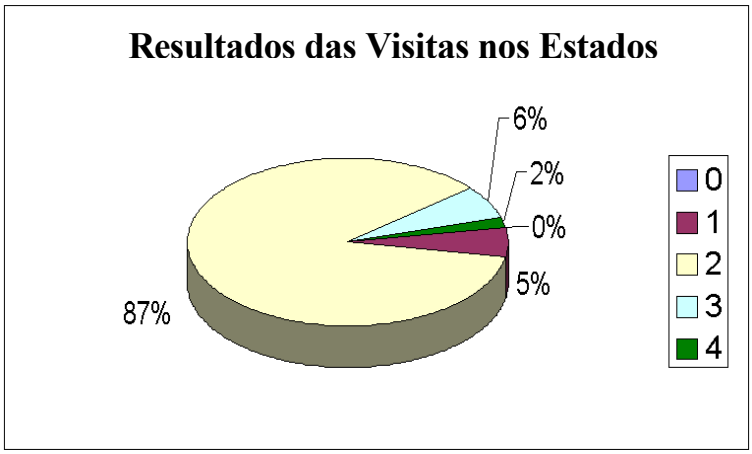

Figura V.43 – Resultados das Visitas nos Estados.

São apresentados os resultados da Política Pedagógica **(P_1)** com o modelo determinístico **M_3 (Excelente)**, seguindo uma simulação com 500 passos, cujo γ é 0.9 e o α é 0.9. Os resultados obtidos na Figura V.44 mostram as médias dos reforços obtidos durante a simulação para o modelo **M_3**. Esses resultados demonstram que o algoritmo Q-Learning seguindo uma política de ações **(μ)** conseguiu convergir para o modelo **M_3 (Excelente)**.

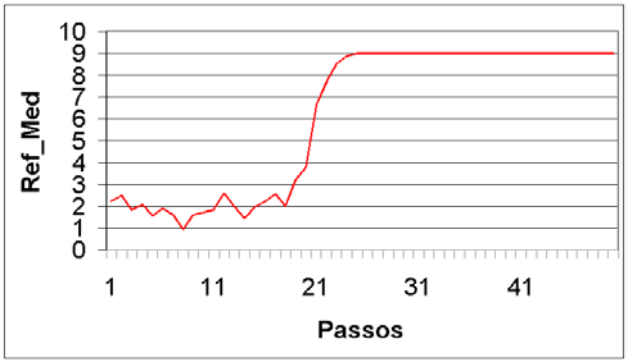

Figura V.44 – Gráfico do Reforço Médio do Modelo Determinístico M_3 para 500 Simulações.

No gráfico da Figura V.45 usa-se a mesma política **(P_1)**, com valores de γ e α igual a 0.9. A Figura V.45 mostra o resultado das médias dos valores de utilidade Q(s,a). Com base nesses valores, o algoritmo consegue definir (através da maximização sobre Q(s,a)) a ação de maior valor de utilidade no estado **s**. Ao longo do tempo, ocorreu convergência para uma política ótima de ações **(μ)** no modelo simulado.

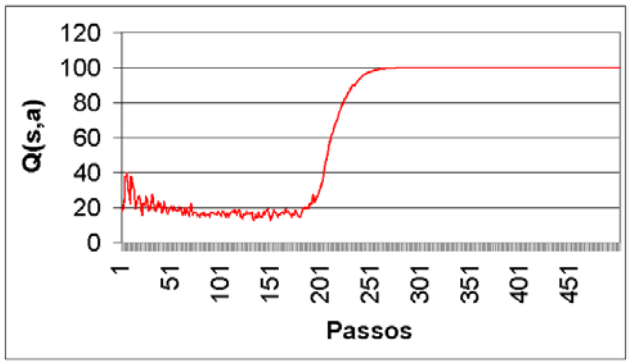

Figura V.45 – Gráfico Q(s,a) do Modelo Determinístico M_3 para 500 Simulações.

A Figura V.46 representa as médias das transições entre os estados cognitivos do modelo simulado. Importante observar que, para obter estes resultados, o protótipo do STI não conhece o modelo: os resultados da Figura V.46 são conseqüências da convergência do Q-Learning para uma política ótima de ações **(μ)**.

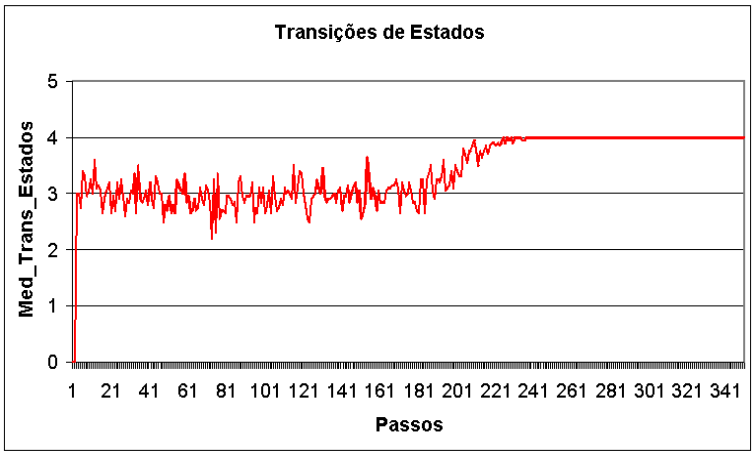

Figura V.46– Gráficos da Média das Transições dos Estados do Modelo Determinístico (M_3) para 500.

No gráfico da Figura V.47observa-se que o algoritmo Q-Learning consegue, através da política ótima de ações **(μ)**, visitar mais o **E_4**, modelo **M_3 (Excelente)**, perfazendo um total de 78 % de visitas para este estado (que representa o modelo o qual se esperava obter com as simulações).

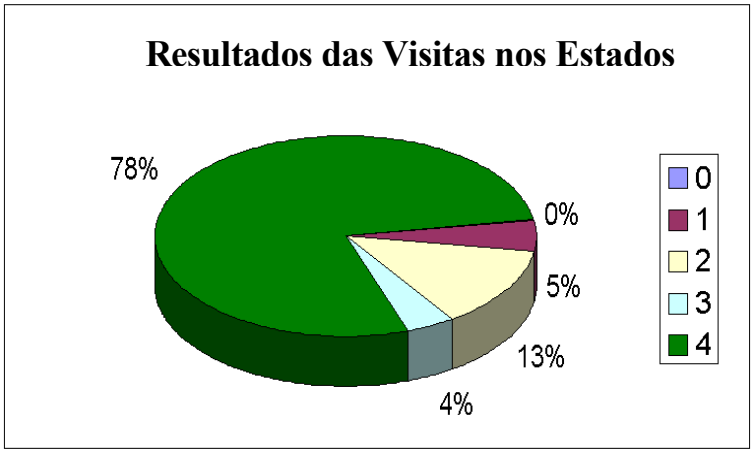

Figura V.47 – Resultados das Visitas nos Estados.

São apresentados os resultados da Política Pedagógica **(P_1)** com o modelo determinístico **M_3 (Excelente)**, seguindo uma simulação com 1000 passos, cujo γ é 0.9 e o α é 0.9. Os resultados obtidos na Figura V.48 mostram as médias dos reforços obtidos durante a simulação para o modelo **M_3**.Esses resultados demonstram que o algoritmo Q-Learning seguindo uma política de ações **(μ)** conseguiu convergir para o modelo **M_3 (Excelente)**.

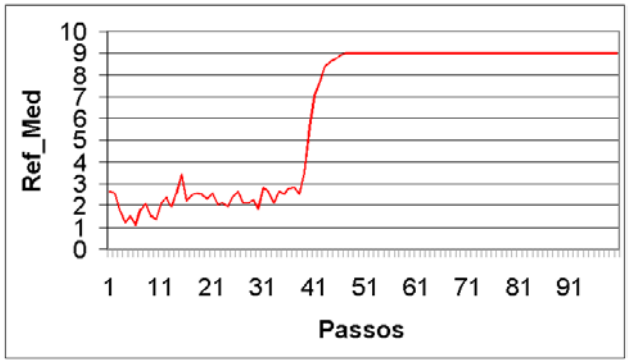

Figura V.48 – Gráfico do Reforço Médio do Modelo Determinístico M_3 para 1000 Simulações.

No gráfico da Figura V.49 usa-se a mesma política **(P_1)**, com valores de γ e α igual a 0.9. A Figura V.49 mostra o resultado das médias dos valores de utilidade Q(s,a). Com base nesses valores, o algoritmo consegue definir (através da maximização sobre Q(s,a)) a ação de maior valor de utilidade no estado **s**. Ao longo do tempo, ocorreu convergência para uma política ótima de ações **(μ)** no modelo simulado.

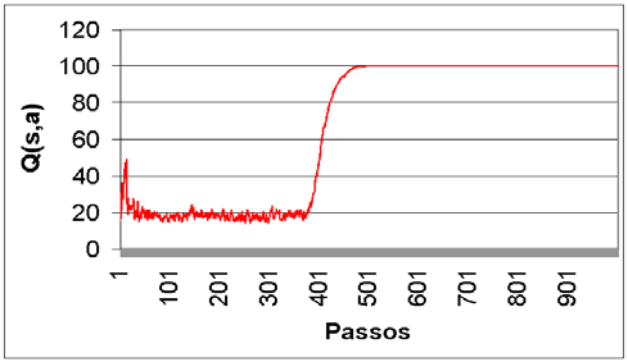

Figura V.49 – Gráfico do Q(s,a) do Modelo Determinístico M_3 para 1000 Simulações.

A Figura V.50 representa as médias das transições entre os estados cognitivos do modelo simulado. Importante observar que, para obter estes resultados, o protótipo do STI não conhece o modelo: os resultados da Figura V.50 são conseqüências da convergência do Q-Learning para uma política ótima de ações **(μ)**.

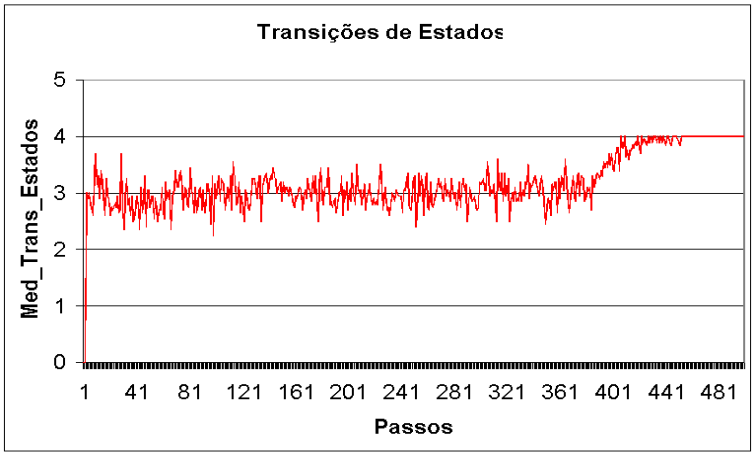

Figura V.50– Gráficos da Média das Transições dos Estados do Modelo Determinístico (M_3) para 1000 Simulações.

No gráfico da Figura V.51. observa-se que o algoritmo Q-Learning consegue, através da política ótima de ações **(μ)**, visitar mais o $\mathbf{E_4}$, modelo $\mathbf{M_3}$**(Excelente)**, perfazendo um total de 78 % de visitas para este estado (que representa o modelo o qual se esperava obter com as simulações).

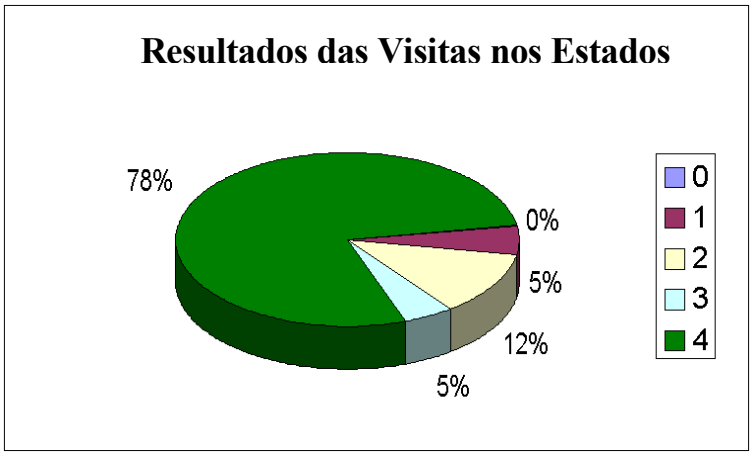

Figura V.51 – Resultados das Visitas nos Estados.

São apresentados os resultados da Política Pedagógica **(P_1)** com o modelo determinístico **M_3 (Excelente)**, seguindo uma simulação com 2500 passos, cujo γ é 0.9 e o α é 0.9. Os Resultados obtidos na Figura V.52 mostram as médias dos reforços obtidos durante a simulação para o modelo **M_3**. Esses resultados demonstram que o algoritmo Q-Learning seguindo uma política de ações **(μ)** conseguiu convergir para o modelo **M_3 (Excelente)**.

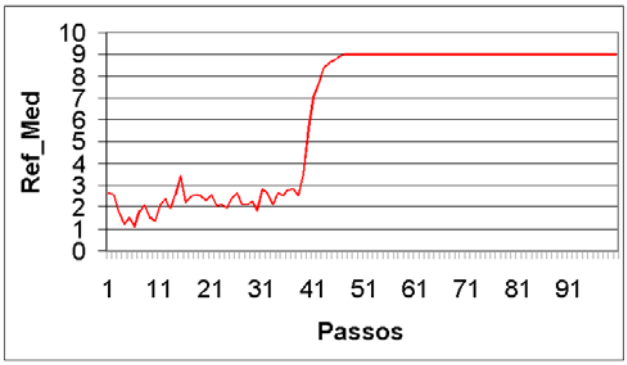

Figura V.52 – Gráfico do Reforço Médio do Modelo Determinístico M_3 para 2500 Simulações.

No gráfico da Figura V.53 usa-se a mesma política **(P_1)**, com valores de γ e α igual a 0.9. A Figura V.53 mostra o resultado das médias dos valores de utilidade Q(s,a). Com base nesses valores, o algoritmo consegue definir (através da maximização sobre Q(s,a)) a ação de maior valor de utilidade no estado **s**. Ao longo do tempo, ocorreu convergência para uma política ótima de ações **(μ)** no modelo simulado.

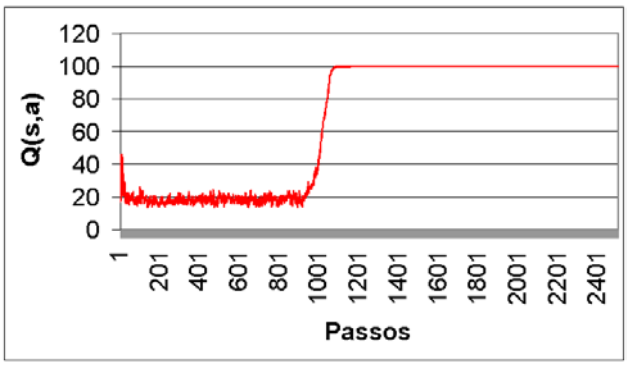

Figura V.53 – Gráfico do Q(s,a) do Modelo Determinístico M_3 para 2500 Simulações.

A Figura V.54 representa as médias das transições entre os estados cognitivos do modelo simulado. Importante observar que, para obter estes resultados, o protótipo do STI não conhece o modelo: os resultados da Figura V.54 são conseqüências da convergência do Q-Learning para uma política ótima de ações **(μ)**.

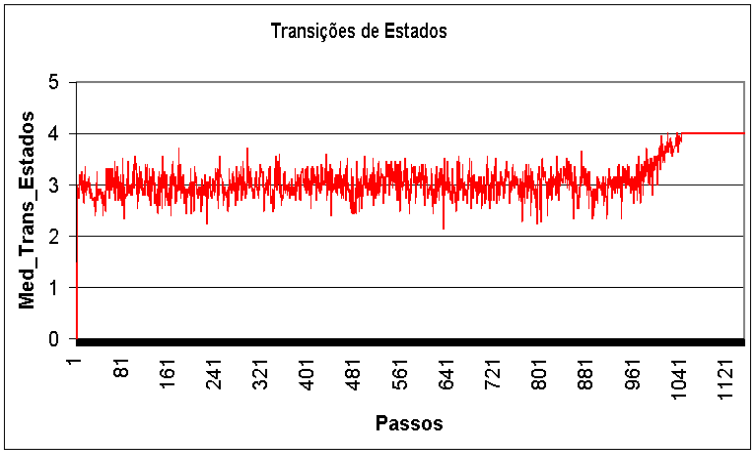

Figura V.54 – Gráficos da Média das Transições dos Estados do Modelo Determinístico (M_3) para 2500 Simulações.

No gráfico da Figura V.55. observa-se que o algoritmo Q-Learning consegue, através da política ótima de ações **(μ)**, visitar mais o **E_4**, modelo **M_3 (Excelente)**, perfazendo um total de 80 % de visitas para este estado (que representa o modelo o qual se esperava obter com as simulações).

Figura V.55 – Resultados das Visitas nos Estados.

São apresentados os resultados da Política Pedagógica **(P_2)** com o modelo Não determinístico **M_1 (Ruim)**, seguindo uma simulação com 500 passos, cujo γ é 0.9 e o α é 0.9. Os Resultados obtidos na Figura V.56mostram as médias dos reforços obtidos durante a simulação para o modelo **M_1**. Esses resultados demonstram que o algoritmo Q-Learning seguindo uma política de ações **(μ)** conseguiu convergir para o modelo **M_1 (Ruim)**.

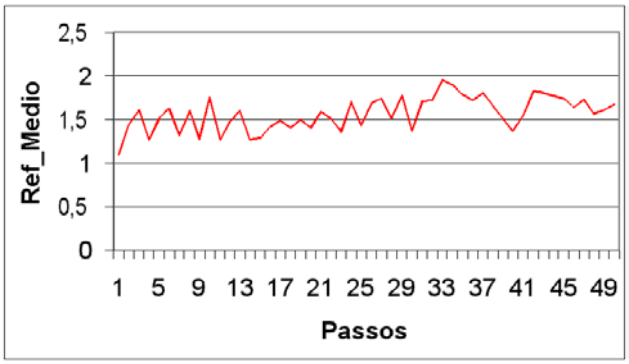

Figura V.56– Gráfico do Reforço Médio do Modelo Não Determinístico M_1 para 500 Simulações.

No gráfico da Figura V.57 usa-se a mesma política **(P_2)**, com valores de γ e α igual a 0.9. A Figura V.57 mostra o resultado das médias dos valores de utilidade Q(s,a). Com base nesses valores, o algoritmo consegue definir (através da maximização sobre Q(s,a)) a ação de maior valor de utilidade no estado **s**. Ao longo do tempo, ocorreu convergência para uma política ótima de ações **(μ)** no modelo simulado.

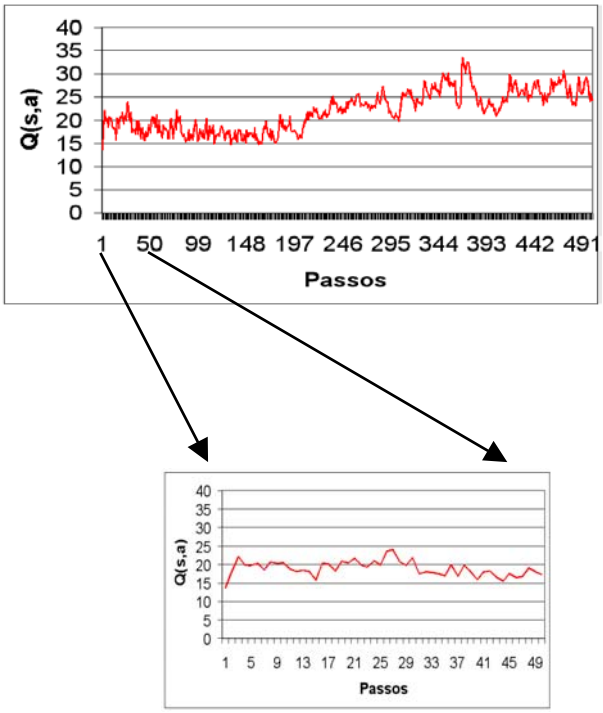

Figura V.57 – Gráfico do Q(s,a) do Modelo Não Determinístico M_1 para 500 Simulações.

A Figura V.58 representa as médias das transições entre os estados cognitivos do modelo simulado. Importante observar que, para obter estes resultados, o protótipo do STI não conhece o modelo: os resultados da Figura V.58 são conseqüências da convergência do Q-Learning para uma política ótima de ações **(μ)**.

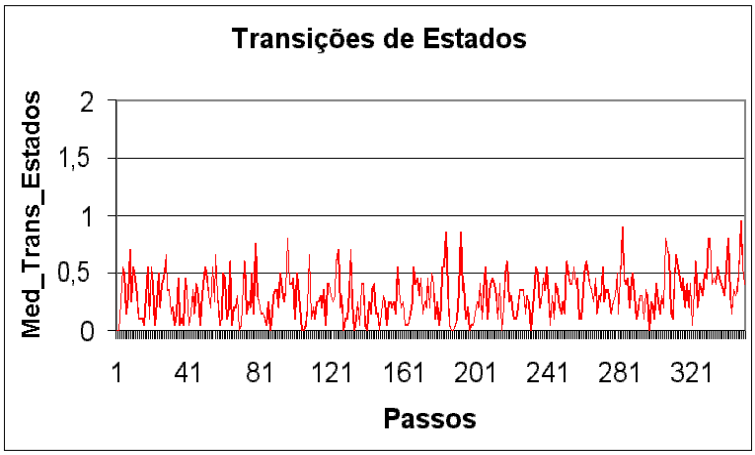

Figura V.40 – Gráficos da Média das Transições dos Estados do Modelo Não Determinístico (M_1) para 500 Simulações.

No gráfico da Figura V.59. observa-se que o algoritmo Q-Learning consegue, através da política ótima de ações **(μ)**, visitar mais o $\mathbf{E_0}$, modelo $\mathbf{M_1}$ **(Ruim)**, perfazendo um total de 86 % de visitas para este estado (que representa o modelo o qual se esperava obter com as simulações).

Figura V.59– Resultados das Visitas nos Estados.

São apresentados os resultados da Política Pedagógica **(P_2)** com o modelo Não Determinístico **M_1 (Ruim)**, seguindo uma simulação com 1000 passos, cujo γ é 0.9 e o α é 0.9. Os resultados obtidos na Figura V.60mostram as médias dos reforços obtidos durante a simulação para o modelo **M_1**. Esses resultados demonstram que o algoritmo Q-Learning seguindo uma política de ações **(μ)** conseguiu convergir para o modelo **M_1 (Ruim)**.

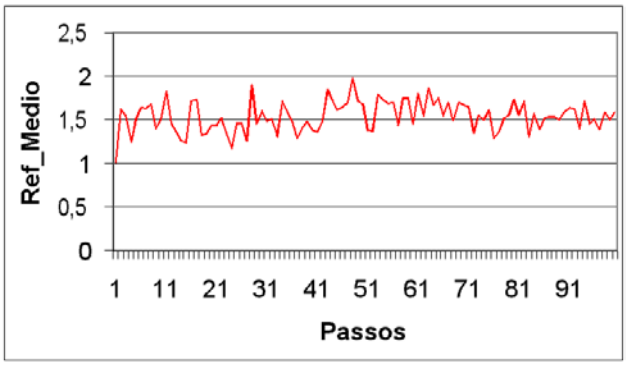

Figura V.60– Gráfico do Reforço Médio do Modelo Não Determinístico M_1 para 1000 Simulações.

No gráfico da Figura V.61 usa-se a mesma política **(P_2)**, com valores de γe α igual a 0.9. A Figura V.61 mostra o resultado das médias dos valores de utilidade Q(s,a). Com base nesses valores, o algoritmo consegue definir (através da maximização sobre Q(s,a)) a ação de maior valor de utilidade no estado **s**. Ao longo do tempo, ocorreu convergência para uma política ótima de ações **(μ)** no modelo simulado.

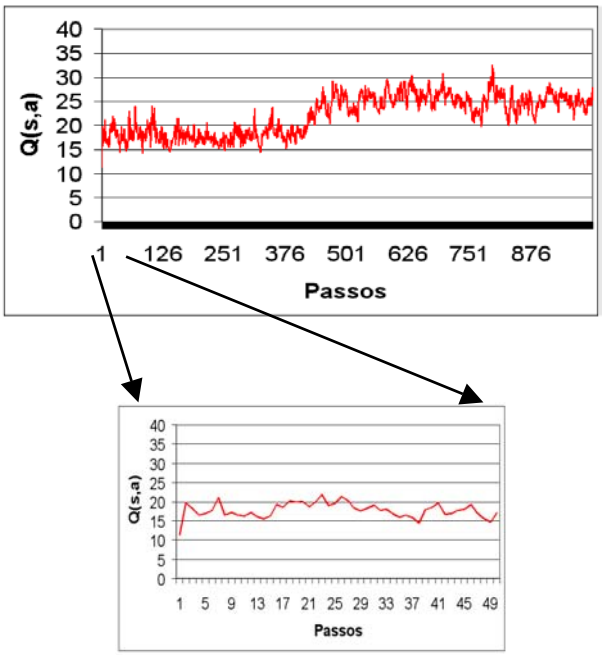

Figura V.61 – Gráfico do Q(s,a) do Modelo Não Determinístico M_1 para 1000 Simulações.

A Figura V.62 representa as médias das transições entre os estados cognitivos do modelo simulado. Importante observar que, para obter estes resultados, o protótipo do STI não conhece o modelo: os resultados da Figura V.62 são conseqüências da convergência do Q-Learning para uma política ótima de ações **(μ)**.

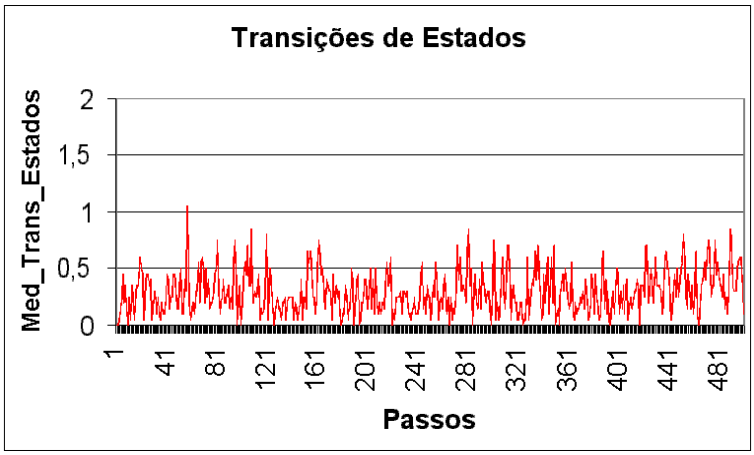

Figura V.62 – Gráficos da Média das Transições dos Estados do Modelo Não Determinístico (M_1) para 1000 Simulações.

No gráfico da Figura V.63. observa-se que o algoritmo Q-Learning consegue, através da política ótima de ações **(μ)**, visitar mais o $\mathbf{E_0}$, modelo $\mathbf{M_1}$ **(Ruim)**, perfazendo um total de 86 % de visitas para este estado (que representa o modelo o qual se esperava obter com as simulações).

Figura V.64– Resultados das Visitas nos Estados.

São apresentados os resultados da Política Pedagógica **(P_2)** com o modelo Não determinístico **M_1 (Ruim)**, seguindo uma simulação com 2500 passos, cujo γ é 0.9 e o α é 0.9. Os Resultados obtidos na Figura V.65 mostram as médias dos reforços obtidos durante a simulação para o modelo **M_1**. Esses resultados demonstram que o algoritmo Q-Learning seguindo uma política de ações **(μ)** conseguiu convergir para o modelo **M_1 (Ruim)**.

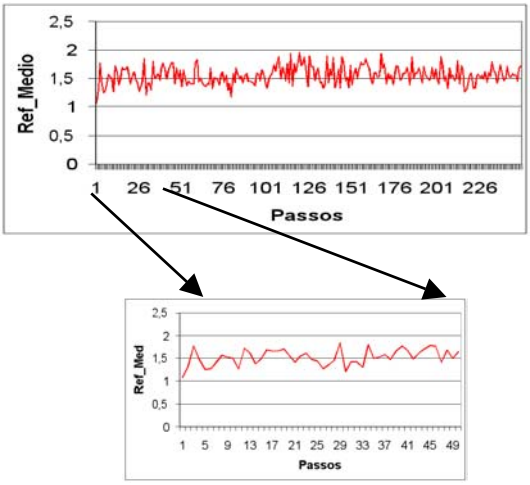

Figura V.65 – Gráfico do Reforço Médio do Modelo Não Determinístico M_1 para 2500 Simulações.

No gráfico da Figura V.66 usa-se a mesma política **(P_2)**, com valores de γe α igual a 0.9. A Figura V.66 mostra o resultado das médias dos valores de utilidade Q(s,a). Com base nesses valores, o algoritmo consegue definir (através da maximização sobre Q(s,a)) a ação de maior valor de utilidade no estado **s**. Ao longo do tempo, ocorreu convergência para uma política ótima de ações **(μ)** no modelo simulado.

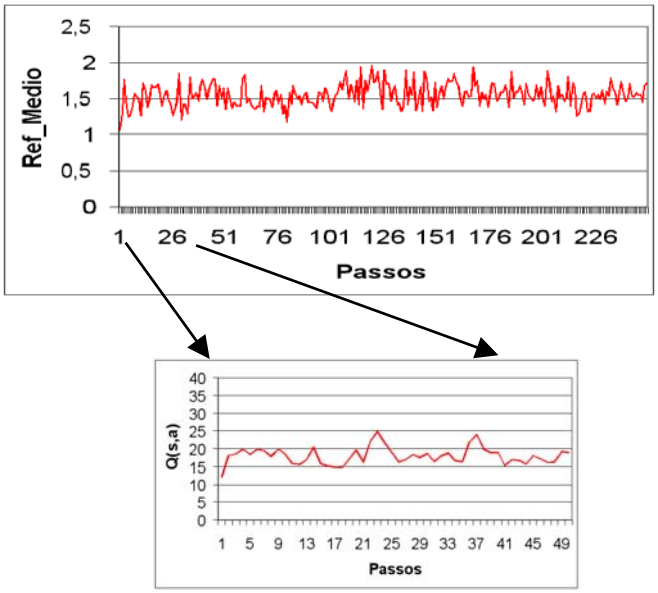

Figura V.66 – Gráfico do Q(s,a) do Modelo Não Determinístico M_1 para 2500 Simulações

A Figura V.67 representa as médias das transições entre os estados cognitivos do modelo simulado. Importante observar que, para obter estes resultados, o protótipo do STI não conhece o modelo: os resultados da Figura V.67 são conseqüências da convergência do Q-Learning para uma política ótima de ações **(μ)**.

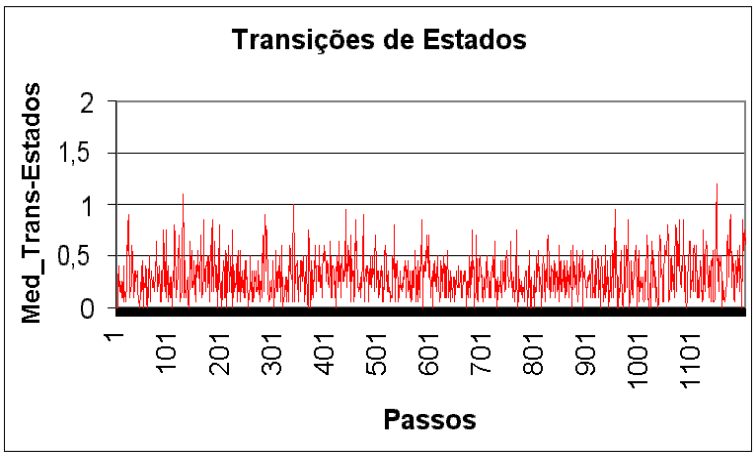

Figura V.67– Gráficos da Média das Transições dos Estados do Modelo Não Determinístico (M_1) para 2500 Simulações.

No gráfico da Figura V.68. observa-se que o algoritmo Q-Learning consegue, através da política ótima de ações **(μ)**, visitar mais o $\mathbf{E_0}$, modelo $\mathbf{M_1}$ **(Ruim)**, perfazendo um total de 84 % de visitas para este estado (que representa o modelo o qual se esperava obter com as simulações).

Figura V.68 – Resultados das Visitas nos Estados.

São apresentados os resultados da Política Pedagógica (P2) com o modelo Não determinístico **M2 (Bom)**, seguindo uma simulação com 500 passos, cujo γ é 0.9 e o α é 0.9. Os resultados obtidos na Figura V.69 mostram as médias dos reforços obtidos durante a simulação para o modelo **M2**. Esses resultados demonstram que o algoritmo Q-Learning seguindo uma política de ações **(μ)** conseguiu convergir para o modelo **M2 (Bom)**.

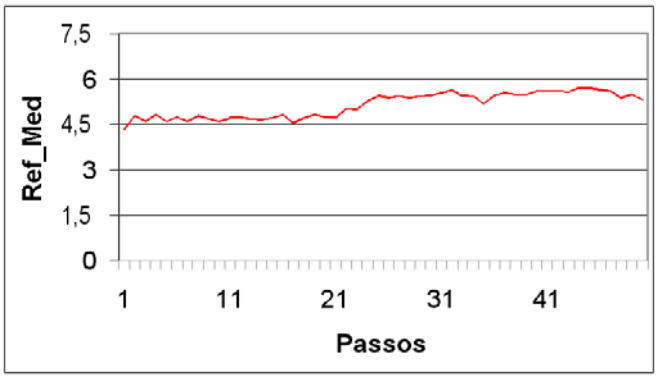

Figura V.69 – Gráfico do Reforço Médio do Modelo Não Determinístico M_2 para 500 Simulações.

No gráfico da Figura V.70 usa-se a mesma política**(P_2)**, com valores de γ e α igual a 0.9. A Figura V.70 mostra o resultado das médias dos valores de utilidade Q(s,a). Com base nesses valores, o algoritmo consegue definir (através da maximização sobre Q(s,a)) a ação de maior valor de utilidade no estado **s**. Ao longo do tempo, ocorreu convergência para uma política ótima de ações **(μ)** no modelo simulado.

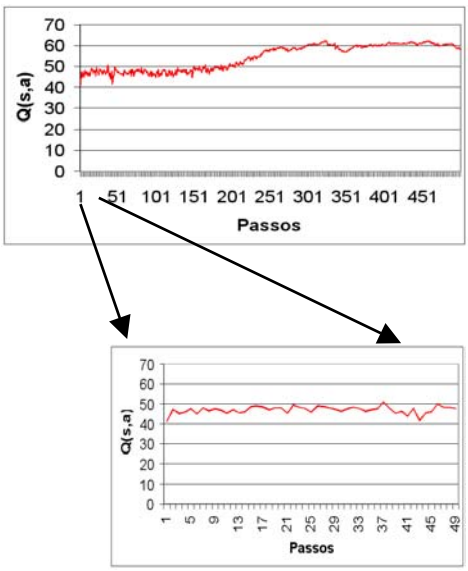

Figura V.70– Gráfico do Q(s,a)do Modelo Não Determinístico M_2 para 500 Simulações.

A Figura V.71 representa as médias das transições entre os estados cognitivos do modelo simulado. Importante observar que, para obter estes resultados, o protótipo do STI não conhece o modelo: os resultados da Figura V.71 são conseqüências da convergência do Q-Learning para uma política ótima de ações **(μ)**.

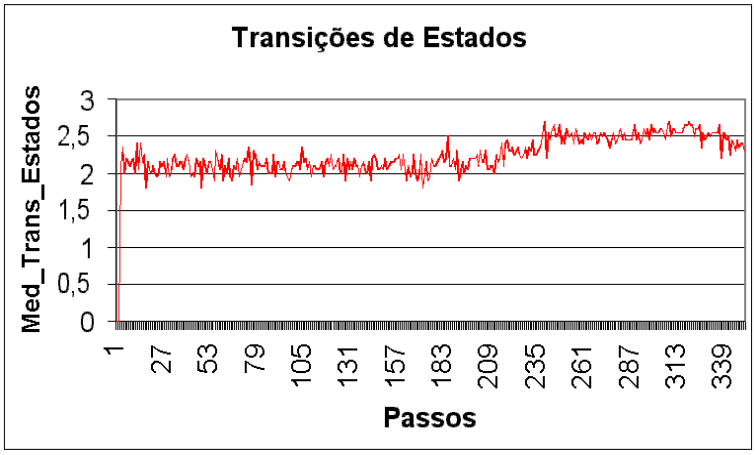

Figura V.71 – Gráficos da Média das Transições dos Estados do Modelo Não Determinístico (M_2) para 500 Simulações.

No gráfico da Figura V.72. observa-se que o algoritmo Q-Learning consegue, através da política ótima de ações **(μ)**, visitar mais o $\mathbf{E_2}$, modelo $\mathbf{M_2}$**(Bom)**, perfazendo um total de 60 % de visitas para este estado (que representa o modelo o qual se esperava obter com as simulações).

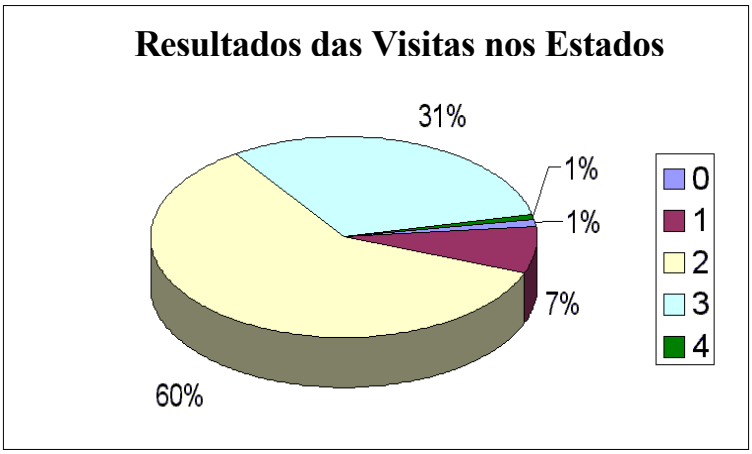

Figura V.72– Resultados das Visitas nos Estados.

São apresentados os resultados da Política Pedagógica **(P_2)** com o modelo Não determinístico **M_2(Bom)**, seguindo uma simulação com 1000 passos, cujo γ é 0.9 e o α é 0.9. Os resultados obtidos na Figura V.73 mostram as médias dos reforços obtidos durante a simulação para o modelo **M_2**. Esses resultados demonstram que o algoritmo Q-Learning seguindo uma política de ações **(μ)** conseguiu convergir para o modelo **M_2(Bom**.

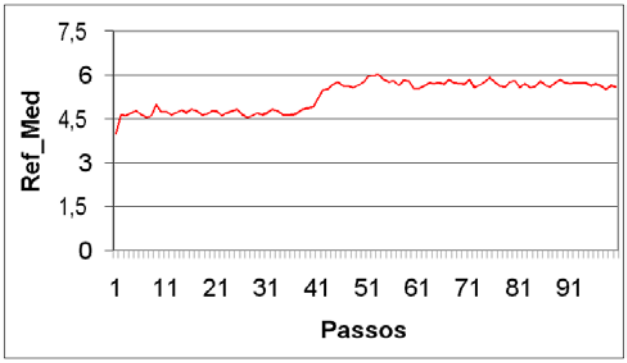

Figura V.73 – Gráfico do Reforço Médio do Modelo Não Determinístico M_2 para 1000 Simulações.

No gráfico da Figura V.74 usa-se a mesma política**(P_2)**, com valores de γe α igual a 0.9. A Figura V.74 mostra o resultado das médias dos valores de utilidade Q(s,a). Com base nesses valores, o algoritmo consegue definir (através da maximização sobre Q(s,a)) a ação de maior valor de utilidade no estado **s**. Ao longo do tempo, ocorreu convergência para uma política ótima de ações **(μ)** no modelo simulado.

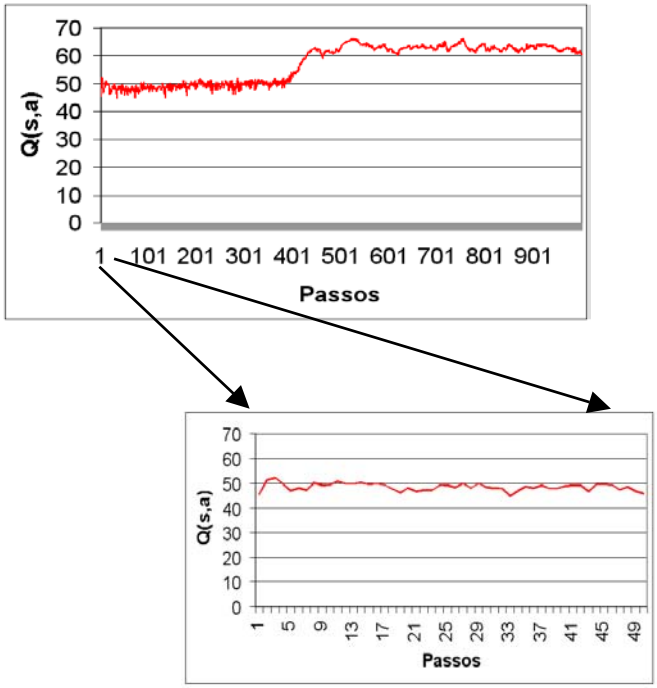

Figura V.74 – Gráfico do Q(s,a) do Modelo Não Determinístico M_2 para 1000 Simulações.

A Figura V.75 representa as médias das transições entre os estados cognitivos do modelo simulado. Importante observar que, para obter estes resultados, o protótipo do STI não conhece o modelo: os resultados da Figura V.75 são conseqüências da convergência do Q-Learning para uma política ótima de ações **(μ)**.

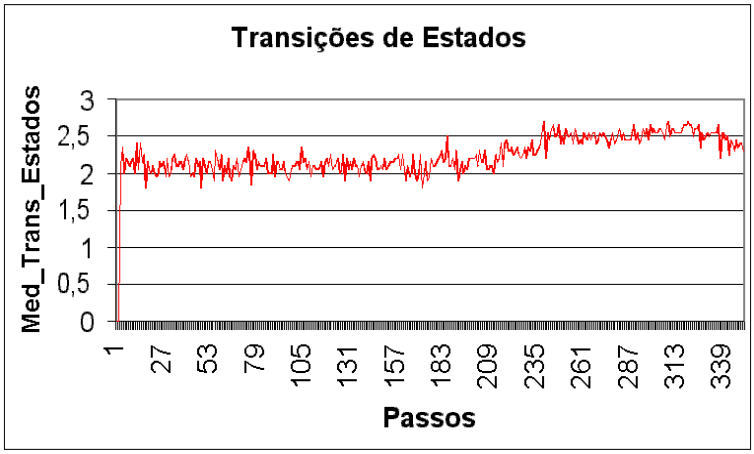

Figura V.75– Gráficos da Média das Transições dos Estados do Modelo Não Determinístico (M_2) para 1000 Simulações.

No gráfico da Figura V.76. observa-se que o algoritmo Q-Learning consegue, através da política ótima de ações **(μ)**, visitar mais o $\mathbf{E_2}$, modelo $\mathbf{M_2}$**(Bom)**, perfazendo um total de 81 % de visitas para este estado (que representa o modelo o qual se esperava obter com as simulações).

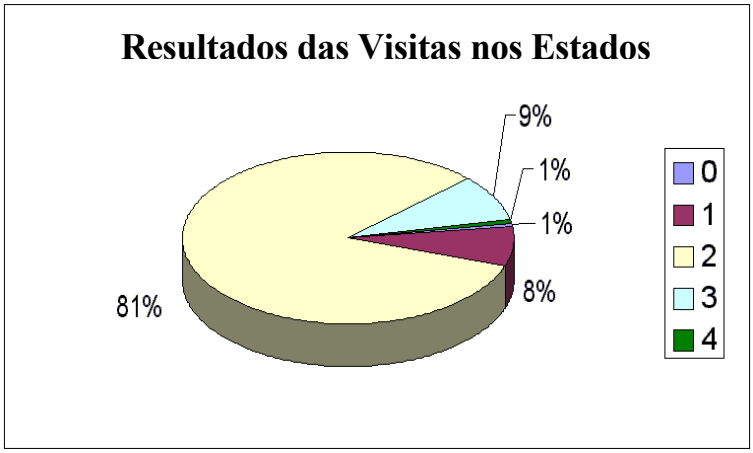

Figura V.76– Resultados das Visitas nos Estados

São apresentados os resultados da Política Pedagógica **(P_2)** com o modelo Não determinístico **M_2(Bom)**, seguindo uma simulação com 2500 passos, cujo γ é 0.9 e o α é 0.9. Os resultados obtidos na Figura V.77mostram as médias dos reforços obtidos durante a simulação para o modelo **M_2**. Esses resultados demonstram que o algoritmo Q-Learning seguindo uma política de ações **(μ)** conseguiu convergir para o modelo **M_2(Bom**.

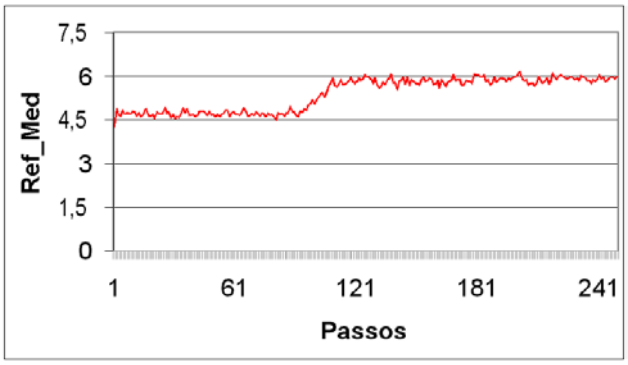

Figura V.77 – Gráfico do Reforço Médio do Modelo Não Determinístico M_2 para 2500 Simulações.

No gráfico da Figura V.78 usa-se a mesma política**(P_2)**, com valores de γ e α igual a 0.9. A Figura V.78 mostra o resultado das médias dos valores de utilidade Q(s,a). Com base nesses valores, o algoritmo consegue definir (através da maximização sobre Q(s,a)) a ação de maior valor de utilidade no estado **s**. Ao longo do tempo, ocorreu convergência para uma política ótima de ações **(μ)** no modelo simulado.

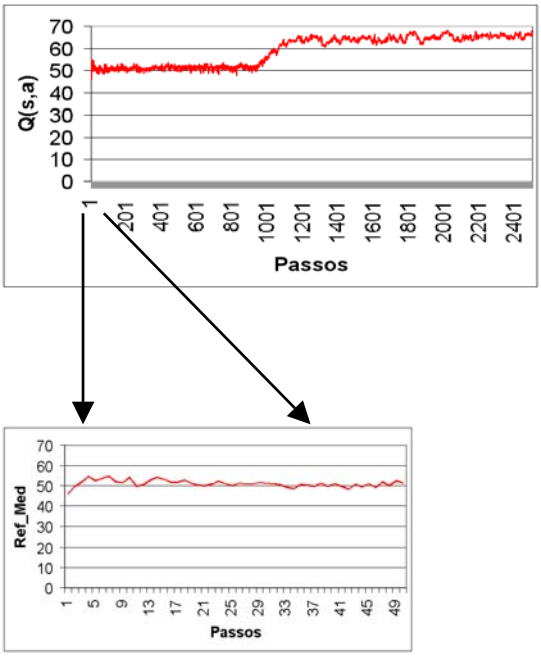

Figura V.78 – Gráfico do Q(s,a) do Modelo Não Determinístico M_2 para 2500 Simulações.

A Figura V.79 representa as médias das transições entre os estados cognitivos do modelo simulado. Importante observar que, para obter estes resultados, o protótipo do STI não conhece o modelo: os resultados da Figura V.79 são conseqüências da convergência do Q-Learning para uma política ótima de ações **(μ)**.

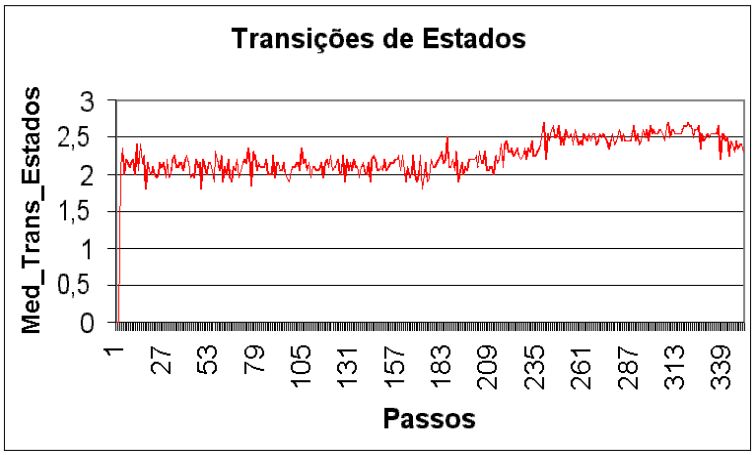

Figura V.79 – Gráficos da Média das Transições dos Estados do Modelo Não Determinístico (M_2) para 2500 Simulações.

No gráfico da Figura V.80. observa-se que o algoritmo Q-Learning consegue, através da política ótima de ações **(μ)**, visitar mais o $\mathbf{E_2}$, modelo $\mathbf{M_2}$**(Bom)**, perfazendo um total de 82 % de visitas para este estado (que representa o modelo o qual se esperava obter com as simulações).

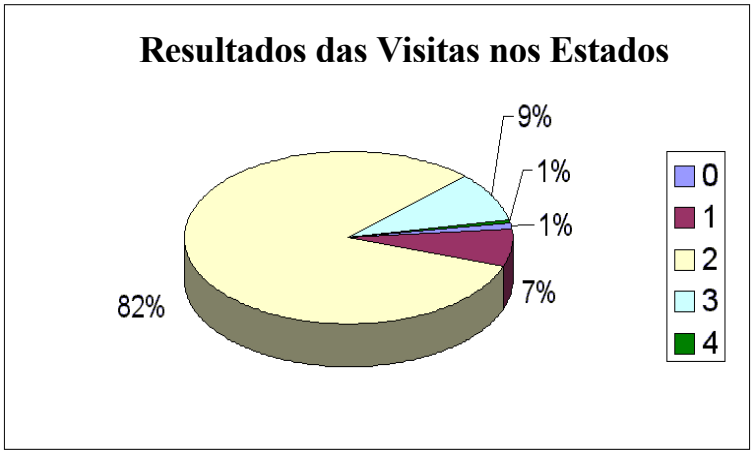

Figura V.80– Resultados das Visitas nos Estados.

São apresentados os resultados da Política Pedagógica (P2) com o modelo Não determinístico **M_3(Excelente)**, seguindo uma simulação com 500 passos, cujo γ é 0.9 e o α é 0.9. Os resultados obtidos na Figura V.81 mostram as médias dos reforços obtidos durante a simulação para o modelo **M_3**. Esses resultados demonstram que o algoritmo Q-Learning seguindo uma política de ações **(μ)** conseguiu convergir para o modelo **M_3(Excelente)**.

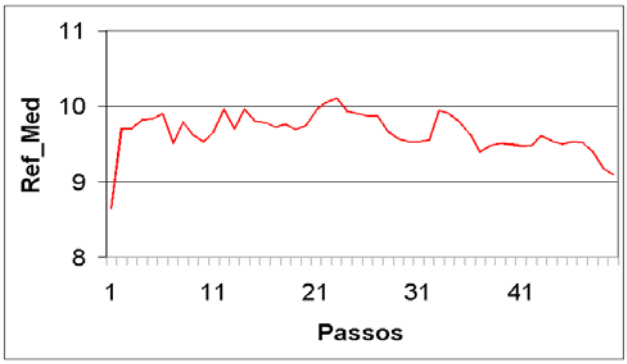

Figura V.81– Gráfico do Reforço Médio do Modelo Não Determinístico M_3 para 500 Simulações.

No gráfico da Figura V.82 usa-se a mesma política**(P_2)**, com valores de γ e α igual a 0.9. A Figura V.82 mostra o resultado das médias dos valores de utilidade Q(s,a). Com base nesses valores, o algoritmo consegue definir (através da maximização sobre Q(s,a)) a ação de maior valor de utilidade no estado **s**. Ao longo do tempo, ocorreu convergência para uma política ótima de ações **(μ)** no modelo simulado.

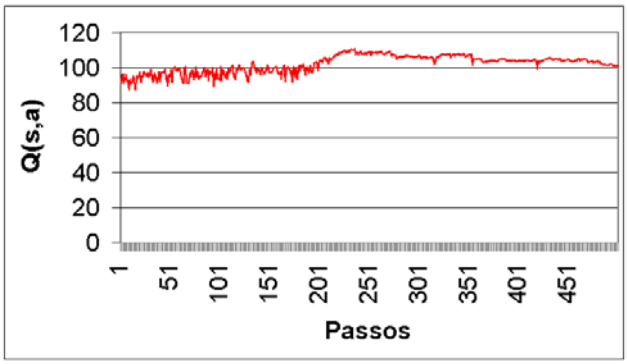

Figura V.82– Gráfico do Q(s,a) do Modelo Não Determinístico M_3 para 500 Simulações.

A Figura V.83 representa as médias das transições entre os estados cognitivos do modelo simulado. Importante observar que, para obter estes resultados, o protótipo do STI não conhece o modelo: os resultados da Figura V.83 são conseqüências da convergência do Q-Learning para uma política ótima de ações **(μ)**.

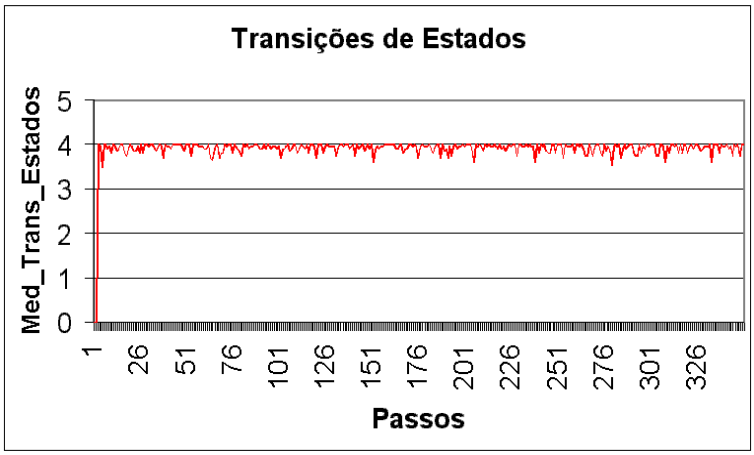

Figura V.83– Gráficos da Média das Transições dos Estados do Modelo Não Determinístico (M_3) para 500 Simulações.

No gráfico da Figura V.84. observa-se que o algoritmo Q-Learning consegue, através da política ótima de ações **(μ)**, visitar mais o **E_4**, modelo **M_3(Excelente)**, perfazendo um total de 96 % de visitas para este estado (que representa o modelo o qual se esperava obter com as simulações).

Figura V.84 – Resultados das Visitas nos Estados.

São apresentados os resultados da Política Pedagógica **(P_2)** com o modelo Não determinístico **M_3(Excelente)**, seguindo uma simulação com 1000 passos, cujo γ é 0.9 e o α é 0.9. Os resultados obtidos na Figura V.85 mostram as médias dos reforços obtidos durante a simulação para o modelo **M_3**. Esses resultados demonstram que o algoritmo Q-Learning seguindo uma política de ações **(μ)** conseguiu convergir para o modelo **M_3(Excelente)**.

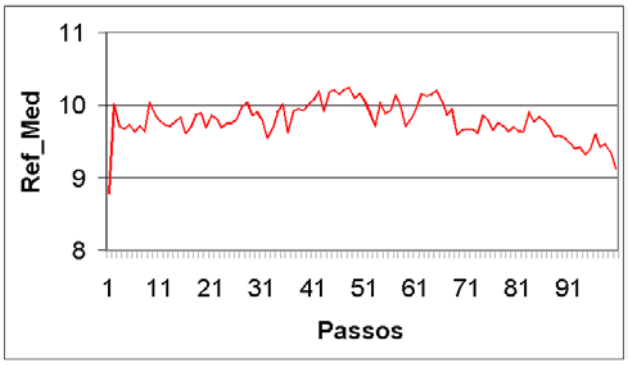

Figura V.85– Gráfico do Reforço Médio do Modelo Não Determinístico M_3 para 1000 Simulações.

No gráfico da Figura V.86 usa-se a mesma política**(P_2)**, com valores de γe α igual a 0.9. A Figura V.86 mostra o resultado das médias dos valores de utilidade Q(s,a). Com base nesses valores, o algoritmo consegue definir (através da maximização sobre Q(s,a)) a ação de maior valor de utilidade no estado **s**. Ao longo do tempo, ocorreu convergência para uma política ótima de ações **(μ)** no modelo simulado.

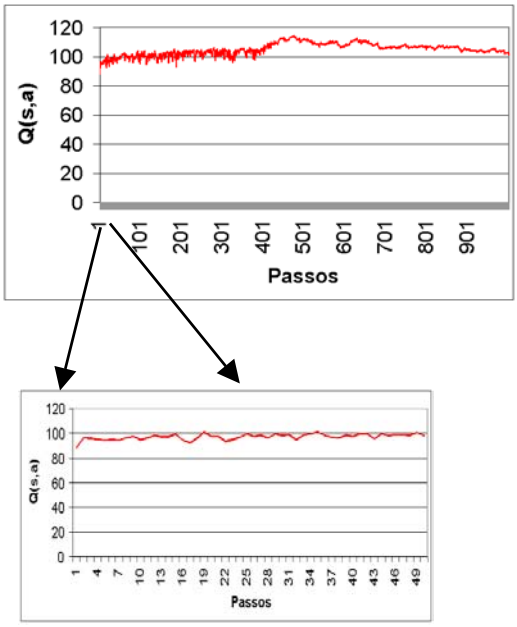

Figura V.86 – Gráfico do Q(s,a) do Modelo Não Determinístico M_3 para 1000 Simulações.

A Figura V.87 representa as médias das transições entre os estados cognitivos do modelo simulado. Importante observar que, para obter estes resultados, o protótipo do STI não conhece o modelo: os resultados da Figura V.87 são conseqüências da convergência do Q-Learning para uma política ótima de ações **(μ)**.

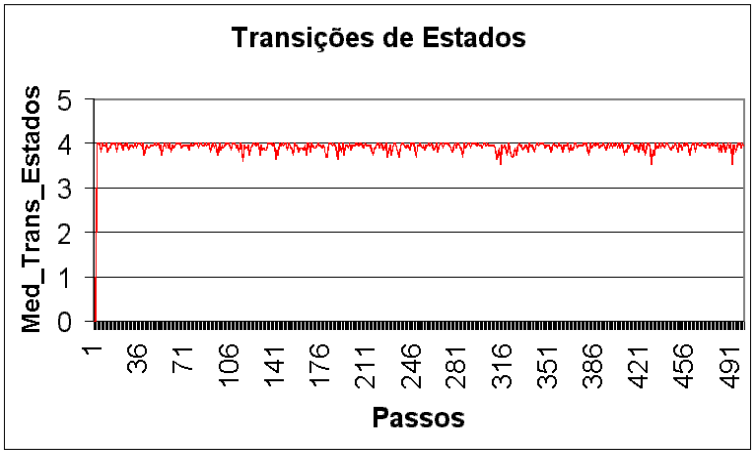

Figura V.87 – Gráficos da Média das Transições dos Estados do Modelo Não Determinístico (M_3) para 1000 Simulações.

No gráfico da Figura V.88. observa-se que o algoritmo Q-Learning consegue, através da política ótima de ações **(μ)**, visitar mais o **E_4**, modelo **M_3(Excelente)**, perfazendo um total de 93 % de visitas para este estado (que representa o modelo o qual se esperava obter com as simulações).

Figura V.88 – Resultados das Visitas nos Estados.

São apresentados os resultados da Política Pedagógica **(P_2)** com o modelo Não determinístico **M_3(Excelente)**, seguindo uma simulação com 2500 passos, cujo γ é 0.9 e o α é 0.9. Os resultados obtidos na Figura V.89 mostram as médias dos reforços obtidos durante a simulação para o modelo **M_3**. Esses resultados demonstram que o algoritmo Q-Learning seguindo uma política de ações **(μ)** conseguiu convergir para o modelo **M_3(Excelente)**

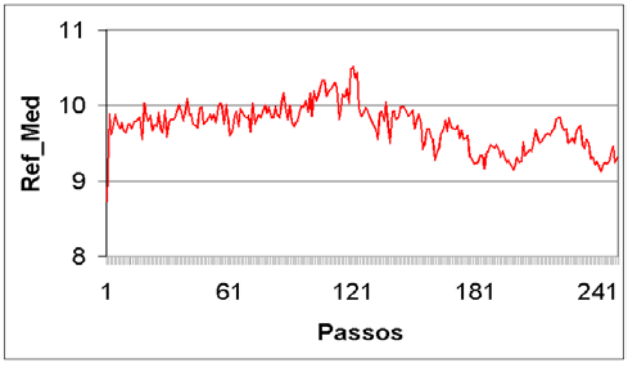

Figura V.89 – Gráfico do Reforço Médio do Modelo Não Determinístico M_3 para 2500 Simulações.

No gráfico da Figura V.90 usa-se a mesma política**(P_2)**, com valores de γ e α igual a 0.9. A Figura V.90 mostra o resultado das médias dos valores de utilidade Q(s,a). Com base nesses valores, o algoritmo consegue definir (através da maximização sobre Q(s,a)) a ação de maior valor de utilidade no estado **s**. Ao longo do tempo, ocorreu convergência para uma política ótima de ações **(μ)** no modelo simulado.

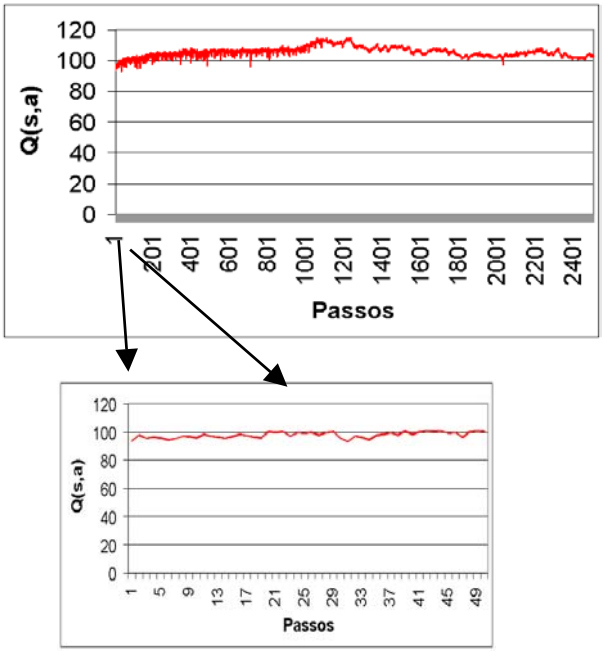

Figura V.90 – Gráfico do Q(s,a) do Modelo Não Determinístico M_3 para 2500 Simulações.

A Figura V.91 representa as médias das transições entre os estados cognitivos do modelo simulado. Importante observar que, para obter estes resultados, o protótipo do STI não conhece o modelo: os resultados da Figura V.91 são conseqüências da convergência do Q-Learning para uma política ótima de ações **(μ)**.

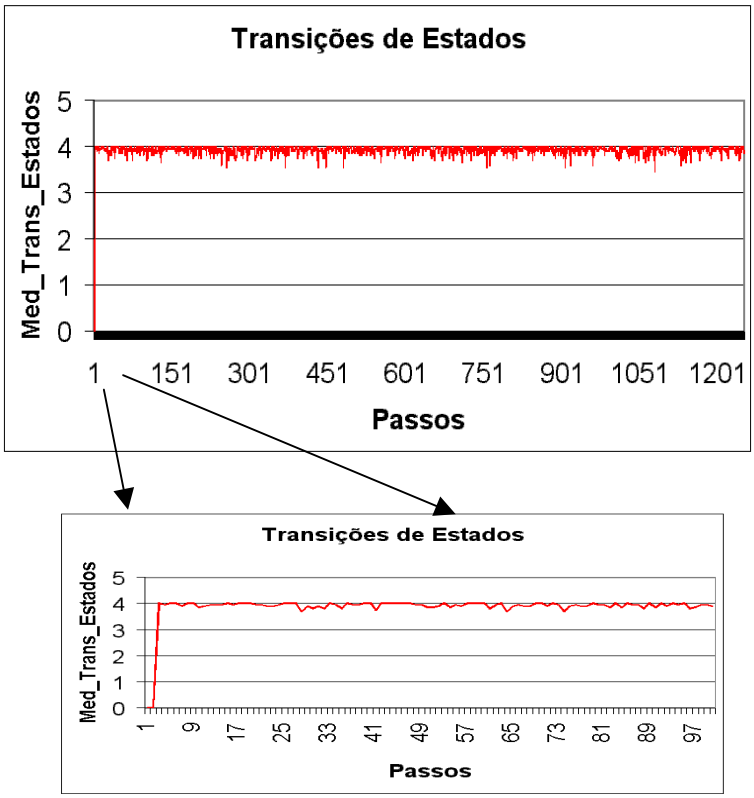

Figura V.91 – Gráficos da Média das Transições dos Estados do Modelo Não Determinístico (M_3) para 2500 Simulações.

No gráfico da Figura V.92. observa-se que o algoritmo Q-Learning consegue, através da política ótima de ações **(μ)**, visitar mais o **E_4**, modelo **M_3(Excelente)**, perfazendo um total de 94 % de visitas para este estado (que representa o modelo o qual se esperava obter com as simulações).

Figura V.92 – Resultados das Visitas nos Estados.

Capítulo VI

CONCLUSÕES, CONTRIBUIÇÕES E TRABALHOS FUTUROS

VI.1. CONCLUSÕES

A técnica de AR foi discutida neste trabalho, concluindo-se que é possível introduzir um módulo de diagnóstico na arquitetura de tutores, para modelagem autônoma do aprendiz..Através dos experimentos foi verificado que, dada uma política pedagógica, consegue-se maximizar as ações do agente tutor ao escolher ações para um dado estado do ambiente (aprendiz). Como conseqüência desta política, obtém-se o estado cognitivo do aprendiz, ou seja, consegue-se as melhores ações para o estado cognitivo inicial até chegar ao estado cognitivo final do aprendiz, previsto na política pedagógica.

É importante ressaltar que o protótipo usado na simulação não conhece os modelos obtidos, então a convergência se dá através do aprendizado da política ótima de ações, ou seja, o que foi determinado na política pedagógica.

O algoritmo Q-Learning, que usa uma tabela de pares estado-ação, representa os estados cognitivos do aprendiz e possibilita a escolha da melhor política a ser seguida. Desta forma, foram obtidos resultados que comprovam o ganho de autonomia do agente tutor em modelar um aprendiz, viabilizando a adoção do módulo diagnóstico na arquitetura tradicional dos tutores para modelar autonomamente o perfil do aprendiz.

Devido a fatores já explicados no capítulo II deste trabalho, o perfil do aprendiz é de difícil modelamento, ou seja, falta uma base teórica para representar satisfatoriamente o modelo do aprendiz. Com base nos resultados obtidos neste trabalho e a partir de uma fundamentação teórica da Ciência da Computação, esta tese contribui para a formalização da modelagem autônoma do aprendiz em Tutores Inteligentes.

VI.2. CONTRIBUIÇÕES E TRABALHOS FUTUROS

Uma contribuição deste trabalho foi o resultado positivo obtido na simulação com o uso do algoritmo Q-Learning aplicado a um agente tutor que inclui na sua arquitetura o módulo diagnóstico, para que desta forma se consiga dar um passo importante na questão da modelagem autônoma do aprendiz em um tutor inteligente. Acredita-se que este trabalho venha contribuir para a formalização computacional do problema de modelagem do estado cognitivo do aprendiz de forma dinâmica. Sendo assim, os resultados viabilizam a utilização de tutores inteligentes que utilizam o Sistema de AR em ambientes on-line, como por exemplo Internet.

Outra contribuição deste trabalho é deixar dados para estudos de viabilidade de desempenho no que se refere à redução do tempo de convergência do algoritmo, para uso mais efetivo de tutores inteligentes com o sistema de AR .

Além das contribuições citadas acima, podemos evidenciar possibilidades de trabalhos futuros, como:

- O interfaceamento de tutores inteligentes já existentes no mercado com o sistema de AR.
- Com base nesta arquitetura, desenvolver um sistema de Tutoria Inteligente, onde se possa utilizar múltiplas estratégias, independente do domínio.

  Com a estrutura desenvolvida neste trabalho, adaptá-la para o uso de multiagentes.

VI.3. DIFICULDADES ENCONTRADAS

Esta seção discute brevemente algumas dificuldades encontradas durante a elaboração desta dissertação. As principais dizem respeito à bibliografia e a implementação do algoritmo para a simulação.

Principais dificuldades:

- Bibliografia escassa sobre aplicação de Aprendizagem por Reforço em Sistemas Tutores Inteligentes.
- A implementação foi inicializada do zero, não havendo possibilidade de reaproveitamento de código.
- Dificuldade de achar um tutor que tivesse seu código fonte aberto para adaptá-lo ao sistema de AR.

VI.4. LIMITAÇÕES

As seguintes limitações devem ser consideradas no escopo deste trabalho:

- Pode existir um número elevado de ações a serem tomadas em determinado estado cognitivo do aprendiz. Dependendo do estado podem existir diversas estratégicas pedagógicas a serem adotadas.
- Não foi considerado o interfaceamento com tutores já existentes.
- Lentidão de convergência do algoritmo, que pode ser parcialmente compensada pelo uso de variantes baseadas em aproximações compactas ou planos de ação obtidos em simulação [Sutton & Barto 1998].

Em relação ao alcance e à extensibilidade, a solução apresentada pode teoricamente fazer uso de um número ilimitado de estados e de ações. Naturalmente, a expressão "número ilimitado" refere-se à capacidade teórica de armazenamento; sabe-se, contudo, que a memória disponível representa o limite físico real.

Outra restrição é com relação ao armazenamento e a geração dos valores para o algoritmo de aprendizagem. Os valores Q's do algoritmo Q-Learning são armazenados em matrizes, o que eventualmente pode limitar as dimensões das soluções desenvolvidas com a solução proposta. Uma solução possível, seria a substituição das matrizes por um mecanismo aproximador de função, como uma rede neural.

REFERÊNCIAS BIBLIOGRÁFICAS

ANDERSON, J. R. The expert module. In: POLSON, M. C.; ICHARDSON, J. J. (Eds) *Foundations on intelligent tutoring Systems* Hillsdale. N. Jersey: L Erlbaum Association Publishers. P 21-53, 1988.

ALMEIDA, M. E. *Informática e formação de professores.* [S.l.]: MEC, 2000. v.1
(Série de Estudos Educação a Distância)

BELLMAN, R. *Dynamic programming*. Princeton: Princeton University Press, 1957.

BROW, J.S.; BURTON, R.B. Pedagogical natural language and knowledge engineering techniques in SOPHIE I, II, and III. In: SLEEMAN, D.; BROWN,
J.S (Eds) *Intelligent tutoring systems***.** New York: Academic Press, 1982.

BURNS, H.; PARLETT, J. W.; REDFIELD, C. L. *Intelligent tutoring systems:* evolutions: I desing. [S.l.]: Lawrence Erlbaum-Verlag 1991.

BUSTAMANTE, S. *Ensinar e deixar aprender*: a formação do facilitador Logo. Petrópolis: Universidade Católica de Petrópolis, 1996.

CARBONELL,J. R. Al in CAI: an artificial intelligence approach to computer assisted instruction.*IEEE Transactions on Man-Machine Systems*, v. 11, n. 4, p.190-202, 1970.

CORREA, F. M. *Uma arquitetura de diálogos entre agentes cognitivos distribuídos.* 1994. Tese (Doutorado em Ciência da Computação) – Universidade Federal do Rio de Janeiro/COPPE, Rio de Janeiro.

COSTA, M. C. *Uma arquitetura baseada em agentes para suporte ao ensino à distância.* 1999. Tese. (Doutorado em Ciência da Computação) – Universidade Federal de Santa Catarina, Florianópolis.

DAMICO,C. *Modelo de usuário para sistemas tutores inteligentes*. Porto Alegre: CPGCC/UFRGS, 1995.(Exame de Qualificação em Profundidade).

DIETTERICH, T G. *Machine Learning,* Netherlands, V.22, Jan/Feb/Mar, 1996.

FRASSON, C.; MENGELLE, T.; AIMEUR, E. Using pedagogical agents in a multi-strategic intelligent tutoring system. In: WORLD CONFERENCE ON ARTIFICIAL INTELLIGENCE IN EDUCATION, 8.,1997. *Proceedings...* Kobe: AIED,1997. p.53-64

GALLEGARI, D. A. *Aplicando aprendizagem por reforço a uma arquitetura multiagente para suporte ao ensino de educação ambiental.* 2000. Dissertação. (Mestrado em Ciência da Computação) – Pontifícia Universidade Católica do Rio Grande do Sul, [S.l.].

GIRAFFA, L. M. M. *Uma arquitetura de tutor utilizando estados mentais* Rio Grande do Sul. 1999. Tese. (Doutorado em Ciência da Computação) – Pontifícia Universidade Católica do Rio Grande do Sul, [S.l.].

HALFF, H. M. Curriculum and isntruction in automated tutors. In: Poson, M. C.; RICHARDSON, J. J. (Eds) *Foundations on intelligent tutoring systems*. Hillsdale: L.Erlbaum Association Pub.,1988. p.79 -108

HENDLEY, R.; JURASCHECK, N. *Cascade*: introducing ai into cbt. *Computers Education*, v.18, n.1-3, p.71-76, 1992.

HOLLAND, J. H. Escaping brittleness: the possibilities os general-purpose learning algorithms applied to parallel rule-based systems. In: MICHALSKI, R. S.; CARBONELl, J. G.; MITCHELL, T. M. (Eds.) *Machine learning*: an artificial intelligence approach. [S.l.]: Morgan Kaufmann, 1986. v.2

HOPFIELD, J. J. Neural networks and physical systems with emergent collective computational abilities. In: NATIONAL ACADEMY OF SCIENCES OF THE U.S.A., 81, 1982. *Proceedings* … [S.l.: s.n.] 1982. p. 3088-3092

KAELBLING, L. P. The importance of being adaptable. In: The Artificial Life Route to Artificial Intelligence. Building Embodied, Situated Agents. Steels, Luc; Brooks, Rodney (eds.). Lawrence Erlbaum Assoc., USA, 1995.

KONZEN, A. A.; FROZZA, R. Clever tutor: an intelligent tutor system applying a hybrid strategy of teaching. In: INTERNATIONAL CONFERENCE ON CONPUTERS AND ADVANCED TECHNOLOGY IN EDUCATION, 2000, Cancun. *Proceedings ...* Cancun: [s.n], 2000.

LEITE, A. S. *Um modelo de sistema educativo cognitivista baseado em tutoria inteligente adaptativa via aderência conceitual.* 185f. 1999. Tese (Doutorado em Informática) - Instituto Tecnológico de Aeronáutica, São José dos Campos.

MARIETTO, M. G. *Definição dinâmica de estratégias instrucionais em sistema de tutoria inteligente*: uma abordagem multiagentes na www. 188f. 2000. Tese (Doutorado em Informática) - Instituto Tecnológico de Aeronáutica, São Jose dos Campos.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanet in nervous activity. *Bulletin of Mathematical Biophysics*, n. 5, p. 115 - 133, 1943.

MINSKY, M. L.; PAPERT, S. A. *Perceptrons.* Cambridge: MIT Press, 1969.

MORGAN, J.; MESSENGER, R. *THAID*: a sequential search program for the analysis of nominal scale dependent variables. [S.l.]: Institute for Social Research, University of Michigan, 1973. (Technical report)

OSWALDO, V. L.; ANGÉLICA, A. *Una revisión del estado de los sistemas cooperativos multiagente en la tutoría y la formación*. Madrid: Universidad Politécnica de Madrid, 2000.

QUINLAN, J. R. Generating production rules from decision trees. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 10., 1987, Milan. *Proceedings*... Milan: Morgan Kaufmann, 1987. p. 304-307

QUINLAN, J. R. *C4.5 programs for machine learning*. [S.l.]: Morgan Kaufmann,1988.

RIBEIRO, C. H. C. Embedding a priori knowledge in reinforcement learning. *Journal of Intelligent and Robotic Systems,* v. 2, n.1, p. 51 - 71, 1998.

ROSATELLI, M. C. *Um ambiente inteligente para aprendizado colaborativo no ensino a distância utilizando o método de casos*. 1999. Tese (Doutorado em Ciência da Computação) – Universidade Federal de Santa Catarina, Florianópolis.

ROSENBLATT, F. The Perceptron: a probabilistic model for information storage and organization in the brain. *Psycological Review*, n. 65, p. 386-408, 1958.

SABBATINI, R. *Modelagem e simulação em microcomputadores*. In: SEMINÁRIO BRASILEIRO DE TECNOLOGIA EDUCACIONAL, 16., 1984, PORTO Alegre. Anais... Porto Alegre: [s.n.], 1984. p.118-119

SILVEIRA, R. A.; BICA, F.; VICCARI, R. M. JADE: Java agents for distance education framework. In: IBEROAMERICAN WORKSHOP ON DISTRIBUTED ARTIFICIAL INTELLIGENCE AND MULTI-AGENT SYSTEMS, 2000, Atibaia. *Anais*... Atibaia: [S.l.], 2000. p.112–122.

SKINNER, B. F. Are theories of learning necessary*? Psychological Review*, v. 57, n. 4, p.193-216, 1950.

SLEEMAN, D. Assessing aspects of competence in basic algebra. In: SLEEMAN, D. ; BROWN, J. S. (Eds) *Intelligent tutoring systems***.** New York: Academic Press, 1982.

SUTTON, R. S.; BARTO A. G. Reforcement learning: an Introduction. Cambridge: MIT Press, 1998.

TEDESCO, P. A.; SELF, J. Using meta-cognitive conflicts to support group problem solving. In: INTELLIGENCE TUTORING SYSTEMS, 2000, Montreal. *Anais*... Montreal: ITS, 2000. p.112–119.

TSITSIKLIS, J.; BERTSEKAS, D. Neuro-dynamic programming. [S.l.]: Athena Scientific, 1996.

VANLENH, K. Student modeling. In: POLSON, C.; RICHARDSON, J. J. (EDS) *Foundations on itelligent tutoring systems*. Hillsdale: E. Erlbaum Association Pub., 1988. p 55-78

VALENTE, J. A. Diferentes usos do computador na educação. In: COMPUTADORES e conhecimento: repensando a educação. Campinas:Unicamp, 1993[a]

VICCARI, R. M. *Um tutor inteligente para a programação em lógica idealização, projeto e desenvolvimento*. 1990. Tese (Doutorado em Ciência da Computação) - Universidade de Coimbra, Coimbra.

WATKINS, C. J. H. *Learning from delayed rewards*. 1989. Thesis. (PhD in thesis) - University of Cambridge.

WATKINS, C. J.C.H.; DAYAN, P. Q-learning. *Machine Learning,* v. 8, n. ¾, p. 279-292, 1992.

WEIB, Gerhard; SEN, Sandip. Adaption and learning in multi-agent systems. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 14., 1995, Montreal. *Proceedings...* Montreal: Morgan Kaufmann, 1995.

ZOUAQ, A.; FRASSON, C.; ROUANE, K. The explanation agent. In: INTELLIGENCE TUTORING SYSTEMS, 2000, Montreal. *Proceedings* ... Montreal: [S.n.], 2000. p.112–119.

Anexo 1 – Política Pedagógica (P_1)

```
if (estado==0) /* Estado cognitivo E0*/
      {
       if ((acao>=0) && (acao<=5) && (acertos>=0 && acertos<=4))
              estadot=0;
       if ((acao>=0) && (acao<=5) && (acertos>4 && acertos<=6))
              estadot=1;
       if ((acao>=0) && (acao<=5) && (acertos>6 && acertos<=8))
              estadot=2;
       if ((acao>=0) && (acao<=5) && (acertos>8 && acertos<=10))
              estadot=3;
       if ((acao>=6) && (acao<=9) && (acertos>=0 && acertos<=5))
              estadot=0;
       if ((acao>=6) && (acao<=9) && (acertos>5 && acertos<=7))
              estadot=1;
       if ((acao>=6) && (acao<=9) && (acertos>7 && acertos<=8))
              estadot=2;
       if ((acao>=6) && (acao<=9) && (acertos>8 && acertos<=9))
              estadot=3;
       if ((acao>=6) && (acao<=9) && (acertos>9 && acertos<=10))
              estadot=4;
      }
if (estado==1) /* Estado cognitivo E1*/
      {
       if ((acao>=0) && (acao<=5) && (acertos>=0 && acertos<=4))
              estadot=0;
       if ((acao>=0) && (acao<=5) && (acertos>4 && acertos<=6))
              estadot=1;
       if ((acao>=0) && (acao<=5) && (acertos>6 && acertos<=10))
              estadot=2;
       if ((acao>=6) && (acao<=9) && (acertos>=0 && acertos<=5))
              estadot=0;
       if ((acao>=6) && (acao<=9) && (acertos>5 && acertos<=7))
              estadot=1;
       if ((acao>=6) && (acao<=9) && (acertos>7 && acertos<=8))
              estadot=2;
       if ((acao>=6) && (acao<=9) && (acertos>8 && acertos<=9))
              estadot=3;
       if ((acao>=6) && (acao<=9) && (acertos>9 && acertos<=10))
              estadot=4;
```

```
      }
if (estado==2) /* Estado cognitivo E2*/
      {
       if ((acao>=0) && (acao<=5) && (acertos>=0 && acertos<=5))
              estadot=0;
       if ((acao>=0) && (acao<=5) && (acertos>5 && acertos<=6))
              estadot=1;
       if ((acao>=0) && (acao<=5) && (acertos>6 && acertos<=8))
              estadot=2;
       if ((acao>=0) && (acao<=5) && (acertos>8 && acertos<=10))
              estadot=3;
       if ((acao>=6) && (acao<=9) && (acertos>=0 && acertos<=7))
              estadot=1;
       if ((acao>=6) && (acao<=9) && (acertos>7 && acertos<=8))
              estadot=2;
       if ((acao>=6) && (acao<=9) && (acertos>8 && acertos<=9))
              estadot=3;
       if ((acao>=6) && (acao<=9) && (acertos>9 && acertos<=10))
              estadot=4;
      }
if (estado==3) /* Estado cognitivo E3*/
      {
       if ((acao>=0) && (acao<=5) && (acertos>=0 && acertos<=7))
              estadot=0;
       if ((acao>=0) && (acao<=5) && (acertos>7 && acertos<=9))
              estadot=1;
       if ((acao>=0) && (acao<=5) && (acertos>9 && acertos<=10))
              estadot=2;
       if ((acao>=6) && (acao<=7) && (acertos>=0 && acertos<=5))
              estadot=0;
       if ((acao>=6) && (acao<=7) && (acertos>5 && acertos<=7))
              estadot=1;
       if ((acao>=6) && (acao<=7) && (acertos>7 && acertos<=8))
              estadot=2;
       if ((acao>=6) && (acao<=7) && (acertos>8 && acertos<=9))
              estadot=3;
       if ((acao>=6) && (acao<=7) && (acertos>9 && acertos<=10))
              estadot=4;
       if ((acao>=8) && (acao<=9) && (acertos>=0 && acertos<=5))
              estadot=1;
       if ((acao>=8) && (acao<=9) && (acertos>5 && acertos<=7))
```

```
                estadot=2;
        if ((acao>=8) && (acao<=9) && (acertos>7 && acertos<=9))
                estadot=3;
        if ((acao>=8) && (acao<=9) && (acertos>9 && acertos<=10))
                estadot=4;
       }
if (estado==4) /* Estado cognitivo E4*/
       {
        if ((acao>=0) && (acao<=5) && (acertos>=0 && acertos<=7))
                estadot=0;
        if ((acao>=0) && (acao<=5) && (acertos>7 && acertos<=8))
                estadot=1;
        if ((acao>=0) && (acao<=5) && (acertos>8 && acertos<=9))
                estadot=2;
        if ((acao>=0) && (acao<=5) && (acertos>9 && acertos<=10))
                estadot=3;
        if ((acao>=6) && (acao<=8) && (acertos>=0 && acertos<=6))
                estadot=1;
        if ((acao>=6) && (acao<=8) && (acertos>6 && acertos<=7))
                estadot=2;
        if ((acao>=6) && (acao<=8) && (acertos>7 && acertos<=8))
                estadot=3;
        if ((acao>=6) && (acao<=8) && (acertos>8 && acertos<=10))
                estadot=4;
        if ((acao==9) && (acertos>=0 && acertos<=7))
                estadot=2;
        if ((acao==9) && (acertos>7 && acertos<=8))
                estadot=3;
        if ((acao==9) && (acertos>8 && acertos<=10))
                estadot=4;
       }
```

Anexo 2 – Política Pedagógica (P_2)

```
      /*Política Pedagogica 2*/
      if (estado==0) // Estado cognitivo E0
      {
       if ((acao>=0) && (acao<=5) && (acertos>=0 && acertos<=5))
                 estadot=0;
       if ((acao>=0) && (acao<=5) && (acertos>5 && acertos<=7))
                 estadot=1;
       if ((acao>=0) && (acao<=5) && (acertos>7 && acertos<=8))
                 estadot=2;
       if ((acao>=0) && (acao<=5) && (acertos>8 && acertos<=9))
                 estadot=3;
       if ((acao>=0) && (acao<=5) && (acertos==10))
                 estadot=4;
       if ((acao>=6) && (acao<=9) && (acertos>=0 && acertos<=3))
                 estadot=0;
       if ((acao>=6) && (acao<=9) && (acertos>3 && acertos<=6))
                 estadot=1;
       if ((acao>=6) && (acao<=9) && (acertos>6 && acertos<=8))
                 estadot=2;
       if ((acao>=6) && (acao<=9) && (acertos>8 && acertos<=9))
                 estadot=3;
       if ((acao>=6) && (acao<=9) && (acertos>9 && acertos<=10))
                 estadot=4;
      }
if (estado==1) //Estado cognitivo E1
      {
       if ((acao>=0) && (acao<=5) && (acertos>=0 && acertos<=4))
                 estadot=0;
       if ((acao>=0) && (acao<=5) && (acertos>4 && acertos<=7))
                 estadot=1;
       if ((acao>=0) && (acao<=5) && (acertos>7 && acertos<=8))
                 estadot=2;
       if ((acao>=0) && (acao<=5) && (acertos>8 && acertos<=9))
                 estadot=3;
       if ((acao>=0) && (acao<=5) && (acertos==10))
                 estadot=4;
       if ((acao>=6) && (acao<=9) && (acertos>=0 && acertos<=2))
                 estadot=0;
       if ((acao>=6) && (acao<=9) && (acertos>2 && acertos<=6))
```

```
                    estadot=1;
        if ((acao>=6) && (acao<=9) && (acertos>6 && acertos<=8))
                    estadot=2;
        if ((acao>=6) && (acao<=9) && (acertos>8 && acertos<=9))
                    estadot=3;
        if ((acao>=6) && (acao<=9) && (acertos>9 && acertos<=10))
                    estadot=4;
      }
if (estado==2) // Estado cognitivo E2
      {
        if ((acao>=0) && (acao<=5) && (acertos>=0 && acertos<=5))
                    estadot=0;
        if ((acao>=0) && (acao<=5) && (acertos>5 && acertos<=7))
                    estadot=1;
        if ((acao>=0) && (acao<=5) && (acertos>7 && acertos<=8))
                    estadot=2;
        if ((acao>=0) && (acao<=5) && (acertos>8 && acertos<=9))
                    estadot=3;
        if ((acao>=0) && (acao<=5) && (acertos==10))
                    estadot=4;
        if ((acao>=6) && (acao<=9) && (acertos>=0 && acertos<=4))
                    estadot=0;
        if ((acao>=6) && (acao<=9) && (acertos>4 && acertos<=7))
                    estadot=1;
        if ((acao>=6) && (acao<=9) && (acertos>7 && acertos<=8))
                    estadot=2;
             if ((acao>=6) && (acao<=9) && (acertos>8 && acertos<=9))
                    estadot=3;
        if ((acao>=6) && (acao<=9) && (acertos>9 && acertos<=10))
                    estadot=4;
      }
```

```
if (estado==3) // Estado cognitivo E3
      {
       if ((acao>=0) && (acao<=5) && (acertos>=0 && acertos<=6))
                  estadot=0;
       if ((acao>=0) && (acao<=5) && (acertos>6 && acertos<=8))
                  estadot=1;
       if ((acao>=0) && (acao<=5) && (acertos>8 && acertos<=8))
                  estadot=2;
       if ((acao>=0) && (acao<=5) && (acertos>8 && acertos<=9))
                  estadot=3;
       if ((acao>=0) && (acao<=5) && (acertos==10))
                  estadot=4;
       if ((acao>=6) && (acao<=9) && (acertos>=0 && acertos<=2))
                  estadot=0;
       if ((acao>=6) && (acao<=9) && (acertos>2 && acertos<=5))
                  estadot=1;
       if ((acao>=6) && (acao<=9) && (acertos>5 && acertos<=7))
                  estadot=2;
       if ((acao>=6) && (acao<=9) && (acertos>7 && acertos<=9))
                  estadot=3;
       if ((acao>=6) && (acao<=9) && (acertos>9 && acertos<=10))
                  estadot=4;
      }
if (estado==4) // Estado cognitivo E4
      {
       if ((acao>=0) && (acao<=5) && (acertos==0))
                  estadot=0;
       if ((acao>=0) && (acao<=5) && (acertos>=1 && acertos<=4))
                  estadot=1;
       if ((acao>=0) && (acao<=5) && (acertos>4 && acertos<=7))
                  estadot=2;
       if ((acao>=0) && (acao<=5) && (acertos>7 && acertos<=9))
                  estadot=3;
       if ((acao>=0) && (acao<=5) && (acertos==10))
                  estadot=4;
       if ((acao>=6) && (acao<=9) && (acertos==0))
                  estadot=0;
       if ((acao>=6) && (acao<=9) && (acertos>=1 && acertos<=4))
                  estadot=1;
       if ((acao>=6) && (acao<=9) && (acertos>4 && acertos<=7))
                  estadot=2;
```

```
 if ((acao>=6) && (acao<=9) && (acertos>7 && acertos<=8))
            estadot=3;
      if ((acao>=6) && (acao<=9) && (acertos>8 && acertos<=10))
            estadot=4;
}
```

Anexo 3 – Modelo M1(Ruim) - Determinístico

```
/*Modelo1*/
                  if(estado==0)
                        {
                        if ((acao>=0) && (acao<=5))//Permanece em E0
                              {
                                    acertos=0;
                                    erros=10;
                                    //Acao Boa-Estado desejado para o
                                      Modelo1
                              }
                              if((acao>=6)&&(acao<=9))//Vai para E1
                              {
                              acertos=6;
                              erros=4;
                              ref=-3; //Ref 3-3=0->Evita ficar entre os E0
                                          E1 e E2

                              }
                        }
                   if(estado==1)
                        {
                        if((acao>=0) && (acao<=5))//Vai para E2
                              {
                              acertos=7;
                              erros=3;
                              ref=-6;  //Ref  5-6=-1->Tentar  acao  com  valor
                                              cognitivo maior

                              }
                        if((acao>=6)&&(acao<=9))//Volta para E0
                         {
                              acertos=0;
                              erros=10;
                              ref=-2; //Ref=1-2=-1->Acao ruim penalizacao
                         }
                        }
```

```
if(estado==2)
                        {
                        if((acao>=0)&&(acao<=5)) //Vai para E3
                              {
                                    acertos=9;
                                    erros=1;
                                    ref=-8;//Ref 7-8=-1->Tentar acao com
                                             valor cognitivo maior
                              }
                         if((acao>=6)&&(acao<=9))//Volta para E0
                              {
                                    acertos=0;
                                    erros=10;
                                    ref=-2;//Ref=1-2=-1->Acao ruim
                                             penalizacao
                              }
                        }

                  if(estado==3)
                        {
                        if((acao>=0) && (acao<=5))//Volta ao  E0
                              {
                               acertos=0;
                               erros=10;
                               ref=-2;  //Ref  1-2=-1->Tentar  acao  com  valor
                                               cognitivo maior
                              }
                        if((acao>=6)&&(acao<=9))//Volta ao E0
                              {
                               acertos=0;
                               erros=10;
                               ref=-2; //Ref=1-2=-1->Acao ruim penalizacao
                              }
                        }
                   if(estado==4)//O modelo 1 nao preve transicoes para
                                  este Estado
                        {
                        if((acao>=0) && (acao<=5))//Volta ao E0
                              {
                               acertos=0;
                               erros=10;
```

```
        ref=-2;//Ref=1-2=-1->Acao ruim penalizacao
      }
if((acao>=6) && (acao<=9))//Volta ao E0
      {
            acertos=0;
            erros=10;
      ref=-2;//Ref=1-2=-1->Acao ruim
                     penalizacao
      }
}
```

Anexo 4 – Modelo M2(Bom) - Determinístico

```
/* Modelo2 */
                    if(estado==0)
                              {
                               if ((acao>=0) && (acao<=5))//Permanece em
                                                              E0
                                    {
                                          acertos=0;
                                          erros=10;
                                          ref=-2; //Ref 1-2=-1->Tentar
                                          acao com valor cognitivo maior
                                    }
                                    if((acao>=6)&&(acao<=9))//Vai para o
                                                               E1
                                    {
                                          acertos=7;
                                          erros=3;
                                    }
                         }
                    if(estado==1)
                         {
                              if((acao>=0) && (acao<=5))//Permanece em  E1
                                    {
                                          acertos=6;
                                          erros=4;
                                          ref=-4; //Ref 3-4=-1->Tentar
                                           acao com valor cognitivo maior
                                    }
                              if((acao>=6)&&(acao<=9))//Vai para o E2
                                    {
                                          acertos=8;
                                          erros=2;
                                          ref=-5;//Ref 5-5=0 Evita ficar
                                                   entre os E1 e E2
                                    }
                         }
                    if(estado==2)
                         {
                              if((acao>=0)&&(acao<=5)) //Permanece em E2
                                    {
```

```
                        acertos=8;
                        erros=2;
                        //Acao Boa-Estado desejado para
                          o Modelo2
                  }
            if((acao>=6)&&(acao<=9))//Vai para  E3
                  {
                        acertos=9;
                        erros=1;
                        ref=-7;//Ref 7-7=0 Evita ficar
                                 entre os E2 e E3
                  }
      }
if(estado==3)
      {
            if((acao>=0) && (acao<=5))//Permanece em E3
                  {
                        acertos=9;
                        erros=1;
                        ref=-8; //Ref 7-8=-1->Tentar
                        acao com valor cognitivo maior
                  }
            if((acao>=6)&&(acao<=7))//Volta ao E2
                  {
                        acertos=8;
                        erros=2;
                        ref=-6; //Ref=5-6=-1->Acao ruim
                                penalizacao
                  }
            if((acao>=8)&&(acao<=9))//Vai para E4
                  {
                        acertos=10;
                        erros=0;
                        ref=-10;//Ref 10-10=0 Evita
                        ficar entre os E2 e E4
                  }
      }
if(estado==4)
      {
            if((acao>=0) && (acao<=5))//Volta ao E2
                  {
```

```
                        acertos=9;
                        erros=1;
                        ref=-6; //Ref=5-6=-1->Acao ruim
                        penalizacao
                  }
            if((acao>=6) && (acao<=8))//Volta para E2
                  {
                        acertos=7;
                        erros=3;
                        ref=-6;//Ref=5-6=-1->Acao ruim
                        penalizacao
                  }
            if(acao==9)//Volta ao  E2
                  {
                        acertos=5;
                        erros=5;
                        ref=-6;//Ref=5-6=-1->Acao ruim
                        penalizacao
                  }
}
```

Anexo 5 – Modelo M3(Excelente) - Determinístico

```
/* Modelo3 */
                    if(estado==0)
                         {
                               if ((acao>=0) && (acao<=5))//Vai para E3
                                     {
                                           acertos=10;
                                           erros=0;
                                     }
                               if((acao>=6)&&(acao<=9))//Vai para E4
                                     {
                                      acertos=10;
                                      erros=0;
                                      ref=-10;//Ref=10-10=0-> Evita ficar
                                      entre os E0 e E4
                                     }
                         }
                    if(estado==1)
                         {
                               if((acao>=0) && (acao<=5))//Vai para E2
                                     {
                                           acertos=10;
                                           erros=0;
                                           ref=-6;//Ref 5-6=-1->Tentar ação
                                           com valor cognitivo maior
                                     }
                               if((acao>=6)&&(acao<=9))//Vai para E4
                                     {
                                           acertos=10;
                                           erros=0;
                                           ref=-10;//Ref=10-10=0-> Evita
                                           ficar entre os E1 e E4
                                     }
                         }
                    if(estado==2)
                         {
                               if((acao>=0)&&(acao<=5))//Vai para E1
                                     {
                                           acertos=6;
                                           erros=4;
```

```
                ref=-4;//Ref 3-4=-1->Tentar ação
                com valor cognitivo maior
            }
        if((acao>=6)&&(acao<=9))//Vai para E4
            {
                acertos=10;
                erros=0;
                ref=-10;//Ref=10-10=0-> Evita
                ficar entre os E2 e E4
            }
    }
```

```
if(estado==3)
      {
            if((acao>=0) && (acao<=5))//Volta ao E2
                  {
                        acertos=10;
                        erros=0;
                        ref=-6; //Ref 5-6=-1->Tentar
                        acao com valor cognitivo maior
                  }
            if((acao>=6)&&(acao<=7))//Permanece em E3
                  {
                        acertos=9;
                        erros=1;
                        ref=-8;//Ref 7-8=-1->Tentar ação
                        com valor cognitivo maior
                  }
            if((acao>=8)&&(acao<=9))//Vai para E4
                  {
                        acertos=10;
                        erros=0;
                        ref=-10;//Ref 10-10=0 Evita
                        ficar entre os E3 e E4
                  }
      }
 if(estado==4)
      {
            if((acao>=0) && (acao<=5))//Volta para E2
                  {
                        acertos=9;
                        erros=1;
                        ref=-6;//Ref=5-6=-1->Acao ruim
                        penalizacao
                  }
            if((acao>=6) && (acao<=8))//Volta em E3
                  {
                        acertos=8;
                        erros=2;
                        ref=-8;//Ref=7-8=-1->Acao ruim
                        penalizacao
                  }
            if(acao==9)//Permanece em E4
```

```
                    {
                        acertos=10;
                        erros=0;
                        //Acao Boa-Estado desejado para
                          o Modelo3
                    }
            }
```

Anexo 6 – Modelo M1(Ruim) – Não Determinístico

```
/* Modelo 1 */
aux1=rand()%20;
if (estado==0)
 {

      if(((acao>=0)&&(acao<=5))&&((aux1>=0)&&(aux1<=16)))
                                //80% Permanecer em E0
                               {
                                acertos=0;
                                erros=10;
                               }

      if(((acao>=0)&&(acao<=5))&&((aux1>=17)&&(aux1<=18)))
                                //15% Ir para E1
                               {
                                acertos=6;
                                erros=4;
                               }
                        if(((acao>=0)&&(acao<=5))&&(aux1==19))
                                // 5% Ir para E2
                               {
                                acertos=8;
                                erros=2;
                               }

      if(((acao>=6)&&(acao<=9))&&((aux1>=0)&&(aux1<=17)))
                                //85% Permanecer em E0
                               {
                                acertos=0;
                                erros=10;
                               }
                               if(((acao>=6)&&(acao<=9))&&(aux1==18))
                                //10% Ir para E1
                        {
                               acertos=4;
                               erros=6;
                        }
                        if (aux1==19)
                        {
```

```
                                aux2=rand()%5;

        if(((acao>=6)&&(acao<=9))&&((aux2>=0)&&(aux2<=3)))
                                      //4% Ir para E2
                                {
                                     acertos=7;
                                     erros=3;
                                }
                                     if (aux2==4)
                                     {
                                           aux3=rand()%2;

        if(((acao>=6)&&(acao<=9))&&(aux3==0))
                                                 //0.5% Ir para E3
                                           {
                                            acertos=9;
                                            erros=1;
                                           }

        if(((acao>=6)&&(acao<=9))&&(aux3==1))
                                                 //0.5% Ir para E4
                                           {
                                            acertos=10;
                                            erros=0;
                                           }
                                     }
                           }
   }
  if (estado==1)
   {

        if(((acao>=0)&&(acao<=5))&&((aux1>=0)&&(aux1<=17)))
                                //90% Voltar para E0
                               {
                                acertos=0;
                                erros=10;
                               }
                               if(((acao>=0)&&(acao<=5))&&(aux1==18))
                                //5% Permanecer em E1
                               {
                                acertos=6;
```

```
                        erros=4;
                       }
                 if(((acao>=0)&&(acao<=5))&&(aux1==19))
                        //5% Ir para E2
                       {
                        acertos=8;
                        erros=2;
                       }

if(((acao>=6)&&(acao<=9))&&((aux1>=0)&&(aux1<=18)))
                        //95% Voltar para E0
                       {
                        acertos=0;
                        erros=10;
                       }
                       if (aux1==19)
                        {
                             aux2=rand()%5;

if(((acao>=6)&&(acao<=9))&&((aux2>=0)&&(aux2<=3)))
                                     //4% Permanecer em E1
                             {
                                  acertos=6;
                                  erros=4;
                             }
                             if (aux2==4)
                                  {
                                   aux3=rand()%5;

if(((acao>=6)&&(acao<=9))&&((aux3>=0)&&(aux3<=2)))
                                            //0.5% Ir para E2
                                        {
                                         acertos=8;
                                         erros=2;
                                        }

if(((acao>=6)&&(acao<=9))&&(aux3==3))
                                            //0.25% Ir para E3
                                        {
                                         acertos=9;
                                         erros=1;
```

```
                                                }

      if(((acao>=6)&&(acao<=9))&&(aux3==4))
                                                //0.25% Ir para em E4
                                                {
                                                 acertos=10;
                                                 erros=0;
                                                }
                                 }
                        }
 }
if (estado==2)
 {
                            if(((acao>=0)&&(acao<=5))&&((aux1>=0)&&(aux1<=18)))
                                  //95% Voltar em E0
                             {
                                  acertos=0;
                                  erros=10;
                             }
                            if (aux1==19)
                             {
                                  aux2=rand()%5;

      if(((acao>=0)&&(acao<=5))&&((aux2>=0)&&(aux2<=3)))
                                        //4% Voltar para E1
                                 {
                                      acertos=6;
                                      erros=4;
                                 }
                                  if(((acao>=0)&&(acao<=5))&&(aux2==4))
                                        //1% permanecer em E2
                                 {
                                      acertos=8;
                                      erros=2;
                                 }
                             }

      if(((acao>=6)&&(acao<=9))&&((aux1>=0)&&(aux1<=18)))
                                      //95% Voltar para E0
                                  {
                                   acertos=10;
```

```
                                erros=0;
                               }
                          if (aux1==19)
                               {
                                     aux2=rand()%5;

        if(((acao>=6)&&(acao<=9))&&((aux2>=0)&&(aux2<=3)))
                                           //4% Voltar para E1
                                     {
                                           acertos=6;
                                           erros=4;
                                     }
                                     if (aux2==4)
                                           {
                                            aux3=rand()%5;

        if(((acao>=6)&&(acao<=9))&&((aux3>=0)&&(aux3<=2)))
                                            //0.5% Permanecer em E2
                                     {
                                           acertos=8;
                                           erros=2;
                                     }
                                    if(((acao>=6)&&(acao<=9))&&(aux3==3))
                                            //0.25% Ir para E3
                                     {
                                           acertos=9;
                                           erros=1;
                                     }
                                  if(((acao>=6)&&(acao<=9))&&(aux3==4))
                                     //0.25% Ir para E4
                                    {
                                     acertos=10;
                                     erros=0;
                                    }
                               }
                        }
   }
  if (estado==3)
   {
                            if(((acao>=0)&&(acao<=5))&&((aux1>=0)&&(aux1<=18)))
                               //95% Voltar em E0
```

```
                    {
                         acertos=0;
                         erros=10;
                    }
                    if (aux1==19)
                         {
                          aux2=rand()%5;

if(((acao>=0)&&(acao<=5))&&((aux2>=0)&&(aux2<=3)))
                                        //4% Voltar para E1
                              {
                               acertos=6;
                               erros=4;
                              }
                          if(((acao>=0)&&(acao<=5))&&(aux2==4))
                                        //1% Voltar para E2
                              {
                               acertos=8;
                               erros=2;
                              }
                         }

if(((acao>=6)&&(acao<=9))&&((aux1>=0)&&(aux1<=18)))
                          //95% Voltar para E0
                         {
                          acertos=0;
                          erros=10;
                         }
                    if (aux1==19)
                         {
                          aux2=rand()%5;

if(((acao>=6)&&(acao<=9))&&((aux2>=0)&&(aux2<=3)))
                               //4% Voltar para E1
                              {
                               acertos=6;
                               erros=4;
                              }
                          if (aux2==4)
                              {
                               aux3=rand()%5;
```

```
      if(((acao>=6)&&(acao<=9))&&((aux3>=0)&&(aux3<=2)))
                                                //0.5% Voltar para E2
                                          {
                                           acertos=8;
                                           erros=2;
                                          }
                                     if(((acao>=6)&&(acao<=9))&&(aux3==3))
                                                //0.25% Permanecer em E3
                                          {
                                           acertos=9;
                                           erros=1;
                                          }
                                     if(((acao>=6)&&(acao<=9))&&((aux3==4)))
                                                //0.25% Ir para E4
                                          {
                                           acertos=10;
                                           erros=0;
                                          }
                                    }
                          }
 }
if (estado==4)
 {

      if(((acao>=0)&&(acao<=5))&&((aux1>=0)&&(aux1<=18)))
                                    //95% Voltar em E0
                               {
                                    acertos=0;
                                    erros=10;
                               }
```

```
                    if(((acao>=0)&&(acao<=5))&&(aux1==19))
                                //5% Voltar para E1
                            {
                                acertos=6;
                                erros=4;
                            }

    if(((acao>=6)&&(acao<=9))&&((aux1>=0)&&(aux1<=19)))
                            //100% Voltar para E0
                            {
                                acertos=0;
                                erros=10;
                            }
                    }
}
```

Anexo 7– Modelo M2(Bom) – Não Determinístico

```
/* Modelo2 */
aux1=rand()%20;
if (estado==0)
{
            if(((acao>=0)&&(acao<=5))&&((aux1>=0)&&(aux1<=1)))
            {
                  aux2=rand()%10;
                  if(((acao>=0)&&(acao<=5))&&((aux2>=0)&&(aux2<=7)))
                        //8% Ir para E3
                  {
                   acertos=9;
                   erros=1;
                  }
                  if(((acao>=0)&&(acao<=5))&&(aux2==8))
                        //1% Ir para E1
                  {
                   acertos=6;
                   erros=4;
                  }
                  if (aux2==9)
                  {
                         aux3=rand()%2;
                         if(((acao>=0)&&(acao<=5))&&(aux3==0))
                                    //1% Permanecer em E0
                         {
                              acertos=1;
                              erros=9;
                         }

                        if(((acao>=0)&&(acao<=5))&&(aux3==1))
                              //0.5% Ir para E4
                        {
                         acertos=10;
                         erros=0;
                        }
                  }
            }
            if(((acao>=0)&&(acao<=5))&&((aux1>=2)&&(aux1<=19)))
                  //90% Ir para E2
```

```
{
 acertos=8;
 erros=2;
}
if(((acao>=6)&&(acao<=9))&&((aux1>=0)&&(aux1<=1)))
{
      aux2=rand()%10;
      if(((acao>=6)&&(acao<=9))&&((aux2>=0)&&(aux2<=7)))
            //8% Ir para E3
      {
       acertos=9;
       erros=1;
      }
      if(((acao>=6)&&(acao<=9))&&(aux2==8))
            //1% Ir para E1
      {
       acertos=6;
       erros=4;
      }
      if (aux2==9)
      {
             aux3=rand()%2;
             if(((acao>=6)&&(acao<=9))&&(aux3==0))
                        //1% Permanecer em E0
             {
                  acertos=1;
                  erros=9;
             }

            if(((acao>=6)&&(acao<=9))&&(aux3==1))
                  //0.5% Ir para E4
            {
             acertos=10;
             erros=0;
            }
      }
}
if(((acao>=6)&&(acao<=9))&&((aux1>=2)&&(aux1<=19)))
      //90% Ir para E2
{
 acertos=8;
```

```
            erros=2;
            }
}
if (estado==1)
{
       if(((acao>=0)&&(acao<=5))&&((aux1>=0)&&(aux1<=1)))
            {
                  aux2=rand()%10;
                  if(((acao>=0)&&(acao<=5))&&((aux2>=0)&&(aux2<=7)))
                        //8% Ir para E3
                  {
                   acertos=9;
                   erros=1;
                  }
                  if(((acao>=0)&&(acao<=5))&&(aux2==8))
                        //1% Permanecer em E1
                  {
                   acertos=6;
                   erros=4;
                  }
                  if (aux2==9)
                  {
                         aux3=rand()%2;
                         if(((acao>=0)&&(acao<=5))&&(aux3==0))
                                    //1% Voltar para E0
                         {
                              acertos=1;
                              erros=9;
                         }

                        if(((acao>=0)&&(acao<=5))&&(aux3==1))
                              //0.5% Ir para E4
                        {
                         acertos=10;
                         erros=0;
                        }
                  }
            }
            if(((acao>=0)&&(acao<=5))&&((aux1>=2)&&(aux1<=19)))
                  //90% Ir para E2
            {
```

```
 acertos=8;
 erros=2;
}
if(((acao>=6)&&(acao<=9))&&((aux1>=0)&&(aux1<=1)))
{
      aux2=rand()%10;
      if(((acao>=6)&&(acao<=9))&&((aux2>=0)&&(aux2<=7)))
            //8% Ir para E3
      {
       acertos=9;
       erros=1;
      }
      if(((acao>=6)&&(acao<=9))&&(aux2==8))
            //1% Permanecer em E1
      {
       acertos=6;
       erros=4;
      }
      if (aux2==9)
      {
             aux3=rand()%2;
             if(((acao>=6)&&(acao<=9))&&(aux3==0))
                        //1% Voltar para E0
             {
                  acertos=1;
                  erros=9;
             }

            if(((acao>=6)&&(acao<=9))&&(aux3==1))
                  //0.5% Ir para E4
            {
             acertos=10;
             erros=0;
            }
      }
}
if(((acao>=6)&&(acao<=9))&&((aux1>=2)&&(aux1<=19)))
      //90% Ir para E2
{
 acertos=8;
 erros=2;
```

```
            }
}
if (estado==2)
{
            if(((acao>=0)&&(acao<=5))&&((aux1>=0)&&(aux1<=1)))
            {
                  aux2=rand()%10;
                  if(((acao>=0)&&(acao<=5))&&((aux2>=0)&&(aux2<=7)))
                        //8% Ir para E3
                  {
                   acertos=9;
                   erros=1;
                  }
                  if(((acao>=0)&&(acao<=5))&&(aux2==8))
                        //1% Voltar para E1
                  {
                   acertos=6;
                   erros=4;
                  }
                  if (aux2==9)
                  {
                         aux3=rand()%2;
                         if(((acao>=0)&&(acao<=5))&&(aux3==0))
                                    //1% Voltar para E0
                         {
                              acertos=1;
                              erros=9;
                         }

                        if(((acao>=0)&&(acao<=5))&&(aux3==1))
                              //0.5% Ir para E4
                        {
                         acertos=10;
                         erros=0;
                        }
                  }
            }
            if(((acao>=0)&&(acao<=5))&&((aux1>=2)&&(aux1<=19)))
                  //90% Permanecer em E2
            {
             acertos=8;
```

```
 erros=2;
}
if(((acao>=6)&&(acao<=9))&&((aux1>=0)&&(aux1<=1)))
{
      aux2=rand()%10;
      if(((acao>=6)&&(acao<=9))&&((aux2>=0)&&(aux2<=7)))
            //8% Ir para E3
      {
       acertos=9;
       erros=1;
      }
      if(((acao>=6)&&(acao<=9))&&(aux2==8))
            //1% Voltar para E1
      {
       acertos=6;
       erros=4;
      }
      if (aux2==9)
      {
             aux3=rand()%2;
             if(((acao>=6)&&(acao<=9))&&(aux3==0))
                        //1% Voltar para E0
             {
                  acertos=1;
                  erros=9;
             }

            if(((acao>=6)&&(acao<=9))&&(aux3==1))
                  //0.5% Ir para E4
            {
             acertos=10;
             erros=0;
            }
      }
}
if(((acao>=6)&&(acao<=9))&&((aux1>=2)&&(aux1<=19)))
      //90% Permanecer em E2
{
 acertos=8;
 erros=2;
}
```

```
}
if (estado==3)
{
            if(((acao>=0)&&(acao<=5))&&((aux1>=0)&&(aux1<=1)))
            {
                  aux2=rand()%10;
                  if(((acao>=0)&&(acao<=5))&&((aux2>=0)&&(aux2<=7)))
                        //8% Permanece em E3
                  {
                   acertos=9;
                   erros=1;
                  }
                  if(((acao>=0)&&(acao<=5))&&(aux2==8))
                        //1% Voltar para E1
                  {
                   acertos=6;
                   erros=4;
                  }
                  if (aux2==9)
                  {
                         aux3=rand()%2;
                         if(((acao>=0)&&(acao<=5))&&(aux3==0))
                                    //1% Voltar para E0
                         {
                              acertos=1;
                              erros=9;
                         }

                        if(((acao>=0)&&(acao<=5))&&(aux3==1))
                              //0.5% Ir para E4
                        {
                         acertos=10;
                         erros=0;
                        }
                  }
            }
            if(((acao>=0)&&(acao<=5))&&((aux1>=2)&&(aux1<=19)))
                  //90% Voltar para E2
            {
             acertos=8;
             erros=2;
```

```
            }
            if(((acao>=6)&&(acao<=9))&&((aux1>=0)&&(aux1<=1)))
            {
                  aux2=rand()%10;
                  if(((acao>=6)&&(acao<=9))&&((aux2>=0)&&(aux2<=7)))
                        //8% Permanecer em E3
                  {
                   acertos=9;
                   erros=1;
                  }
                  if(((acao>=6)&&(acao<=9))&&(aux2==8))
                        //1% Voltar para E1
                  {
                   acertos=6;
                   erros=4;
                  }
                  if (aux2==9)
                  {
                         aux3=rand()%2;
                         if(((acao>=6)&&(acao<=9))&&(aux3==0))
                                    //1% Voltar para E0
                         {
                              acertos=1;
                              erros=9;
                         }

                        if(((acao>=6)&&(acao<=9))&&(aux3==1))
                              //0.5% Ir para E4
                        {
                         acertos=10;
                         erros=0;
                        }
                  }
            }
            if(((acao>=6)&&(acao<=9))&&((aux1>=2)&&(aux1<=19)))
                  //90% Voltar para E2
            {
             acertos=8;
             erros=2;
            }
}
```

```
if (estado==4)
{
                  if(((acao>=0)&&(acao<=5))&&((aux1>=0)&&(aux1<=1)))
            {
                  aux2=rand()%10;
                  if(((acao>=0)&&(acao<=5))&&((aux2>=0)&&(aux2<=7)))
                        //8% Vltar para E3
                  {
                   acertos=9;
                   erros=1;
                  }
                  if(((acao>=0)&&(acao<=5))&&(aux2==8))
                        //1% Voltar para E1
                  {
                   acertos=6;
                   erros=4;
                  }
                  if (aux2==9)
                  {
                         aux3=rand()%2;
                         if(((acao>=0)&&(acao<=5))&&(aux3==0))
                                    //1% Voltar para E0
                         {
                              acertos=1;
                              erros=9;
                         }

                        if(((acao>=0)&&(acao<=5))&&(aux3==1))
                              //0.5% Permanecer em E4
                        {
                         acertos=10;
                         erros=0;
                        }
                  }
            }
            if(((acao>=0)&&(acao<=5))&&((aux1>=2)&&(aux1<=19)))
                  //90% Voltar para E2
            {
             acertos=8;
             erros=2;
            }
```

```
            if(((acao>=6)&&(acao<=9))&&((aux1>=0)&&(aux1<=1)))
            {
                  aux2=rand()%10;
                  if(((acao>=6)&&(acao<=9))&&((aux2>=0)&&(aux2<=7)))
                        //8% Voltar para E3
                  {
                   acertos=9;
                   erros=1;
                  }
                  if(((acao>=6)&&(acao<=9))&&(aux2==8))
                        //1% Voltar para E1
                  {
                   acertos=6;
                   erros=4;
                  }
                  if (aux2==9)
                  {
                         aux3=rand()%2;

                         if(((acao>=6)&&(acao<=9))&&(aux3==0))
                                    //1% Voltar para E0
                         {
                              acertos=1;
                              erros=9;
                         }

                        if(((acao>=6)&&(acao<=9))&&(aux3==1))
                              //0.5% Permanecer em E4
                        {
                         acertos=10;
                         erros=0;
                        }
                  }
            }
            if(((acao>=6)&&(acao<=9))&&((aux1>=2)&&(aux1<=19)))
                  //90% Voltar para E2
            {
             acertos=8;
             erros=2;
            }
}
```

Anexo 8– Modelo M3(Excelente) – Não Determinístico

```
/* Modelo3 */
aux1=rand()%20;
if (estado==0)
{
            if(((acao>=0)&&(acao<=5))&&((aux1>=0)&&(aux1<=19)))
                  //100% Ir para E4
            {
             acertos=10;
             erros=0;
            }
            if(((acao>=6)&&(acao<=9))&&((aux1>=0)&&(aux1<=18)))
            //95% Ir para E4
            {
             acertos=10;
             erros=0;
            }
            if (aux1==19)
            {
                        aux2=rand()%5;
                        if(((acao>=6)&&(acao<=9))&&((aux2>=0)&&(aux2<=3)))
                              //4% Ir para E3
                        {
                             acertos=9;
                             erros=1;
                        }
                        if (aux2==4)
                        {
                                        aux3=rand()%5;

      if(((acao>=6)&&(acao<=9))&&((aux3>=0)&&(aux3<=2)))
                                              //0.5% Ir para E2
                                        {
                                         acertos=8;
                                         erros=2;
                                        }

      if(((acao>=6)&&(acao<=9))&&(aux3==3))
                                              //0.25% Ir para em E1
                                        {
                                         acertos=6;
```

```
                                        erros=4;
                                       }

      if(((acao>=6)&&(acao<=9))&&(aux3==4))
                                             //0.25% Permanecer em E0
                                       {
                                        acertos=0;
                                        erros=10;
                                       }
                        }
            }
}
if (estado==1)
{
            if(((acao>=0)&&(acao<=5))&&((aux1>=0)&&(aux1<=19)))
                  //100% Ir para E4
            {
             acertos=10;
             erros=0;
            }
            if(((acao>=6)&&(acao<=9))&&((aux1>=0)&&(aux1<=18)))
            //95% Ir para E4
            {
             acertos=10;
             erros=0;
            }
            if (aux1==19)
            {
                        aux2=rand()%5;
                        if(((acao>=6)&&(acao<=9))&&((aux2>=0)&&(aux2<=3)))
                              //4% Ir para E3
                        {
                             acertos=9;
                             erros=1;
                        }
                        if (aux2==4)
                        {
                                        aux3=rand()%5;

      if(((acao>=6)&&(acao<=9))&&((aux3>=0)&&(aux3<=2)))
                                             //0.5% Ir para E2
```

```
                                        {
                                         acertos=8;
                                         erros=2;
                                        }

      if(((acao>=6)&&(acao<=9))&&(aux3==3))
                                              //0.25% Permanecer em E1
                                        {
                                         acertos=6;
                                         erros=4;
                                        }

      if(((acao>=6)&&(acao<=9))&&(aux3==4))
                                              //0.25% Voltar para E0
                                        {
                                         acertos=0;
                                         erros=10;
                                        }
                        }

            }
}
if (estado==2)
{
            if(((acao>=0)&&(acao<=5))&&((aux1>=0)&&(aux1<=18)))
                  //95% Ir para E4
            {
             acertos=10;
             erros=0;
            }
            if (aux1==19)
            {
                        aux2=rand()%5;
                        if(((acao>=0)&&(acao<=5))&&((aux2>=0)&&(aux2<=3)))
                              //4% Ir para E3
                        {
                             acertos=9;
                             erros=1;
                        }
                        if (aux2==4)
                        {
```

```
                                        aux3=rand()%5;

     if(((acao>=0)&&(acao<=5))&&((aux3>=0)&&(aux3<=2)))
                                              //0.5% Permanecer em E2
                                        {
                                         acertos=8;
                                         erros=2;
                                        }

     if(((acao>=0)&&(acao<=5))&&(aux3==3))
                                              //0.25% Voltar para E1
                                        {
                                         acertos=6;
                                         erros=4;
                                        }

     if(((acao>=5)&&(acao<=5))&&(aux3==4))
                                              //0.25% Voltar para E0
                                        {
                                         acertos=0;
                                         erros=10;
                                        }
                        }
           }
if(((acao>=6)&&(acao<=9))&&((aux1>=0)&&(aux1<=18)))
           //95% Ir para E4
           {
            acertos=10;
            erros=0;
           }
           if (aux1==19)
           {
                        aux2=rand()%5;
                        if(((acao>=6)&&(acao<=9))&&((aux2>=0)&&(aux2<=3)))
                              //4% Ir para E3
                        {
                             acertos=9;
                             erros=1;
                        }
                        if (aux2==4)
                        {
```

```
                                        aux3=rand()%5;

      if(((acao>=6)&&(acao<=9))&&((aux3>=0)&&(aux3<=2)))
                                              //0.5% Permanecer em E2
                                        {
                                         acertos=8;
                                         erros=2;
                                        }

      if(((acao>=6)&&(acao<=9))&&(aux3==3))
                                              //0.25% Voltar para em E1
                                        {
                                         acertos=6;
                                         erros=4;
                                        }

      if(((acao>=6)&&(acao<=9))&&(aux3==4))
                                              //0.25% Voltar para E0
                                        {
                                         acertos=0;
                                         erros=10;
                                        }
                        }

            }
}
if (estado==3)
{
                  if(((acao>=0)&&(acao<=5))&&((aux1>=0)&&(aux1<=18)))
                  //95% Ir para E4
            {
             acertos=10;
             erros=0;
            }
            if (aux1==19)
            {
                        aux2=rand()%5;
                        if(((acao>=0)&&(acao<=5))&&((aux2>=0)&&(aux2<=3)))
                              //4% Permanecer em E3
                        {
                             acertos=9;
```

```
                    erros=1;
                }
                if (aux2==4)
                {
                                aux3=rand()%5;

if(((acao>=0)&&(acao<=5))&&((aux3>=0)&&(aux3<=2)))
                                      //0.5% Voltar para E2
                                {
                                 acertos=8;
                                 erros=2;
                                }

if(((acao>=0)&&(acao<=5))&&(aux3==3))
                                      //0.25% Voltar para E1
                                {
                                 acertos=6;
                                 erros=4;
                                }

if(((acao>=0)&&(acao<=5))&&(aux3==4))
                                      //0.25% Voltar para E0
                                {
                                 acertos=0;
                                 erros=10;
                                }
                }
        }
        if(((acao>=6)&&(acao<=9))&&((aux1>=0)&&(aux1<=17)))
        //95% Ir para E4
         {
              acertos=10;
              erros=0;
         }
         if(((acao>=6)&&(acao<=9))&&(aux1==18))
        //5% Permanecer E3
         {
              acertos=9;
              erros=1;
         }
         if (aux1==19)
```

```
            {
                        aux2=rand()%5;
                        if(((acao>=6)&&(acao<=9))&&((aux2>=0)&&(aux2<=2)))
                              //4% Voltar para E2
                        {
                              acertos=8;
                              erros=2;
                        }
                       if(((acao>=6)&&(acao<=9))&&(aux2==3))
                              //0.5% Voltar para E1
                       {
                        acertos=6;
                        erros=4;
                       }
                       if(((acao>=6)&&(acao<=9))&&(aux2==4))
                              //0.5% Voltar para E0
                       {
                        acertos=0;
                        erros=10;
                       }
            }
}
if (estado==4)
{
            if(((acao>=0)&&(acao<=5))&&((aux1>=0)&&(aux1<=18)))
                  //95% Permanecer em E4
            {
             acertos=10;
             erros=0;

            }
            if (aux1==19)
            {
                        aux2=rand()%5;
                        if(((acao>=0)&&(acao<=5))&&((aux2>=0)&&(aux2<=3)))
                              //4% Voltar para E3
                        {
                              acertos=9;
                              erros=1;
                        }
                        if (aux2==4)
```

```
                    {
                                        aux3=rand()%5;

if(((acao>=0)&&(acao<=5))&&((aux3>=0)&&(aux3<=2)))
                                              //0.5% Voltar para E2
                                        {
                                         acertos=8;
                                         erros=2;
                                        }

if(((acao>=0)&&(acao<=5))&&(aux3==3))
                                              //0.25% Voltar para E1
                                        {
                                         acertos=6;
                                         erros=4;
                                        }

if(((acao>=0)&&(acao<=5))&&(aux3==4))
                                              //0.25% Voltar para E0
                                        {
                                         acertos=0;
                                         erros=10;
                                        }
                    }
      }
       if(((acao>=6)&&(acao<=9))&&((aux1>=0)&&(aux1<=17)))
      //95% Permanecer em E4
       {
            acertos=10;
            erros=0;
       }
       if(((acao>=6)&&(acao<=9))&&(aux1==18))
      //5% Voltar para E3
       {
            acertos=9;
            erros=1;
       }
       if (aux1==19)
       {
                    aux2=rand()%5;
                    if(((acao>=6)&&(acao<=9))&&((aux2>=0)&&(aux2<=2)))
```

```
                    //4% Voltar para E2
                {
                    acertos=8;
                    erros=2;
                }
                if(((acao>=6)&&(acao<=9))&&(aux2==3))
                    //0.5% Voltar para E1
                {
                acertos=6;
                erros=4;
                }
                if(((acao>=6)&&(acao<=9))&&(aux2==4))
                    //0.5% Voltar para E0
                {
                acertos=0;
                erros=10;
                }
            }
        }
```

FOLHA DE REGISTRO DO DOCUMENTO

1. CLASSIFICAÇÃO/TIPO	2. DATA	3. DOCUMENTO N°	4. N° DE PÁGINAS
TM	03 de novembro de 2003	CTA/ITA-IEC/TM-004/2003	162

5. TÍTULO E SUBTÍTULO:
Utilização de aprendizado por reforço para modelagem autônoma do aprendiz em sistemas tutores inteligentes

6. AUTOR(ES):

Marcus Vinicius Carvalho Guelpeli

7. INSTITUIÇÃO(ÕES)/ÓRGÃO(S) INTERNO(S)/DIVISÃO(ÕES):

Instituto Tecnológico de Aeronáutica. Divisão de Ciência da Computação – ITA/IEC

8. PALAVRAS-CHAVE SUGERIDAS PELO AUTOR:

Sistemas tutores inteligentes, Aprendizado por reforço, Algoritmo Q-learning.

9.PALAVRAS-CHAVE RESULTANTES DE INDEXAÇÃO:

Arquitetura (computadores); Algoritmos; Aprendizagem; Inteligência artificial; Programas de sistemas de computadores; Computação.

10. APRESENTAÇÃO: **X Nacional** **Internacional**

ITA, São José dos Campos, 2003, 162p.

11. RESUMO:

Este trabalho apresenta um novo modelo de módulo de diagnóstico para ser incluído na arquitetura tradicional de Sistemas de Tutores Inteligentes. Neste módulo, são aplicadas técnicas de Aprendizado por Reforço (algoritmo Q-Learning), o que possibilita modelar autonomamente o aprendiz. Um valor de utilidade é calculado baseado em uma tabela de pares estado-ação, a partir da qual o algoritmo estima reforços futuros que representam os estados cognitivos do aprendiz. A melhor política a ser usada pelo tutor para qualquer estado cognitivo do aprendiz é disponibilizada pelo próprio algoritmo de Aprendizado por Reforço, sem que seja necessário um modelo explícito do aprendiz..

12. GRAU DE SIGILO:

(X) OSTENSIVO **( ) RESERVADO** **( ) CONFIDENCIAL** **( ) SECRETO**