

HERBERT SCHÄFER

**CONCEITOS E FERRAMENTAS PARA APOIAR O ENSINO DE XADREZ
ATRAVÉS DE COMPUTADORES**

Dissertação apresentada como requisito parcial à obtenção do grau de Mestre. Curso de Pós-Graduação em Informática, Setor de Ciências Exatas, Universidade Federal do Paraná.

Orientador: Prof. Alexandre Ibrahim Direne

**CURITIBA
2000**

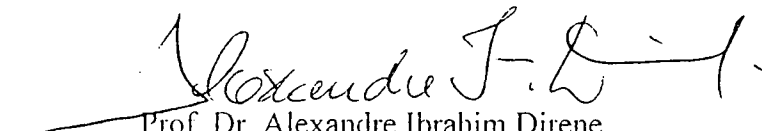


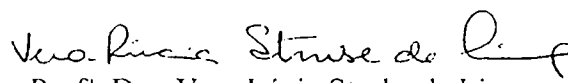
Ministério da Educação
Universidade Federal do Paraná
Mestrado em Informática

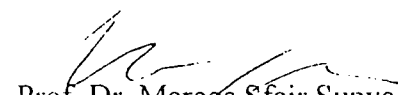
PARECER

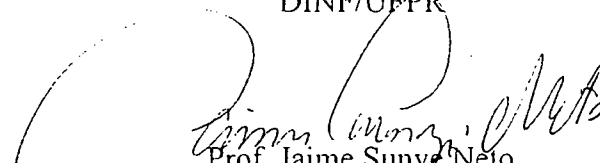
Nós, abaixo assinados, membros da Banca Examinadora da defesa de Dissertação de Mestrado em Informática do aluno *Herbert Schafer*, avaliamos o trabalho intitulado "**Conceitos e Ferramentas para Apoiar o Ensino de Xadrez Através de Computadores**", cuja defesa foi realizada no dia 09 de outubro de 2000. Após a avaliação, decidimos pela aprovação do Candidato.

Curitiba, 09 de outubro de 2000.


Prof. Dr. Alexandre Ibrahim Direne
Presidente - DINF/UFPR


Prof. Dra. Vera Lúcia Strube de Lima
PUC/RS


Prof. Dr. Marcos Sfair Sunye
DINF/UFPR


Prof. Jaime Sunye Neto
CEX-PR

HERBERT SCHÄFER

CONCEITOS E FERRAMENTAS PARA APOIAR O ENSINO DE XADREZ
ATRAVÉS DE COMPUTADORES

Dissertação aprovada como requisito parcial à obtenção do grau de Mestre.

Curso de Pós-Graduação em Informática, Setor de Ciências Exatas,
Universidade Federal do Paraná, pela comissão formada pelos professores:

Orientador: Prof. Dr. Alexandre Ibrahim Direne
Presidente – DINF/UFPR

Prof^a. Dra. Vera Lúcia Strube de Lima
PUC/RS

Prof. Dr. Marcos Sfair Sunye
DINF/UFPR

Prof. Jaime Sunye Neto
CEX-PR

Curitiba, 09 de outubro de 2000

É preciso ter em mente que a tragédia da vida não está em não atingirmos nosso objetivo. A tragédia está em não termos qualquer objetivo pelo qual lutar. Não é uma calamidade morrerem com sonhos a realizar, mas é uma calamidade não sonhar. Não é uma desgraça não atingir as estrelas, mas é uma desgraça não ter estrelas para atingir. Pecado não é fracassar, é ter pouca ambição.

Benjamin E. Mays (*Vantage*)

Dedico este trabalho à minha esposa e a minha filha, pelo amor, carinho e apoio que sempre me deram e em especial ao meu amigo e orientador Alexandre por ter me ensinado primeiro a sonhar, e depois a lutar pelos meus sonhos.

AGRADECIMENTOS

Primeiramente a Deus, que nos momentos certos, interviu poderosamente na minha vida, possibilitando que este dia finalmente chegasse.

À minha esposa Rossana, por ter acreditado e investido em mim, ao me persuadir a fazer o vestibular para o curso de Bacharelado em Informática.

À Coordenação de Pós-Graduação em Informática por ter me aceito no seu quadro de alunos e possibilitado, sempre de maneira excepcional, que eu realizasse este trabalho.

Ao Departamento de Informática da UFPR como um todo e individualmente a cada um dos professores e funcionários deste departamento pelo apoio e pelas condições de trabalho, sempre muito boas. Especialmente ao professor André Pires Guedes, por seu apoio e orientação no meu projeto de graduação, fundamental para a minha opção pelo Mestrado. Ao professor Marcos Sunye, pelas constantes palavras de incentivo. Agradeço ainda aos professores Urban, Olga, Bruno e Renato pelo apoio e ao José Carlos, da Coordenação, pela ajuda e esforço.

Ao Jaime Sunye, pelas palavras de apoio e incentivo.

Ao meu sempre orientador Alexandre, que mais do que orientador, foi um verdadeiro amigo e irmão para mim.

Aos meus amigos.

SUMÁRIO

LISTA DE ILUSTRAÇÕES	viii
LISTA DE TABELAS	viii
LISTA DE ABREVIATURAS	ix
RESUMO	x
ABSTRACT	xi
1 INTRODUÇÃO	1
1.1 O PROBLEMA.....	2
1.2 VISÃO GERAL DO AMBIENTE.....	4
1.3 ESTRUTURA DA DISSERTAÇÃO.....	6
2 TRABALHOS CORRELATOS	7
2.1 ENSINO DE CONCEITOS VISUAIS	7
2.1.1 Princípios e Perícia em Conceitos Visuais	8
2.1.2 Tutores Inteligentes para o Ensino de Conceitos Visuais	11
2.2 LINGUAGENS E FERRAMENTAS DE AUTORIA.....	13
2.3 ENSINO DE ESTRATÉGIAS DE JOGOS E JOGOS EDUCATIVOS.....	17
3 O NÍVEL INSTRUCIONAL	21
3.1 A FERRAMENTA FEREN E SUAS INTERFACES.....	21
3.2 A INTERFACE PASSIVA E O MÉTODO DE DESCOBERTA GUIADA	23
3.3 A INTERFACE ATIVA	26
3.4 PERSPECTIVAS DE APLICAÇÃO.....	29
4 O NÍVEL PRODUÇÃO..	32
4.1 FERRAMENTA DE DESCRIÇÃO DE SITUAÇÕES E ESTUDO DE CASOS	33
4.1.1 Arquitetura básica.....	34
4.1.2 Interfaces da Ferramenta DESCA.....	35
4.2 CLASSIFICADOR DE PGN's	38

4.2.1 A Inadequação de técnicas clássicas de IA	38
4.2.2 Uma abordagem alternativa às técnicas clássicas de IA	42
4.3 CONSTRUTOR DE CASOS PARA ESTUDO	44
4.4 SEQUENCIALIZADOR DE CASOS E ARRANJOS.....	45
5 O NÍVEL CONCEITUAL.....	49
5.1 FERRAMENTA DE DESCRIÇÃO DE CLASSES DE SITUAÇÕES	50
5.1.1 Arquitetura Básica	50
5.1.2 Interface da Ferramenta	52
5.2 CLASSES DE SITUAÇÕES DO DOMÍNIO	53
5.2.1 Composição de uma Classe de Situações.....	54
5.2.2 Anatomia do Tabuleiro	57
5.2.3 A construção das Regras e as Primitivas de Enquadramento	59
5.3 REPRESENTAÇÃO PEDAGÓGICA DAS CLASSES DE SITUAÇÕES.....	61
5.3.1 Regras Pedagógicas	62
6 CONCLUSÃO.....	66
6.1 A CONTRIBUIÇÃO.....	66
6.2 TRABALHOS FUTUROS	67
REFERÊNCIAS BIBLIOGRÁFICAS	69

LISTA DE ILUSTRAÇÕES

1.1 OS SISTEMAS DE ENSINO-APRENDIZAGEM DE ACORDO COM O TIPO DE INTERFACE E TIPO DE AUTORIA	2
1.2 O SISTEMA SAEX	5
3.1 O NÍVEL INSTRUCIONAL.....	22
3.2 A FERRAMENTA FEREN COM SUAS DUAS INTERFACES: A INA À ESQUERDA E A INPA À DIREITA.....	23
3.3 DIAGRAMA DO MECANISMO DE APRESENTAÇÃO DO AMBIENTE DE EXPLORAÇÃO DA INPA	25
3.4 DIAGRAMA DO PROCESSO DINÂMICO DE NAVEGAÇÃO PELA BASE DE DADOS.....	26
3.5 DOIS INSTANTES DA APRESENTAÇÃO DE UM ESTUDO DE CASO PELA INTERFACE INA.....	28
3.6 CLASSIFICAÇÃO DOS MECANISMOS PEDAGÓGICOS PARA A TRANSMISSÃO DE CONHECIMENTO.....	30
4.1 O NÍVEL DE PRODUÇÃO.....	33
4.2 O SAEX PERMITE A EXISTÊNCIA DE DIFERENTES CURSOS EM NÍVEL DE PRODUÇÃO.....	34
4.3 CLASSIFICADOR DE ARRANJOS DE JOGOS EM PGN.....	35
4.4 CONSTRUTOR DE CASOS PARA ESTUDOS.....	36
4.5 SEQUENCIALIZADOR DE CASOS E ARRANJOS.....	37
4.6 O ALGORITMO MINIMAX.....	39
4.7 A ÁRVORE DE PESQUISA CONSTRUÍDA PELO MINIMAX	41
4.8 EXEMPLO DE ARQUIVO TEXTO NO FORMATO PGN.....	43
5.1 O NÍVEL CONCEITUAL	50

5.2 O NÍVEL CONCEITUAL E O PROCESSO DE INTERAÇÃO COM OS DEMAIS NÍVEIS.....	51
5.3 A FERRAMENTA DECLASS.....	53
5.4 UMA DESCRIÇÃO PARCIAL DAS REGRAS DE CLASSIFICAÇÃO DE UMA CLASSE DE SITUAÇÕES.....	54
5.5 A DISPOSIÇÃO INICIAL DO TABULEIRO DE XADREZ.....	58
5.6 O TABULEIRO DE XADREZ E SUAS COORDENADAS.....	58
5.7 EXEMPLO DE INSTRUÇÕES APLICÁVEIS AO OBJETO BOARD.....	63
5.8 EXEMPLO DE REGRA PEDAGÓGICA.....	64
5.9 O RESULTADO DA APLICAÇÃO DA REGRA PEDAGÓGICA DA FIGURA 5.7.....	65

LISTA DE TABELAS

3.1 ALGUNS DOS MAIS COMUNS MÉTODOS DE REFORÇO E COMPLEMENTAÇÃO DO APRENDIZADO.....	24
4.1 PRIMITIVAS DO CONSTRUTOR DE CASOS DE ESTUDO.....	45
5.1 OS SÍMBOLOS UTILIZADOS NO SISTEMA ALGÉBRICO PARA INDICAR A “QUALIDADE DE UMA JOGADA”.....	59
5.2 ALGUMAS DAS PRIMITIVAS DISPONIBILIZADAS PELA FERRAMENTA DECLASS.....	62

LISTA DE ABREVIATURAS E SIGLAS

CAI	<i>COMPUTER AIDED INSTRUCTION</i>
IA	INTELIGÊNCIA ARTIFICIAL
ITS	<i>INTELLIGENT TUTORING SYSTEMS</i>
RUI	<i>REPRESENTATION FOR UNDERSTANDING IMAGES</i>
UFPR	UNIVERSIDADE FEDERAL DO PARANÁ
PGN	<i>PORTABLE GAME NOTATION</i>

RESUMO

Este trabalho apresenta conceitos e ferramentas para gerar sistemas tutores inteligentes (ITS) desenvolvidos especificamente para o ensino de conceitos visuais para o Jogo de Xadrez. Faz-se uma revisão literária sobre o ensino de conceitos visuais através de sistemas computacionais, atingindo o escopo de ambientes de autoria. Nesta revisão, constata-se a inexistência de tutores automáticos que levem em consideração o forte processamento visual primário exigido na formação de perícia para o Jogo de Xadrez. Este texto apresenta um *shell* de ITS para o ensino destes conceitos visuais e discute a aplicabilidade limitada de idéias básicas de algoritmos de busca heurística da IA Clássica. O protótipo de ITS deste trabalho, denominado SAEX, combina ferramentas de autoria com ITS, oferecendo um arcabouço integrado, apropriado para o desenvolvimento dos referidos sistemas de ensino. Ao final, são indicadas possibilidades reais de utilização deste ITS e de trabalhos futuros.

ABSTRACT

This work presents concepts and software tools for generating intelligent tutoring systems (ITS) specifically developed for the teaching of visual concepts aimed at Chess learners. A literature review is presented on computer based teaching of visual concepts as well as on authoring systems and tutoring shells. It is shown that existing systems for teaching Chess do not cover the issues of primary visual interpretation, an important capacity in acquiring expertise according to previous research work. The text presents an ITS shell for the teaching of visual concepts as an alternative to basic ideas of classical AI heuristic search methods. The prototype system SAEX, implemented as part of the work, combines authoring and intelligent systems techniques in an integrated framework to benefit learning. At the end, the text indicates real possibilities of use of this ITS and future research.

CAPÍTULO 1

INTRODUÇÃO

Neste trabalho são formulados conceitos para selecionar e classificar cognitivamente um conjunto de imagens de arranjos de jogos de Xadrez e apresentadas ferramentas de *software* cuja implementação é baseada nestes mesmos conceitos para ordenar e apresentar ao aprendiz este conjunto de imagens. As ferramentas de *software*, desenvolvidas neste trabalho sendo duas de autoria e uma para tutoramento inteligente (ITS), são voltadas para o ensino de “conceitos visuais” e de uso generalizado para o Jogo de Xadrez. O objetivo é apoiar, através do uso de computadores, o ensino de perícia para o Jogo de Xadrez, permitindo assim simular a experiência do professor, a fim de minimizar o tempo de aprendizado e possibilitar a utilização de diversas estratégias pedagógicas.

Os sistemas de ensino por computador, os chamados *Computer Aided Instruction* (CAI), estão sendo cada vez mais usados em muitas áreas do ensino. Sistemas desse tipo tem duas vantagens principais, a saber: (a) facilidade de construção, realizada normalmente através de ferramentas de autoria e (b) a liberdade que o aprendiz tem em percorrer o conteúdo a ser ensinado (liberdade de navegação). Possuem, contudo, uma grande desvantagem: suas reações às ações do aprendiz são fixas, quando existem. Para que um sistema de ensino por computador consiga apresentar reações mais adequadas às ações do aprendiz ele deve ter uma certa “inteligência” (Figura 1.1), ou seja, utilizar técnicas da Inteligência Artificial (IA) tais como inferência baseada em regras, heurísticas e outras. Tais sistemas são chamados de Sistemas Tutores Inteligentes (ITS's). Um sistema tutor inteligente para o ensino ou treinamento no Jogo de Xadrez pode passar o conhecimento pericial através de exemplos (imagens). Um sistema tutor

inteligente que permite que as sessões de ensino sejam criadas ou editadas é chamado de ambiente de autoria. A linguagem usada nessa criação é chamada de linguagem de autoria (DIRENE, 1993).

1.1 O PROBLEMA

O Xadrez é um jogo de estratégia que exige do bom jogador uma forte influência do processamento visual primário (LESGOLD, 1984). Todavia, o ensino de Xadrez por intermédio de computadores até o momento não abordou esta importante faceta da formação de perícia. Os ambientes tutoriais automáticos para o Jogo de Xadrez atuais limitam-se a apresentar exemplos pré-determinados em uma ordem previamente estabelecida, levando em consideração apenas o grau de dificuldade estratégica ou táticas dos exemplos, em detrimento da componente visual dos mesmos. Os ambientes tutoriais automáticos existentes usam uma ordem fixa, pré-estabelecida de exemplos ou medidas de ordenação como tipicidade e similaridade (TVERSKY, 1978), todas fortemente baseadas na freqüência com que um exemplar ocorre na realidade. A esta deficiência soma-se outra característica, comum a este tipo de *software*, que é a impossibilidade de um professor humano acessar e alterar estes exemplos. Os tutores automáticos para o ensino de Xadrez são fechados e não permitem alterações de conteúdo, seqüência ou de controles pedagógicos. Sendo assim, a alteração do material instrucional e pedagógico encontra-se inacessível ao professor. Tais sistemas ignoram totalmente as dimensões de autoria e interpretação, o que impede o seu uso de uma forma mais efetiva (Figura 1.1).

O mesmo ocorre na área de Radiologia Médica. Para esta importante área, onde a perícia é também essencialmente visual, não existiam ITS's que possibilitassem o desenvolvimento de autoria. Um dos ITS's desenvolvidos

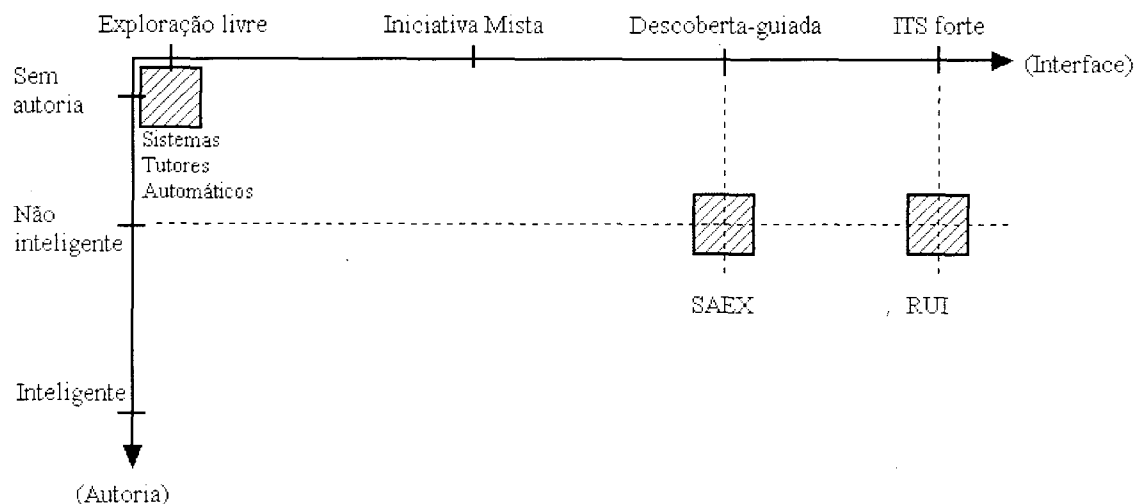


Figura 1.1 – OS SISTEMAS DE ENSINO-APRENDIZAGEM DE ACORDO COM O TIPO DE INTERFACE E TIPO DE AUTORIA

para suprir esta lacuna é o sistema RUI (DIRENE, 1993), próprio para o ensino de conceitos visuais para a área de Radiologia Médica. O sistema RUI (*Representations for Understanding Images*) é um ITS que disponibiliza ferramentas para a autoria (Figura 1.1). Infelizmente não existe até o momento um ITS equivalente para o Jogo de Xadrez.

É necessário desenvolver um ambiente que permita ao mesmo tempo: (a) o aproveitamento da componente visual na formação da perícia no Jogo de Xadrez, e (b) ao professor/autor catalogar exemplos oriundos de jogos “reais” ou construir os seus próprios exemplos por meio da autoria de seqüências e controles pedagógicos de apresentação. Este ambiente deverá ser dotado de uma ferramenta de ensino capaz de manipular os exemplos e de interagir com o aprendiz através de mecanismos pedagógicos previamente estabelecidos, e de ferramentas de autoria baseadas em uma linguagem aberta, que permita ao autor estruturar o conteúdo da forma que julgar mais conveniente.

Este trabalho descreve os conceitos idealizados e utilizados na construção de um ambiente que visa atender a estas especificações. Este ambiente foi denominado SAEX.

1.2 VISÃO GERAL DO AMBIENTE

O ambiente SAEX (Sistema de Apoio ao Ensino de Xadrez) é composto de uma ferramenta genérica (*shell*) de ensino e de duas ferramentas também genéricas para a autoria. Os conceitos e ferramentas descritos neste trabalho são capazes de: (a) permitir o treinamento de aprendizes para avaliar (visualizar/apreciar) e descrever situações de um tabuleiro de Xadrez de forma sistemática e, (b) permitir a autoria de estratégias pedagógicas apropriadas, especialmente quando as situações a serem identificadas podem ser facilmente confundidas com outras cujas táticas de jogo diferem substancialmente.

O ambiente SAEX é dividido em três níveis, sendo os dois primeiros (o Nível Conceitual e o Nível de Produção) voltados para o processo de autoria e o terceiro (o Nível Instrucional) voltado para o ensino de conceitos. Este desmembramento pode ser observado através da Figura 1.2. O processo de autoria foi desmembrado nos Níveis Conceitual e de Produção para permitir que o SAEX seja manipulado por dois tipos distintos de autores: (a) um especialista no domínio e (b) um especialista pedagógico. Enquanto a ferramenta de autoria do Nível Conceitual deve ser utilizada por especialistas no domínio do Jogo de Xadrez, a ferramenta de autoria do Nível de Produção pode ser utilizada também por autores (normalmente professores) com menor especialização no domínio do conhecimento, mas, por sua vez, mais voltados à prática do ensino e à realidade do aprendiz. Já o Nível Instrucional permite que o aprendiz possa assimilar os conceitos editorados pelo especialista através de uma ferramenta composta de: (a) uma interface ativa, na qual o ambiente oferece a instrução formal e, (b) de uma interface passiva, onde o ambiente mediante o método pedagógico da descoberta-guiada (*guided-discovery learning*), permite a livre exploração da base de arranjos e casos de estudo do

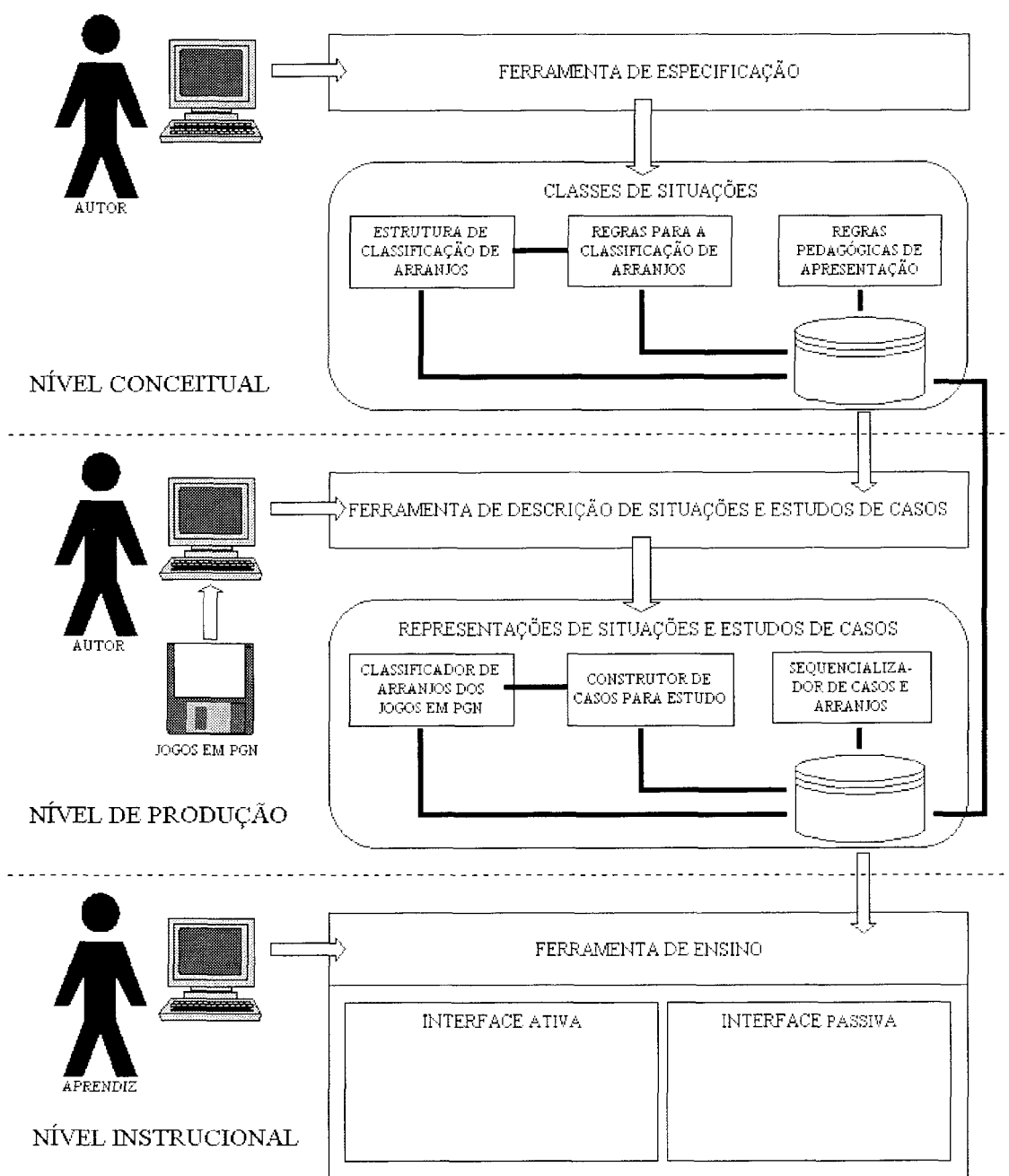


Figura 1.2 – O SISTEMA SAEX

SAEX. Por não possuir um modelo do aprendiz e por possuir um modelo tutorial não desenvolvido, o SAEX não é um exemplo de ITS forte, mas sim um ITS de descoberta guiada, dotado de ferramentas de autoria (Figura 1.1). Estas ferramentas foram totalmente implementadas e, apesar de não totalmente integradas, tornam o SAEX plenamente utilizável.

1.3 ESTRUTURA DA DISSERTAÇÃO

Este trabalho é dividido em seis capítulos. No Capítulo 2 apresenta-se uma revisão sobre o ensino de conceitos visuais, as linguagens e ferramentas voltadas para a autoria em sistemas computacionais e sobre o ensino de estratégias de jogos e jogos educativos.

No Capítulo 3 apresenta-se a ferramenta do Nível Instrucional, a qual através de suas interfaces ativa e passiva propiciará ao aprendiz um ambiente instrucional baseado na descoberta guiada. Neste capítulo também serão apresentadas algumas perspectivas de aplicação do ambiente SAEX.

No Capítulo 4 descreve-se a ferramenta do Nível de Produção, composta de 3 interfaces: (a) o Classificador de PGN's, (b) o Construtor de Casos para Estudo e (c) o Sequencializador de casos e arranjos. Apresenta-se, também, uma análise sobre a não adequação do algoritmo Minimax e a sua substituição, neste tutor, por uma solução que leva em consideração as características visuais dos arranjos de tabuleiro, em detrimento da análise em profundidade do espaço de exploração.

No Capítulo 5 apresenta-se a ferramenta do Nível Conceitual, através da qual o autor cria o ambiente do curso, descrevendo quais serão as classes de situações de tabuleiro a serem estudadas pelo aprendiz. Também será analisada a forma pela qual o autor, ao criar cada uma das classes, irá indicar à ferramenta do Nível de Produção como encontrar nos arquivos PGN os exemplos corretos, e à interface do Nível Instrucional, como apresentar pedagogicamente cada um destes exemplos.

O Capítulo 6 é a conclusão do trabalho, onde apresentam-se as contribuições deste trabalho e algumas linhas sobre trabalhos futuros.

CAPÍTULO 2

TRABALHOS CORRELATOS

O ensino de conceitos visuais é uma área antiga da educação. Reúne desde botânicos, que aprendem a classificar um vegetal olhando para sua folha; geólogos, que são capazes de identificar tipos de minerais a partir de pequenos substratos de rocha; até médicos radiologistas que usam sua visão bem treinada para dar um diagnóstico com base em uma imagem de tomografia computadorizada.

Entretanto, esta perícia não é ensinada em escolas ou universidades, apesar de muitas atividades humanas requererem um forte treinamento para serem executadas com facilidade e precisão. Isto se dá porque atividades que são suficientemente complexas necessitam de muitas horas de prática. Entre estas está o reconhecimento de conceitos visuais. Mesmo um radiologista recém-formado somente adquire perícia depois de alguns anos de trabalho.

Neste capítulo faz-se uma resenha crítica das principais pesquisas relacionadas com este trabalho. Na Seção 2.1 apresentam-se algumas definições básicas sobre o ensino de conceitos visuais, princípios, perícias e tutores inteligentes para conceitos visuais. As linguagens e ferramentas de autoria, e as *shell's* inteligentes serão tratadas na Seção 2.2. Finalmente na Seção 2.3 descrevem-se alguns ambientes para o ensino de estratégias de jogo e de jogos educativos.

2.1 ENSINO DE CONCEITOS VISUAIS

De acordo com Howard (HOWARD, 1987), um conceito é definido como uma representação mental de uma categoria. Uma categoria pode ser

definida como uma classe onde um objeto é nela classificado de acordo com alguma noção de semelhança (HOWARD, 1987). Por exemplo, a categoria das árvores engloba todos os objetos, reais ou imaginários, que podem ser classificados como uma árvore. Por outro lado, o conceito que uma pessoa possui de uma árvore é uma idéia. Ou seja, é apenas uma representação mental, que permite a essa pessoa classificar ou não um objeto como árvore.

Segundo Sharples (SHARPLES, 1991), um conceito visual é uma estrutura mental, com nome, associada a um conjunto de imagens visuais. São exemplos de conceitos visuais: transparente, um padrão têxtil e galáxia espiral.

A diferença básica entre o ensino de conceitos e o de conceitos visuais é que no caso dos conceitos visuais o aprendiz já conhece muito bem a definição dos princípios do domínio visual e só precisa adquirir perícia por meio de classificação das imagens, combinando estes princípios e imagens em esquemas a serem manipulados pelo perito.

2.1.1 Princípios e Perícia em Conceitos Visuais

O processo de aquisição de princípios por parte do aprendiz é essencialmente interno e pessoal. Um aprendiz que está adquirindo princípios recorre a múltiplos processos mentais: associa, compara, avalia, interpreta, classifica, estabelece hipóteses, compreende, imagina, julga, reflete, memoriza, reconhece, recorda (PFROMM NETTO, 1987). Em Radiologia Médica, que é a área que mais estudos tem suscitado no campo de sistemas tutores inteligentes, o corpo de conhecimento de princípios é composto de anatomia, fisiologia, teorias médicas de doenças e geometria projetiva de Radiografia (LESGOLD, 1984).

Já a aquisição de perícia em conceitos visuais consiste basicamente em desenvolver no aprendiz a capacidade de classificar imagens, identificar características e descrever anomalias, combinando estes conhecimentos em

esquemas a serem manipulados pelo perito como se fossem uma única unidade. Por exemplo, em conceitos de Radiologia Médica, a habilidade de classificar imagens, de identificar características visuais (ocultas ou não) e de descrever anomalias constitui uma parte crítica do treinamento de radiologistas (SHARPLES et al, 1997).

Isto ocorre porque a interpretação de radiografias é uma tarefa particularmente difícil, pois a imagem é uma abstração degradada da estrutura física (as estruturas 3D são reduzidas a modulações de intensidade de imagem) e a qualidade da imagem é influenciada através de fatores técnicos como tempo de exposição. Ainda assim um radiologista especialista utiliza tipicamente apenas de 20 a 30 segundos para estudar uma imagem, e Lesgold (LESGOLD et al., 1984) demonstrou que radiologistas peritos podem fazer interpretações precisas depois de examinar uma imagem por somente 2 segundos. Esta perícia advém do desenvolvimento de esquemas mentais, através da automatização do processo de interpretação visual mediante a visualização de um número muito grande de imagens (SHARPLES, 1991).

O desenvolvimento destes esquemas mentais também exige do perito a utilização de um vocabulário técnico extremamente preciso. De fato, Lesgold (SHARPLES, 1991) observou que quando um perito necessita explicar e justificar uma interpretação visual, ele o faz utilizando-se de um vocabulário técnico preciso, que também foi desenvolvido junto com a sua perícia visual.

Estudos realizados por William Chase e Herbert Simon junto a peritos enxadristas demonstraram que um mestre enxadrista pode reconhecer e explicar de 10.000 a 100.000 posições de tabuleiros significativos (LESGOLD, 1984). O perito pode reconhecer cada uma destas situações e fornecer uma resposta específica associada para cada uma delas. Entretanto, isto não significa que o mestre enxadrista seja um mero catalogador de esquemas. Pesquisas sugerem que deve ser aprendida uma larga variedade de

conhecimentos específicos para um humano se tornar um perito (LESGOLD, 1984).

Os estudos sobre diferenças entre iniciantes e especialistas sugerem que um aprendiz passa por diversos estágios durante a aquisição de conhecimento e competência. O conhecimento é adquirido em estágios, onde o primeiro se constitui na aquisição de conhecimento declarativo (ou factual), o segundo na sua compilação e o último estágio no seu uso na forma procedimental (ANDERSON, 1983). Enxadristas iniciantes aprendem a associar características visuais com regiões de um tabuleiro e usam esta associação para guiar a classificação da imagem do tabuleiro como membro de uma classe de situações (por exemplo, um ataque ao “Rei” que resulte em perigo de perda da “Dama”). Na medida em que se tornam mais experientes, estes enxadristas desenvolvem uma abordagem mais sistemática para a análise de jogadas, baseando-se em métodos hipotético-dedutivos. Tais procedimentos podem ser resumidos em três passos: (1) uma passada-de-olhos rápida sobre o tabuleiro à procura de características comuns que indique um arranjo familiar de peças, (2) formar uma hipótese inicial a respeito da classe de situações a que o arranjo pertence e, (3) buscar sistematicamente novas características, mesmo não-visuais, que venham a reforçar ou desconfirmar a hipótese adotada (LESGOLD et al, 1989).

É certo que mestres em Xadrez também são capazes de usar a abordagem hipotético-dedutiva quando necessário (para criar um plano de ação ou para justificar uma jogada), mas eles desenvolvem com o tempo uma enorme habilidade para a rápida comparação de padrões visuais que se baseia em um volumoso estoque de “esquemas” mentais. Ao contrário dos iniciantes, os mestres usam esquemas mentais estruturais fortemente associados aos esquemas puramente visuais, e têm sólidas representações mentais da estrutura do tabuleiro em análise.

Os estudos sobre diferenças entre iniciantes e mestres em Xadrez indicam também que enxadristas de nível intermediário de competência necessitam de assistência na identificação das características críticas de uma jogada, na formação de uma hipótese a respeito da classe de situações encontrada, e na indicação de evidências para confirmação da jogada pretendida. Isto ocorre porque o processo de aquisição de perícia não se desenvolve de forma linear. Lesgold (LESGOLD et al., 1989) descreve as experiências feitas com radiologistas peritos e residentes para descobrir quais são as diferenças nos processos mentais de ambos durante o diagnóstico. Em casos um pouco mais complicados, freqüentemente os residentes iniciantes chegam mais perto do diagnóstico correto que os residentes intermediários.

Estas observações tem sua explicação no fato de que os residentes iniciantes possuem esquemas mais simples para os problemas enquanto que os especialistas possuem esquemas bem mais complexos. Os residentes avançados ainda possuem os esquemas antigos mas estão adquirindo os novos e ainda não tiveram tempo de mapear corretamente os esquemas novos sobre os antigos e mais simples. Isto nos leva à teoria da *Curva U* do aprendizado e *Pandemonium* de Selfridge (SELFRIDGE, 1959). Através desta teoria descobre-se que a distribuição da aquisição das características cognitivas por parte de um humano, no tempo, não é uniforme. Determinadas características são desenvolvidas no começo do aprendizado, outras, no final do aprendizado, e outras durante todo o período (PIMENTEL, 1997).

2.1.2 Tutores Inteligentes para o Ensino de Conceitos Visuais

O ensino de conceitos visuais assistido por computadores ainda se encontra em estágios primários de desenvolvimento (SHARPLES, 1991). Pesquisadores da área de Inteligência Artificial têm feito esforços para produzir tutores que se baseiam em projeto de sistemas apoiado por “modelos

cognitivos” (SHARPLES, 1991), onde os requisitos do sistema computacional são derivados de uma fase anterior de análise fundamentada em teorias de psicologia da cognição. Tais teorias são provenientes da literatura sobre o ensino de conceitos em geral, combinada com a teoria de “protótipos” para a formação de tais conceitos (MERVIS e ROSCH, 1981). Além disso, também são considerados estudos a respeito das diferenças fundamentais entre alunos iniciantes, de nível intermediário e especialistas na interpretação tanto de imagens médicas quanto de posicionamento de peças em um tabuleiro de Xadrez (LESGOLD et al, 1989).

Trabalhos mais recentes foram desenvolvidos na área de Ferramentas Automatizadas para o projeto de Sistemas Tutores Inteligentes (ITS's), aplicados ao ensino de conceitos visuais apoiado por computador (DIRENE, 1997). Na medida do possível, a incorporação de resultados de pesquisa básica na área de imagens médicas (SHARPLES et al, 1997, TEATHER, 1988) tem sido tentada como forma de cobrir o grande espaço existente entre campos interdisciplinares de desenvolvimento. Tal cobertura pode ser atingida com base em semelhanças conceituais entre, por exemplo, o processo de ensino de Radiologia e de Xadrez (LESGOLD et al, 1989).

Para a construção de um tutor automático é necessário que sejam disponibilizados previamente todo o conhecimento específico e características pedagógicas que serão utilizados. Ao contrário do trabalho apoiado por um livro, o valor educacional de um tutor automático está no fato deste ser capaz de apresentar o material de ensino de forma altamente dinâmica, com base em uma gama de casos típicos e atípicos onde a variação dependerá da reação de cada aluno. A eficácia de um tutor automático depende de um acervo de classes de situações (e estratégias associadas). Um tutor automático deve incorporar um subconjunto estratégico destas situações para permitir que o

aprendiz possa “inspecionar” este espaço de situações representadas com suas estratégias e praticar simulações baseadas em estudo de caso.

No processo de desenvolvimento de um tutor para a interpretação de mamografias denominado *RadTutor*, Azevedo (AZEVEDO et al, 1997) apresenta críticas a vários sistemas tutores direcionados a radiologistas, destacando a necessidade de construir sistemas baseados em fundamentos cognitivos como o de Sharples (SHARPLES et al, 1997). O *RadTutor* inclui uma base teórica oriunda de observações da prática do mundo real que enfoca fatores pedagógicos implícitos em sessões de treinamento de residentes em hospitais, produzindo assim um protótipo de um tutor para mamografia apoiado em uma base teórica e empírica consistente.

Estas características tornam a construção de tutores automáticos cara e difícil, especialmente por incluir diversos aspectos interdisciplinares, especialmente os de caráter pedagógico. Se faz necessário portanto construir sistemas de autoria que visem a redução destes custos, assim como a produção automatizada de material de curso para um grande grupo de autores (MURRAY, 1999). Isto é possível pois a maioria dos sistemas de autoria oferece *shell's* que permitem o reuso do mecanismo pedagógico essencial que contém componentes instrucionais ajustáveis em vários níveis de complexidade. A análise destas *shell's*, suas linguagens e ferramentas de autoria será apresentada no item 2.2

2.2 LINGUAGENS E FERRAMENTAS DE AUTORIA

Através de entrevistas com professores, Major e Reichgelt (MAJOR, REICHGELT, 1991) identificaram que uma das razões pelas quais os sistemas tutores inteligentes não são amplamente utilizados em sala de aula é porque a

maioria dos sistemas ensinam de acordo com uma estratégia fixa de ensino. Isso significa que eles não permitem que os professores alterem o modo como o conteúdo é ensinado. Os ambientes integrados de autoria e de distribuição de material didático interativo surgiram para permitir que os professores determinem como os conceitos serão ensinados aos alunos através de um STI (NASCIMENTO, 2000).

Por ambiente integrado de autoria e distribuição de material didático interativo considera-se o conjunto formado por linguagens e ferramentas de autoria e pela ferramenta de ensino/aprendizagem. As arquiteturas dos ambientes de autoria em geral são semelhantes tanto em ambientes inteligentes como nos não inteligentes, tais como os ambientes de exploração e micromundos. Os ambientes de autoria são uma importante família na área do ensino baseado em computador. A grande diferença em relação aos sistemas tradicionais de ensino por computador é que nestes, o conhecimento sobre o domínio e estratégias de ensino são colocados de maneira definitiva no sistema, sendo necessária uma reconstrução do sistema para alterar o conhecimento. Por outro lado, num ambiente de autoria, esse conhecimento é colocado de maneira a permitir que seja alterado, retirado, trocado ou incrementado à vontade pelo autor. Isto constitui a autoria, sem a necessidade de reconstrução do sistema, mas usando ferramentas do sistema, as chamadas ferramentas de autoria, próprias para facilitar esta tarefa.

A arquitetura de um ambiente de autoria é dividida em duas partes principais: modo de autoria e modo de ensino. O modo de ensino, composto pela *shell* de ensino, é a parte do ambiente que interage com o aprendiz para a comunicação do conhecimento. O modo de ensino é um ITS, o qual executa as mesmas funções de um ITS tradicional e tem as mesmas responsabilidades. Quanto ao modo de autoria, um dos trabalhos pioneiros no campo foi o de O'Shea (O'SHEA, 1984), o qual já em 1984 dividia os componentes funcionais

da produção de um sistema tutor inteligente em vários sub-componentes. Por exemplo, O'Shea dividia a produção do modelo pedagógico em: (a) regras de longo prazo (estratégias), (b) regras de curto prazo (táticas) e (c) um gerador de diretivas pedagógicas de curto e longo prazo.

Atualmente existem vários modelos e ferramentas para a construção de tutores inteligentes publicados, nos quais a própria facilidade de criação de material de curso se encarrega de esconder detalhes funcionais que não necessitam ser explicitamente descritos pelos autores (SHARPLES e DU BOULAY, 1988; MAJOR e REICHGELT, 1991; MURRAY e WOOLF, 1992; DIRENE, 1993).

Entre eles podemos destacar o ASIMOV (BINDER e DIRENE, 1999), construído para apoiar o ensino de programação de linguagens imperativas/procedimentais. Esta *shell* de tutor inteligente permite através de autoria, a construção de diferentes soluções de um mesmo problema de programação para que o interpretador pedagógico as manipule e interaja com o estudante de acordo com a preferência do autor. Nele, tanto a linguagem escolhida como linguagem nativa do sistema pode ser adaptada pelo professor, quanto o professor pode ainda construir o seu próprio conjunto de exercícios.

No sistema RUI (DIRENE, 1993), o modo de autoria é a parte do sistema que permite ao autor do ITS expandir ou alterar o conhecimento do modelo do domínio do sistema, ou seja, a edição do conhecimento que será transmitido ao aprendiz (vide capítulo 1).

De uma forma mais restrita, o sistema SATELIT (NASCIMENTO, 2000) é um tutor desenvolvido para apoiar o operador de centrais digitais de comutação telefônica na aquisição de perícia de programação. Ele utiliza-se de modelos internos prévios da solução dos problemas propostos ao operador, criados pelo próprio autor, através de uma meta-linguagem.

Já o Demonstr8 (BLESSING, 1997), visa reduzir a necessidade do autor possuir conhecimento “não-específico” de domínio, através de programação por demonstração, tornando o processo de autoria muito mais simples e rápido. Nele, o autor apenas “demonstra” o conteúdo que deseja explorar, através de exemplos e o próprio sistema tutor identifica a partir de sua base de conhecimento quais serão os conteúdos a serem desenvolvidos com o aluno durante a sessão de ensino-aprendizado.

Diferentemente dos sistemas anteriores, os ambientes de ensino-aprendizagem SIATE (FREIRE, 1997), SASHE (Santos, 1997) e HIPERGRAFE (BEILER, 1996) utilizam-se de recursos de hipertexto para atingir seus objetivos. Segundo Coney, hipertextos são sistemas gerenciadores de informações nos quais estas são armazenadas em uma rede de nós, conectados através de ligações (BEILER, 1996). Quando o hipertexto inclui, além do texto, outras mídias tais como sons, imagens, desenhos e animações, este recebe a denominação de “hipermídia” (BEILER, 1996). Neste trabalho os termos hipertexto e hipermídia serão tratados como sinônimos.

O sistema SIATE propõe uma arquitetura que conjuga técnicas de simulação, de sistemas tutores inteligentes e de hipertexto que permite ao aprendiz: (a) interagir com uma simulação de uma situação real, (b) questionar sobre o que ele observa na simulação, (b) eliminar dúvidas conceituais sobre o domínio e (c) analisar casos análogos, que esclareçam os processos ocorridos na simulação. Já o sistema SASHE é um sistema de autoria que visa auxiliar o autor no processo de edição de conteúdos para um ambiente de hipertexto. Garante-se assim a inclusão do que for considerado pelo autor como essencial ou relevante, sem eliminar o secundário ou complementar, desde que este último não conflite com o essencial e efetivamente contribua com os objetivos do autor e do aprendiz.

Por fim o sistema HIPERGRAFE (BEILER, 1996) implementa uma forma distinta de ensino-aprendizagem, no qual o aprendiz torna-se co-autor do processo de autoria. Através de um ambiente de hipertexto, o aprendiz tem a oportunidade de tornar-se também um produtor de conhecimento. Esta característica representa uma mudança radical na comunicação autor-aprendiz, pois o aprendiz pode, ele mesmo, seguir desenvolvendo o processo de autoria (criar suas anotações no documento, criar novas ligações entre diferentes documentos até então não previstas, modificar o material existente ou até criar e incluir novos materiais).

2.3 ENSINO DE ESTRATÉGIAS DE JOGOS E JOGOS EDUCATIVOS

Independentemente se o jogo é individual - em que o objetivo é, por exemplo, montar um “quebra-cabeças” – ou se é um jogo para dois ou mais indivíduos, três características estão sempre presentes: (a) todo jogo apresenta um objetivo ou situação-problema, (b) um resultado em função deste objetivo e (c) um conjunto de regras determinando os limites dentro dos quais os aspectos (a) e (b) serão considerados. Em um jogo, o desafio ao jogador é encontrar ou produzir os meios que, dentro desses limites, o levarão à vitória, ou seja, a um resultado favorável (MACEDO, 1995).

Nesta busca ou produção dos meios necessários para atingir um resultado favorável, dois desafios estão sempre presentes, mesmo que ignorados pelo jogador: (a) considerar, em cada jogada, diferentes possibilidades e (b) eliminar aquelas possibilidades que prejudicam os objetivos do jogo. São dois desafios diferentes: no primeiro o jogador deve estar atento aos muitos modos de combinar entre si os elementos do jogo (os princípios), e no segundo, escolher aqueles que se configuram como os melhores (a perícia).

Compreender ou dominar um jogo supõe poder considerar – ao mesmo tempo e o melhor possível – estes dois requisitos.

Devido a estas características, os jogos pertencem a uma das mais antigas áreas de pesquisa em inteligência artificial. E dentro destes destacam-se os jogos que lançam mão da figura do “oponente”. Isto porque a presença do oponente torna mais complexo o problema de decidir qual a melhor jogada do que simplesmente percorrer o espaço de busca do problema. O oponente introduz incertezas, porque nunca se sabe o que ele vai ou não fazer. Mas o que torna os jogos efetivamente atraentes sob o ponto de vista da inteligência artificial é que eles normalmente apresentam um espaço muito grande de soluções. O exemplo mais típico é sem dúvida alguma o Jogo de Xadrez.

Até o momento, somente sistemas tutores não inteligentes estão disponíveis para o domínio específico do Jogo de Xadrez (USCF Chess, Série Chessmaster e TASK, entre outros). Enxadristas têm adquirido habilidade, em grande parte, por intermédio de auto-orientação e de tentativas um tanto quanto “desordenadas” ao jogarem com outros durante a fase de aprendizagem. Os sistemas tutores de conceitos visuais podem reduzir de alguma forma o tempo de treinamento por intermédio da exposição do aluno a um conjunto de imagens tutoriais bem planejadas e sob condições de ensino “controladas” (ao invés de “desordenadas”) (SHARPLES, 1991). Porém, nenhuma tentativa de construir tutores inteligentes para o jogo de Xadrez foi feita até o momento.

Além de tudo, a grande maioria dos sistemas automáticos desenvolvidos até o momento para o Jogo de Xadrez assumem que o jogador humano é sempre um especialista. Em outras palavras, estes sistemas não são capazes de identificar deficiências em um aprendiz para que, ao invés de atacá-lo, possam orientá-lo no sentido de reconhecer situações diversas e ensiná-lo a reagir apropriadamente. Os poucos protótipos de sistemas tutores

inteligentes conhecidos para jogos em geral (LABAT, 1993) são versões reduzidas que ainda carecem de fundamentos da própria área de estratégias pedagógicas a serem aplicadas aos jogos. Esta meta ainda requer pesquisas a respeito de formas mais precisas para a representação de jogadas e de extensos catálogos de posições construídas e testadas em ambientes realistas da prática dos jogos.

O WEST (BURTON e BROWN, 1976) é um tutor que se presta a treinar o aprendiz na manipulação das operações matemáticas básicas através de um jogo. Ao apresentar 3 (três) números escolhidos aleatoriamente, o sistema solicita ao aprendiz que, utilizando-se de duas operações matemáticas distintas, gere um quarto valor, que indicará o número de “casas” a serem percorridas em um tabuleiro. O jogo foi projetado de tal forma que, obter o número maior nem sempre é a melhor estratégia devido à presença de atalhos e desvios no tabuleiro. Em alguns momentos a melhor estratégia é conseguir os maiores valores, e em outros, os menores. Assim, os jogadores são encorajados a explorar diferentes modos de combinar os números com os operadores aritméticos.

Outro tutor que utiliza o conceito de jogos educativos para a transmissão de perícia é o QUEST, desenvolvido por Barbara White e John Frederiksen (WHITE e FREDERIKSEN, 1985). Nele, o conhecimento sobre o dispositivo – um circuito elétrico – é representado internamente na forma de um modelo qualitativo. Ao invés de ter acesso a circuitos elétricos simulados, o estudante tem acesso ao próprio modelo conceitual da eletricidade. No QUEST as simulações gráficas e explicações casuais sobre o comportamento de circuito elétrico têm um papel proeminente devido à ênfase colocada em apoiar o desenvolvimento no estudante de modelos de circuitos elétricos em funcionamento. A meta deste tutor é fazer o estudante entender os princípios gerais que governam o comportamento dos circuitos elétricos para poder

predizer os estados em que se encontram os diferentes componentes e assim poder executar um conjunto pequeno de operações para a identificação de problemas. Para este fim, o ambiente é flexível o bastante para permitir ao estudante construir e modificar circuitos enquanto ele ainda está recebendo explicações sobre os estados dos diversos componente e demonstrações sobre como identificar problemas.

Na área de ensino de estratégias de jogos, através de sistemas tutores inteligentes, o único exemplo publicado que se tem notícia é o QUIZ (LABAT, 1991, 1992, 1993). Apesar da estrutura principal do seu modelo pedagógico apoiar-se no conceito de planos, QUIZ é um programa que baseia o seu funcionamento em um mecanismo de agendas. A perícia a ser ensinada está incorporada nas regras de produção que permeiam toda a estrutura do domínio e o modelo de estudante, formando o que se denomina de currículo. Este currículo é composto de uma lista ordenada de temas (a agenda) a serem abordados, durante as sessões de ensino. Estes temas são divididos em módulos e estes em sub-tarefas, criando assim uma estrutura hierárquica para a manipulação dos objetivos instrucionais. Esta estrutura permite que a elaboração e execução do plano instrucional ocorra de forma intercalada, uma vez que os planos são construídos a partir de sub-planos já previamente existentes. Outra característica importante é que esta estrutura separa explicitamente o conhecimento pedagógico (tarefas, planos, regras e meta-regras) dos mecanismos de inferência.

Apesar de possuírem cada qual, características interessantes, nenhum destes 3 (três) tutores presta-se ao ensino do Jogo de Xadrez. O tutor QUIZ, é sem dúvida alguma o que mais se aproxima, por prestar-se ao ensino de estratégias de jogos. Contudo, nem ele nem os demais são capazes de abordar a componente visual necessária para a formação de perícia necessária para este jogo.

CAPÍTULO 3

O NÍVEL INSTRUCIONAL

O ambiente SAEX é dividido em três níveis, sendo os dois primeiros (o Nível Conceitual e o Nível de Produção) voltados para o processo de autoria e o terceiro (o Nível Instrucional) voltado para o ensino de conceitos. Este desmembramento pode ser observado através da Figura 1.2.

Este capítulo descreve como o aprendiz, utilizando-se da ferramenta do Nível Instrucional, pode assimilar os conceitos editorados pelo especialista através: (a) da interface ativa, na qual o ambiente oferece a instrução formal e (b) da interface passiva, onde o ambiente mediante o método pedagógico da descoberta-guiada (*guided-discovery learning*), permite a livre exploração da base de arranjos e casos de estudo do SAEX.

Na Seção 3.1 analisa-se os dois tipos de interface e o método pedagógico da descoberta-guiada, utilizados pela ferramenta de ensino (FEREN). As Seções 3.2 e 3.3 descrevem as suas funcionalidades. A Seção 3.4 analisa as perspectivas de aplicação do SAEX.

3.1 A FERRAMENTA FEREN E SUAS INTERFACES

A ferramenta de Ensino (FEREN) apresenta duas interfaces distintas: (a) a ativa (INA) e (b) a passiva (INPA), tal qual apresentado na Figura 3.1. Toda a interação entre o aprendiz e o sistema tutor ocorre através delas. A utilização de uma ferramenta provida de duas interfaces distintas permite integrar a exploração guiada com mecanismos tutoriais de ensino.

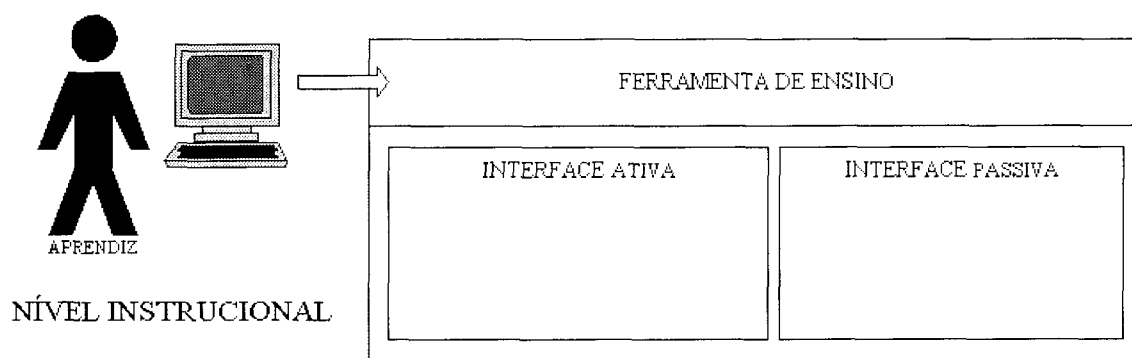


Figura 3.1 – O NÍVEL INSTRUCIONAL

Espera-se através desta ferramenta reduzir a distância entre os dois tipos de conhecimento descritos por Lesgold: (a) o conhecimento dos princípios e (b) o conhecimento pericial. A aquisição dos princípios deverá ocorrer pela interação do aprendiz com um *browser* de banco de dados de imagem (uma interface passiva). Já a aquisição do conhecimento pericial provém da interação do aprendiz com uma interface ativa. A Figura 3.2 apresenta um instantâneo da ferramenta de ensino, onde a interface ativa (INA) é exibida à esquerda, e a interface passiva (INPA) à direita. A INA destina-se à comunicação de conhecimento pericial, enquanto que a INPA presta-se à comunicação de conhecimento de princípios.

Usando a INPA, a interface passiva, o aprendiz procura e seleciona arranjos de jogadas e casos de estudo para adquirir o conhecimento subjacente de princípios, necessário para classificar estes mesmos arranjos e casos. Usando a INA, a interface ativa, o aprendiz desenvolve o conhecimento pericial ao tentar resolver os problemas e identificar as características visuais que a interface lhe apresenta. É através desta interface que o sistema constantemente desafia o aprendiz a fazer uso dos princípios conceituais adquiridos para formar diagnósticos mais precisos e completos.

Cada uma das duas interfaces será analisada com maior detalhamento nas seções seguintes.

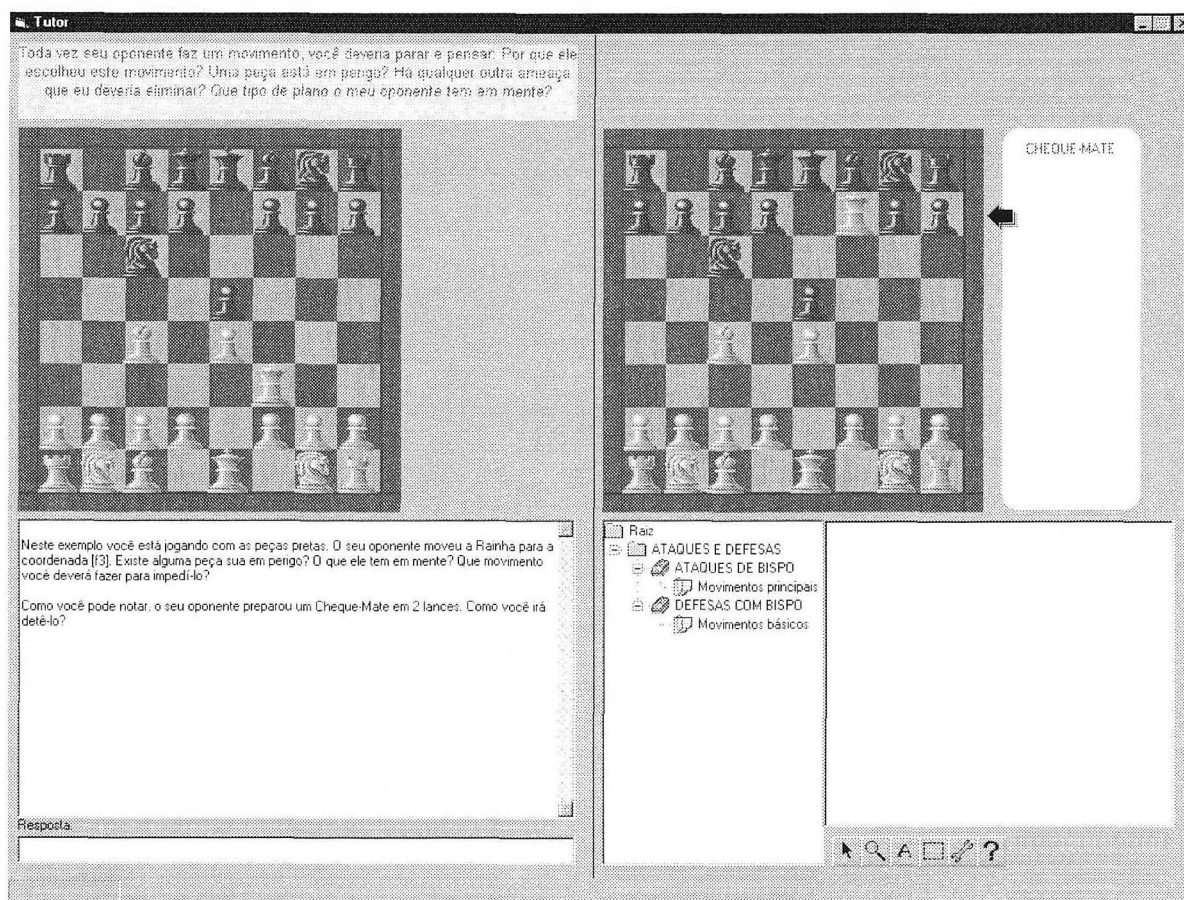


Figura 3.2 – A FERRAMENTA FEREN COM SUAS DUAS INTERFACES: A INA À ESQUERDA E A INPA À DIREITA

3.2 A INTERFACE PASSIVA E O MÉTODO DE DESCOBERTA GUIADA

A descoberta guiada é um método pedagógico comumente utilizado para reforçar e generalizar as habilidades e conteúdos assimilados pelo aprendiz (Tabela 3.1). Por permitir a exploração do ambiente de forma contextualizada e dosada, é um método particularmente útil quando utilizado como preâmbulo ou complementação de um processo de exercitação das habilidades e/ou conteúdos adquiridos. Este método de exploração procura manter sempre um equilíbrio entre dirigir a exploração ou permitir que ela ocorra livremente, levando em consideração, de um lado o estado do aprendiz, e do outro a complexidade do assunto a ensinar.

A Interface INPA permite, através do método de descoberta guiada, que o aprendiz examine a base de dados do SAEX de uma forma mais adequada ao seu aprendizado do que a simples exploração ao acaso. Subdivididos em diferentes classes de situações (vide Capítulo 5) e ordenados através da suas respectivas cargas cognitivas (Capítulo 4), os arranjos de tabuleiro e casos para estudo são listados e apresentados pela INPA obedecendo aos critérios de pertinência e de tipicidade.

Quando o aprendiz explora a base de dados do sistema, a interface lista os exemplos, levando em consideração qual é a classe de situações atualmente em estudo (pertinência) e ordenando os exemplos de tal forma que

Descoberta imprevista	Aprendizagem não planejada, nenhuma instrução é envolvida diretamente.
Descoberta por livre exploração	Abordagem de Bruner (BRUNER, 1968). Os macro objetivos são fixados, e os alunos ficam livres para explorar métodos, sub-objetivos, etc.
Descoberta guiada	Abordagem de Gagné (GAGNÉ, 1985). Os objetivos de cada passo da aprendizagem são fixados. O aprendiz é livre para explorar métodos, mas com guia e ajuda em cada estágio.
Descoberta adaptativa	Abordagem cibernética, guia e reforço corretivo com estrutura de diagnóstico.
Descoberta linear/intrínseca	Direcionada rigidamente, guia e reforço são pré programados, baseados no estudante típico.
Exposição indutiva	o aluno recebe o argumento, não tendo que descobrir a regra correspondente.
Exposição dedutiva	a compreensão do problema é mostrada pela habilidade de aplicá-lo a exemplos. Abordagem baseada em Ausubel (AUSUBEL, 1968)
Aprendizagem de Recepção direcionada (exercício e prática)	Aprendizagem de fatos, sentenças e operações sem entender os conceitos envolvidos. Memorização
Aprendizagem de Recepção imprevista	Fatos e observações, originalmente não planejados, fornecidos por professores, outras fontes e estudantes.

Tabela 3.1 – ALGUNS DOS MAIS COMUNS MÉTODOS DE REFORÇO E COMPLEMENTAÇÃO DO APRENDIZADO

o mais comum, ou o mais simples, ou o que melhor apresente uma determinada característica definida pelo autor seja listado por primeiro e assim sucessivamente (tipicidade) (Figura 3.3). Isto não impede que o aprendiz explore livremente a base de dados, mas o induz a um direcionamento específico. Se o aprendiz selecionar para visualização um determinado exemplo (arranjo ou estudo) de uma classe totalmente diferente daquela em estudo no momento, a interface a apresentará, mas não sem antes questioná-lo de que o exemplo solicitado em nada o auxilia no aprendizado dos conceitos em voga.

Se o exemplo selecionado pelo aprendiz é um arranjo de tabuleiro, a interface permite ainda que o aprendiz: (a) recupere as informações disponíveis sobre a(s) partida(s) à(s) qual(is) este arranjo em particular pertence, tais como nome dos jogadores, local e data de realização; e (b) possa percorrer toda a seqüência de jogadas que foram exploradas. Se o aprendiz optar por percorrer

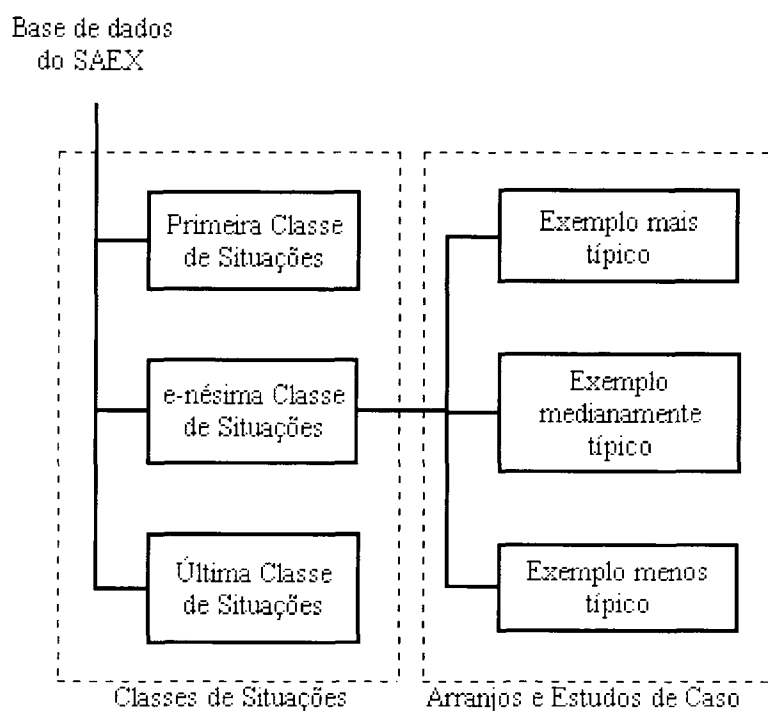


Figura 3.3 – DIAGRAMA DO MECANISMO DE APRESENTAÇÃO DO AMBIENTE DE EXPLORAÇÃO DA INPA

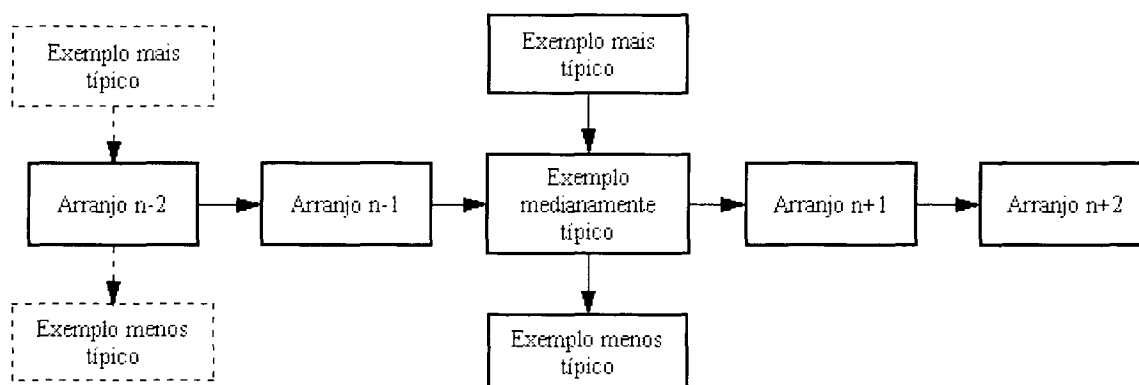


Figura 3.4 – DIAGRAMA DO PROCESSO DINÂMICO DE NAVEGAÇÃO PELA BASE DE DADOS

todos os arranjos do jogo selecionado, poderá, a qualquer tempo obter informações adicionais sobre se o arranjo atual pertence ou não a uma ou mais classes de situações. Caso pertença, o aprendiz poderá localizar dentro desta nova classe os exemplos mais e menos típicos e assim sucessivamente. A Figura 3.4 ilustra este processo dinâmico de navegação.

Se o aprendiz desejar, a interface permite ainda listar e examinar todos os exemplos que satisfaçam a determinadas condições, tais como: (a) pelo nome dos jogadores, (b) através da carga cognitiva do exemplo e (c) a presença ou não de determinadas características visuais.

A interface passiva é também comandada pela interface ativa (INA), apresentada com maiores detalhes na próxima seção.

3.3 A INTERFACE ATIVA

Além de uma interface passiva, com a qual o aprendiz pode construir o seu conhecimento de princípios, a ferramenta FEREN possui também uma interface ativa, denominada INA, capaz de fornecer os meios necessários para o desenvolvimento do que Lesgold denominou de conhecimento pericial.

Ao iniciar-se uma sessão, a interface INA detecta, através do histórico do aprendiz e das parametrizações previamente efetuadas pelo autor, qual a classe de situações e o primeiro exemplo a ser apresentado ao aprendiz. Na Figura 3.5 tem-se um diagrama mostrando lado-a-lado, dois instantes da apresentação de um estudo de caso submetido ao aprendiz. Ao iniciar-se a análise de uma classe de situações, a interface apresenta um texto que explicita qual o tipo de conhecimento pericial que deverá ser assimilado pelo aprendiz. A partir deste momento, todos os arranjos e estudos de caso apresentados serão acompanhados de uma indagação própria, que apesar de ser específica para aquele exemplo, exige uma resposta nos moldes do texto introdutório.

No instante da Figura 3.5, o texto introdutório é apresentado quando uma determinada classe de situações é acessada é o seguinte: “Toda vez que seu oponente faz um movimento, você deveria parar e pensar: Por que ele escolheu este movimento? Uma peça está em perigo? Há qualquer outra ameaça que eu deveria eliminar? Que tipo de plano o meu oponente tem em mente?”

A ferramenta então seleciona um exemplo para ser apresentado pela interface INA ao aprendiz. Terminada a apresentação, a ferramenta fornece ao aprendiz uma indagação, específica para o exemplo que acaba de ser exibido. A resposta esperada pela ferramenta tem um vínculo direto com o texto introdutório. No instante da Figura 3.5, a indagação, por exemplo será: “Neste exemplo você está jogando com as peças pretas. O seu oponente moveu a Dama para a coordenada [f3]. Existe alguma peça sua em perigo? O que ele tem em mente? – Responda agora: Que movimento você deverá fazer para impedi-lo?”

Se o aprendiz responde corretamente à indagação, a interface credita o acerto e, tendo o aprendiz satisfeito os parâmetros definidos previamente

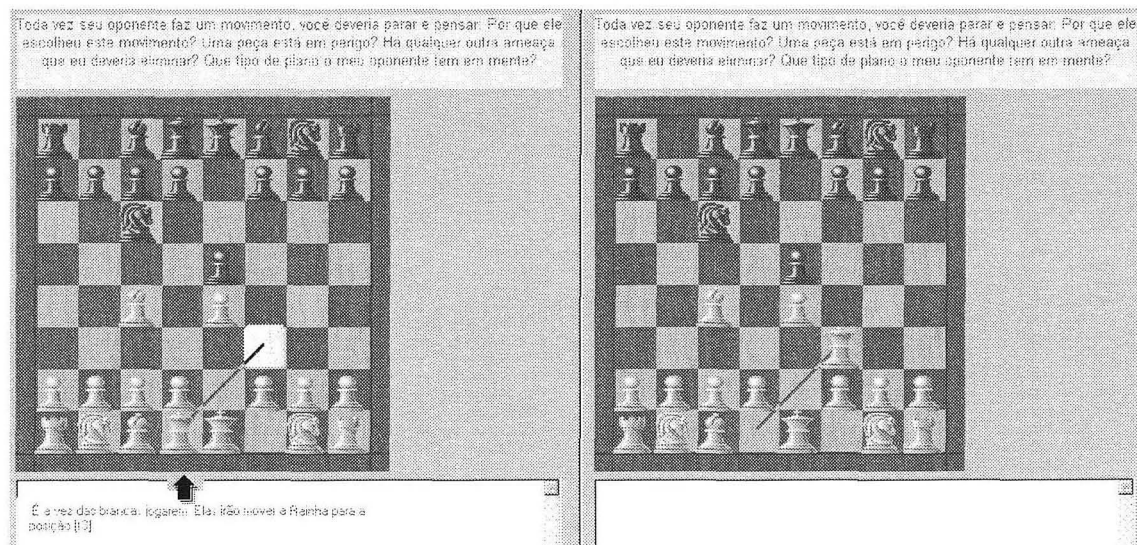


Figura 3.5 – DOIS INSTÂNTES DA APRESENTAÇÃO DE UM ESTUDO DE CASO PELA INTERFACE INA

pele autor, tais como a quantidade de acertos necessários, o mesmo é então promovido para outra classe de situações. Caso o aprendiz acerte mas a promoção não ocorra, outro exemplo de menor tipicidade (mais difícil) é então apresentado, repetindo-se o processo até que o aprendiz seja promovido ou a sessão se encerre. As condições capazes de dar por encerrada uma sessão também são definidas pelo autor durante o processo de autoria.

Se o aprendiz não responde corretamente a indagação, ou a deixa sem resposta, a interface então seleciona outro exemplo, de semelhante grau de tipicidade e o apresenta a partir da interface INPA. Espera-se assim, que através de sucessivos exemplos, e mediante a exploração da base de dados do SAEX, o aprendiz consiga compreender o conceito presente por detrás do exemplo apresentado e assim responder corretamente à indagação da interface. A ferramenta FEREN apresenta novos exemplos, promove o aprendiz para novas classes de situações ou encerra a sessão de ensino, baseada nas respostas do aprendiz face aos exemplos selecionados.

Durante o processo de ensino, a interface INA bloqueia o exemplo em análise, de forma tal a não permitir que o aprendiz utilize a interface INPA para inspecioná-lo e assim burlar o mecanismo de avaliação.

3.4 PERSPECTIVAS DE APLICAÇÃO

O SAEX é um ambiente de ensino-aprendizado específico para a aquisição de perícia em conceitos visuais específicos para o Jogo de Xadrez. Conforme analisado na sub-seção 2.1.1, a aquisição desta perícia consiste basicamente em desenvolver no aprendiz a capacidade de classificar imagens, identificar características e descrever anomalias. Para que a aquisição de perícia seja possível, é necessário então que o aprendiz tenha acesso a um grande número de exemplos, devidamente catalogados em conjuntos de características visuais semelhantes e ordenados.

Os Níveis Conceitual e de Produção do SAEX fornecem justamente as ferramentas necessárias para isto. Enquanto a ferramenta do Nível Conceitual (Capítulo 5) encarrega-se de permitir ao autor criar tantas classes de situações quantas o mesmo julgar necessário, a ferramenta do Nível de Produção (Capítulo 4) permite carregar na base de dados do sistema, um sem número de exemplos, todos derivados de jogos oficiais. Estão atendidos portanto os pré-requisitos para a aquisição de perícia: o SAEX possui ferramentas capazes de carregar na base de dados um grande número de exemplos, devidamente catalogados e ordenados.

Estando o processo de autoria concluso, cabe à ferramenta do Nível Instrucional apresentar estas classes de situações e exemplos ao aprendiz de forma a desenvolver no mesmo a perícia desejada. Mas como desenvolver a perícia de forma eficiente, a partir de todas as classes de situações e de um universo tão grande de exemplos?

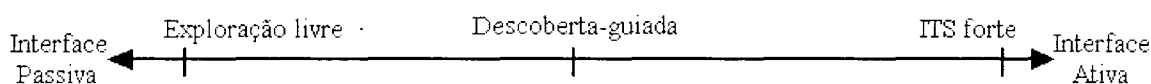


Figura 3.6 – CLASSIFICAÇÃO DOS MECANISMOS PEDAGÓGICOS PARA A TRANSMISSÃO DE CONHECIMENTO

Uma possível resposta a estas indagações é permitir ao aprendiz o livre acesso ao banco de dados do tutor, tal como ocorre com as pesquisas efetuadas na Internet. A este mecanismo pedagógico de ensino denomina-se “Exploração Livre” (extremo esquerdo da Figura 3.6). A exploração livre presta-se bem para a aquisição de princípios. Entretanto, os aprendizes normalmente sentem dificuldade de adquirir perícia a partir deste mecanismo, porque para implementar a exploração livre, a interface torna-se passiva. Em uma interface passiva, sempre deve partir do aprendiz a iniciativa de definir o que pesquisar e de interpretar se os exemplos fornecidos pela interface apresentam ou não um determinado conjunto de características visuais. Entretanto, reconhecer em um determinado exemplo o conjunto de características visuais presentes é justamente perícia que, espera-se, o aprendiz acabe por adquirir.

Assim, apesar da exploração livre ser adequada para a aquisição de princípios, a exploração livre, por si só, não dá garantias de que o aprendiz efetivamente conseguirá adquirir a perícia desejada.

Outra possível resposta às indagações efetuadas é dotar o ambiente do que denominamos de um “tutor forte” (extremo direito da Figura 3.6). Entendemos como um tutor forte um tutor cuja interface com o aprendiz intervêm ativamente no processo de ensino-aprendizagem. Um tutor forte necessariamente deverá implementar um modelo de estudante e um modelo pedagógico desenvolvido. Ao modelo de estudante caberá identificar as características individuais de cada aprendiz, seu nível atual de perícia, suas potencialidades e deficiências. Já ao modelo pedagógico caberá escolher e

apresentar de forma adequada ao aprendiz os exemplos necessários para a compreensão dos conceitos em estudo.

Uma solução intermediária é a implementação de um mecanismo de descoberta-guiada (centro da Figura 3.6). Através da descoberta-guiada têm-se um meio-termo entre uma interface de exploração totalmente livre e a existência de um tutor forte. Na descoberta-guiada, o aprendiz é quem determina, em última instância, se ele deseja pesquisar a base de dados do ambiente ou acompanhar o desenrolar de uma atividade instrucional desenvolvida pelo tutor. Assim sendo, ora a ferramenta de ensino comporta-se de forma passiva, ora de forma ativa.

O SAEX implementa em sua ferramenta FEREN o método pedagógico da descoberta-guiada, através de suas duas interfaces, a INA e a INPA. Através destas interfaces, o aprendiz pode tanto pesquisar a base de dados do sistema quanto assistir a uma sessão de ensino. A interface ativa pode, em determinado momento, utilizar a interface passiva para pesquisar um outro exemplo que melhor exemplifique um conteúdo. E o aprendiz também pode, por sua vez, utilizar a interface passiva para obter o conhecimento necessário para responder a questionamentos da interface ativa, quanto utilizar a interface ativa para responder a dúvidas surgidas a partir da exploração da base de dados do tutor.

Este processo dinâmico de interação permite que o aprendiz dite suas próprias regras de conduta que, respeitadas pela ferramenta, acreditamos que propiciarão um maior conforto e eficácia na utilização da ferramenta.

Visto o processo de ensino-aprendizagem, nos dois próximos capítulos será analisado como o autor pode incorporar dentro do sistema o conhecimento pericial necessário.

CAPÍTULO 4

O NÍVEL DE PRODUÇÃO

O processo de autoria foi desmembrado nos Níveis Conceitual e de Produção para permitir que o SAEX seja manipulado por dois tipos distintos de autores: (a) um especialista no domínio e (b) um especialista pedagógico. Este desmembramento pode ser observado através da Figura 1.2. Enquanto a ferramenta de autoria do Nível Conceitual deve ser utilizada por especialistas no domínio do Jogo de Xadrez, a ferramenta de autoria do Nível de Produção pode ser utilizada também por autores (normalmente professores) com menor especialização no domínio do conhecimento, mas, por sua vez, mais voltados à prática do ensino e à realidade do aprendiz.

Este capítulo descreve como o autor, utilizando-se da ferramenta de autoria do nível de produção, pode: (a) incorporar à base de dados do SAEX arranjos de jogos previamente disponibilizados em arquivos no formato PGN, (b) criar seus próprios casos para estudo e (c) organizar estes arranjos e casos em uma seqüência adequada para apresentação pela *shell* de ensino.

Na Seção 4.1 apresenta-se a ferramenta de descrição de situações e estudos de casos (DESCA), com sua arquitetura interna e interface. A Seção 4.2 descreve o mecanismo de incorporação dos arranjos de jogos armazenados no formato PGN na base de dados do SAEX. Os recursos de autoria disponíveis para a construção de casos para estudo serão descritos e analisados na Seção 4.3, e a Seção 4.4 apresenta a forma pela qual o SAEX permite ao autor sequencializar estes arranjos e casos para estudo.

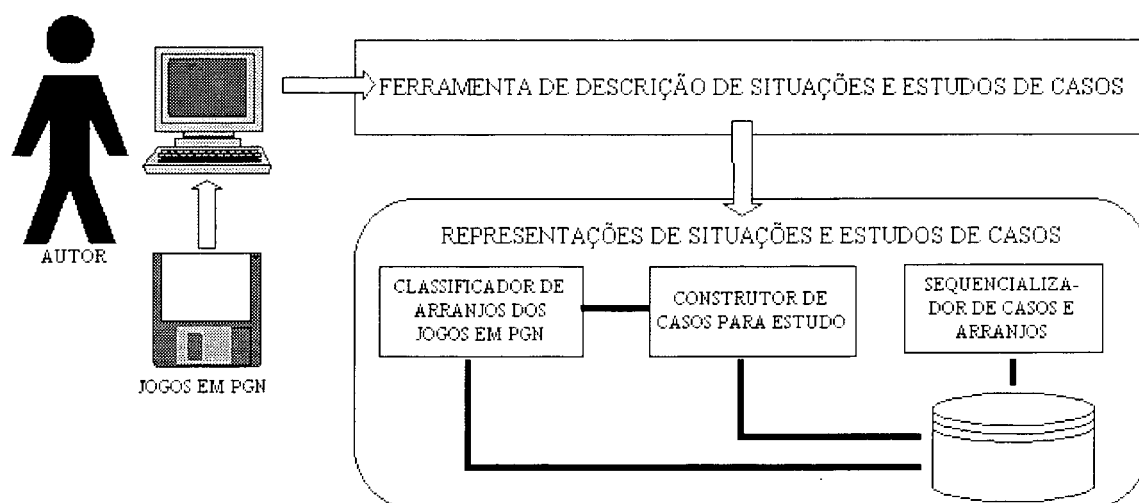


Figura 4.1 – O NÍVEL DE PRODUÇÃO

4.1 FERRAMENTA DE DESCRIÇÃO DE SITUAÇÕES E ESTUDO DE CASOS

No nível de produção, as classes de situações e seus respectivos mecanismos pedagógicos *default* já foram criados e armazenados na base de dados do SAEX pela ferramenta do Nível Conceitual (vide capítulo 5). Resta agora ao autor: (a) instanciar estas classes, através da adição de arranjos de jogos pré-existentes e da criação de casos particulares de estudo e (b) fornecer à *shell* de ensino (vide capítulo 3) a seqüência considerada ideal pelo autor para a apresentação destas classes, arranjos e estudos. Estas atividades de autoria são possíveis através do uso de uma ferramenta específica, denominada DESCA.

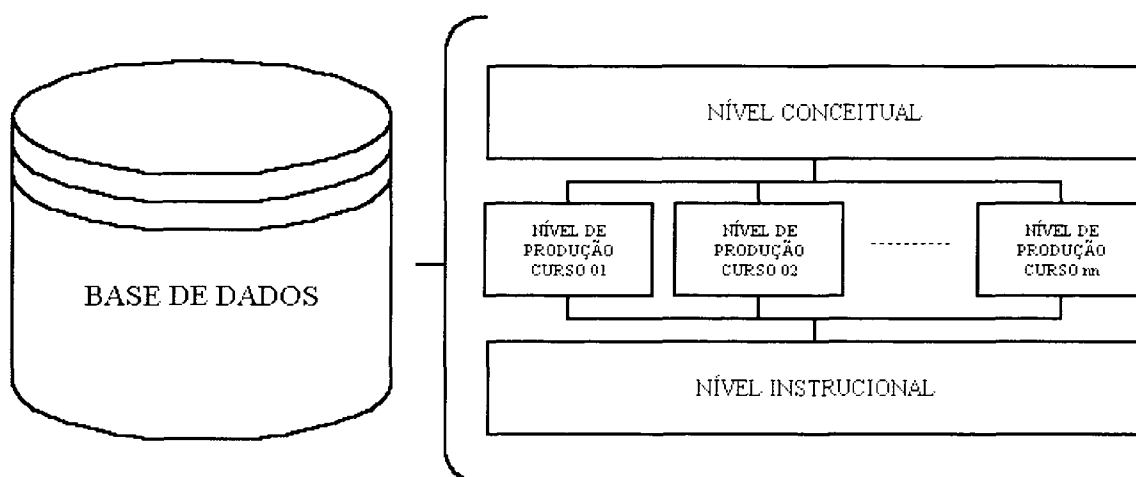


Figura 4.2 – O SAEX PERMITE A EXISTÊNCIA DE DIFERENTES CURSOS EM NÍVEL DE PRODUÇÃO

4.1.1 Arquitetura básica

O Nível de Produção é constituído de uma ferramenta de autoria (DESCA) e de uma base de dados. Esta ferramenta possui 3 (três) módulos distintos (Figura 4.1): (a) o Classificador de Arranjos de Jogos em PGN; (b) o Construtor de Casos para Estudo e (c) o Sequencializador de Casos e Arranjos.

A DESCAs utiliza as definições e parâmetros construídos no Nível Conceitual, obtidos a partir da própria base de dados do SAEX. Isto permite ao Nível de Produção ser totalmente independente dos demais níveis. Entre outras vantagens, esta independência permite ao autor do nível de produção criar sobre uma mesma base de dados diferentes cursos, o que pode propiciar, por exemplo, a criação de diferentes exemplos e sessões de ensino, para diferentes tipos de alunos. Isto é possível porque o conhecimento referente ao Nível de Produção pode ser replicado em diferentes instâncias (Figura 4.2). Têm-se assim uma única base de dados, mas que é capaz de armazenar diferentes cursos do Nível de Produção. A DESCAs permite manipular apenas um curso por vez, e cabe a um humano (aprendiz ou professor) selecionar qual versão deve ser utilizada durante uma determinada sessão de ensino.

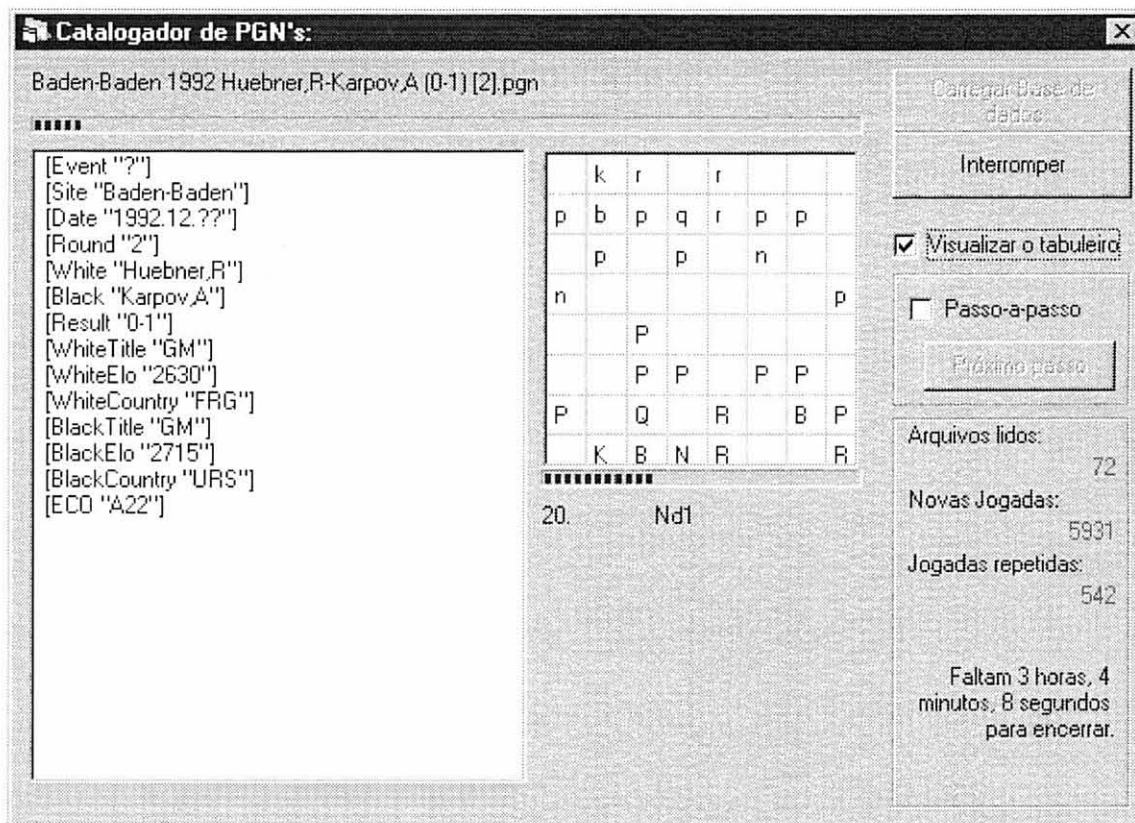


Figura 4.3 – CLASSIFICADOR DE ARRANJOS DE JOGOS EM PGN

Para permitir a autoria, a ferramenta DESCAs possui 3 (três) interfaces gráficas distintas para facilitar o trabalho do autor e que permite a manipulação dos dados armazenados na base do SAEX.

4.1.2 Interfaces da Ferramenta DESCAs

A primeira interface é a do classificador de arranjos de jogos em PGN (Figura 4.3). É ela a responsável por selecionar os diversos arranjos de tabuleiros que atendam às especificidades de cada uma das classes de situações construídas no Nível Conceitual. Através desta interface, o autor pode indicar quais são os arquivos que contêm os jogos desejados e em qual base de dados estes mesmos jogos deverão ser instanciados nas respectivas classes de situações. A partir deste momento a camada de software adjacente à interface se encarrega de: (a) ler cada um dos arquivos selecionados, (b) recuperar os jogos armazenados no formato PGN nestes arquivos, (c) recriar

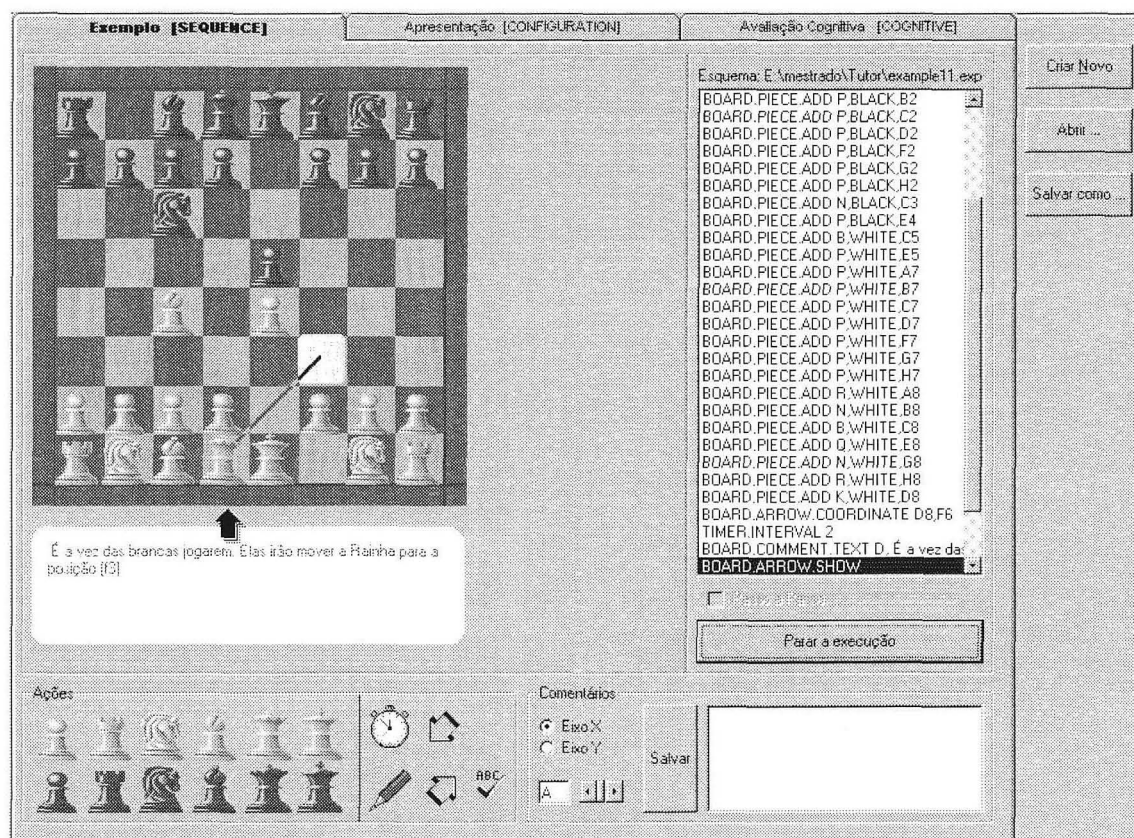


Figura 4.4 – CONSTRUTOR DE CASOS PARA ESTUDOS

cada um dos arranjos que compõem estes jogos e (d) verificar se cada um destes arranjos pode ser classificado como um exemplo de determinada classe de situações através do exame das características visuais de cada arranjo e classe. Se um determinado arranjo é considerado pelo classificador como um exemplo para determinada classe de situações, o mesmo é instanciado como objeto daquela classe e tanto ele quanto os demais arranjos deste jogo são armazenados na base de dados do SAEX. Se um determinado jogo não possuir nenhum arranjo instanciado, o mesmo é descartado.

A segunda interface da ferramenta DESCAs permite ao autor criar seus próprios estudos de casos, que quando concluídos, também deverão ser instanciados como objetos de uma determinada classe de situações (Figura 4.4). Os arranjos são ‘imagens’ estáticas de determinados momentos dentro de um jogo. Entretanto, em muitos casos, é necessário criar ‘animações’ onde o aprendiz possa visualizar o desenrolar de uma seqüência de jogadas. Os

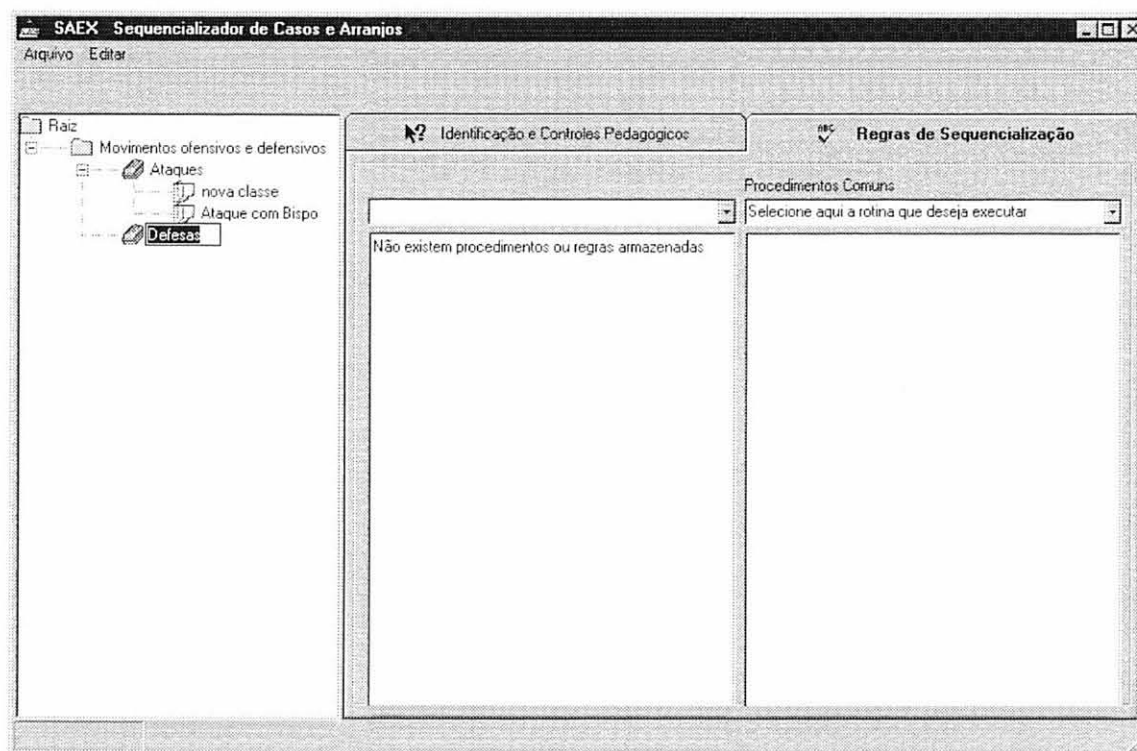


Figura 4.5 – SEQUENCIALIZADOR DE CASOS E ARRANJOS

estudos de casos são os objetos que permitem criar estas “animações”.

Através desta interface o autor pode dispor de um tabuleiro da maneira que julgar mais conveniente, construindo o seu caso passo-a-passo. Através da utilização de primitivas manipuladas graficamente, o autor pode: (a) adicionar, mover, capturar ou promover peças; (b) dispor textos, comentários e linhas de apoio; (c) retirar ou colocar as marcações de coordenadas nas laterais do tabuleiro e (d) manipular os intervalos de tempo entre as ações.

A terceira interface que compõe a Ferramenta DESCA, em Nível de Produção, é a que permite a sequencialização dos arranjos e estudos de casos dentro das diferentes classes de situações (Figura 4.5). Através desta interface, o autor pode atribuir parâmetros que conferem a um caso ou arranjo uma certa ordem, a qual pode ser (ou não) baseada em sua carga cognitiva. Tal carga, como será visto com mais detalhes na Seção 4.4, é constituída de diversos fatores a serem definidos pelo autor.

Nos próximos itens estão descritos as funções e princípios utilizados por cada um destes 3 (três) módulos.

4.2 CLASSIFICADOR DE PGN's

4.2.1 A Inadequação de técnicas clássicas de IA

Na IA clássica, quando se imagina em como desenvolver estratégias para jogos tais como o Xadrez, a solução normalmente apontada é utilizar um dos diversos algoritmos de busca que existem, dentre os quais o que mais se destaca é o Minimax (RICH, 1993) (um breve esquema de funcionamento do algoritmo Minimax é apresentado na Figura 4.6). No Xadrez isto se dá porque este jogo apresenta, durante a sua execução, aproximadamente 35^{100} (ou 10^{120}) arranjos possíveis (ainda que apenas 10^{40} destes possam ser considerados válidos), e o algoritmo Minimax possui características próprias que permitem reduzir substancialmente este espaço de busca. A força do algoritmo Minimax está: (a) na sua função de avaliação e (b) nos cortes alfa e beta. O fator de ramificação médio dos cortes alfa e beta foi analisado em profundidade por Baudet e Pearl (BAUDET, 1978, PEARL, 1982).

Sempre que a função de avaliação recebe um arranjo do tabuleiro, ela estima um valor numérico real desejavelmente proporcional à probabilidade de vitória de um dos jogadores (e à probabilidade de derrota do jogador adversário). A função de avaliação é antes de tudo uma função heurística, cujo valor de retorno é uma aproximação matemática das análises efetuadas por jogadores humanos. As funções utilizadas nos jogos de Xadrez normalmente apresentam diversos fatores, cada qual representando uma sub-área do domínio. Como exemplo de fator, podemos indicar:

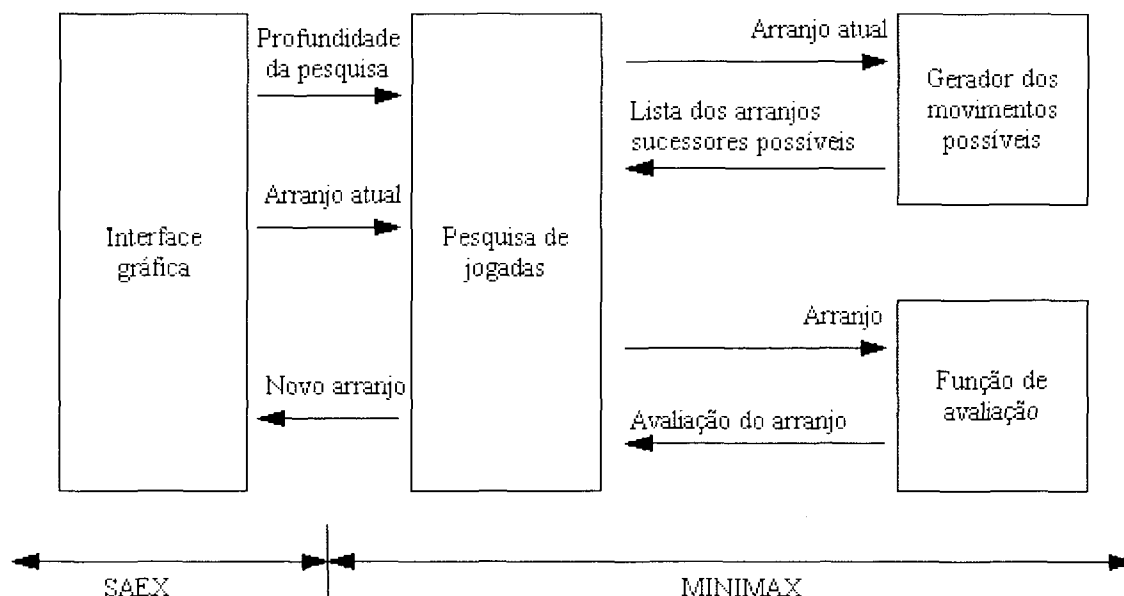


Figura 4.6 – O ALGORITMO MINIMAX

- Material: peão = 1, cavalo e bispo = 3, torre = 5, dama = 9. Valores como estes podem variar dependendo da situação e do instante do jogo (por exemplo, a obrigatoriedade de manter o par de bispos até o final do jogo);
- Posição: nos primórdios da história do Jogo de Xadrez pensava-se que o controle do centro era fundamental. Hoje é consenso entre os mestres enxadristas que o controle do centro é muito importante, mas não o único;
- Defesa: os aspectos defensivos de uma posição têm relação com a segurança do rei;
- Tempo: define a “precisão” com que a função pode analisar certos aspectos do arranjo.

Muitas funções de avaliação (f) permitem a independência entre seus fatores componentes, onde cada um (w_1, w_2, \dots, w_n), por sua vez, é uma função linear com um peso global a ela associado (x_1, x_2, \dots, x_n), na forma geral abaixo:

$$f = x_1 \cdot w_1 + x_2 \cdot w_2 + \dots + x_n \cdot w_n$$

Para maximizar a eficácia da função de avaliação, faz-se necessário que o algoritmo construa uma árvore de pesquisa. Para tanto o Minimax precisa determinar continuamente os conjuntos de sucessores legais para um determinado arranjo do tabuleiro. Esta é a função do gerador de movimentos possíveis: determinar todas as possibilidades de movimentos que um jogador poderá efetuar a partir de um dado arranjo do tabuleiro, respeitando as regras do jogo.

Uma vez que a árvore de pesquisa atingiu a profundidade desejada, a função de avaliação (que sempre é aplicada apenas aos nós terminais da árvore) encarrega-se de determinar a qualidade do arranjo (Figura 4.7). Para aumentar a eficiência deste algoritmo utiliza-se os cortes alfa e beta, que efetuam a “poda” da árvore e conseqüentemente a poda do esforço explorativo e de avaliação desnecessário.

Entretanto, para a construção de um tutor, esta solução apresenta-se insuficiente, porque: (a) a função heurística é difícil de ser construída e analisada pelo especialista no domínio, por necessitar que a sua perícia iminentemente visual seja “traduzida” para expressões matemáticas e (b) o Minimax pode determinar com rapidez qual a melhor jogada, mas é inadequado para indicar o porquê da escolha.

Ainda assim foram realizadas alterações na estrutura do Minimax, na tentativa de contornar estas dificuldades. Foi criada uma lista de funções (em substituição à função de avaliação original), e uma lista de listas, cada qual contendo os arranjos percorridos pelo Minimax, do arranjo selecionado até o arranjo atual.

Através da lista de funções tentou-se eliminar, ainda que em parte, a dificuldade do especialista para criar a função de avaliação, que passou a ser composta de funções individuais, geradas a partir de primitivas que simbolizavam pequenas porções do conhecimento pericial do especialista. Já através da lista de listas dos arranjos percorridos procurou-se desenvolver um

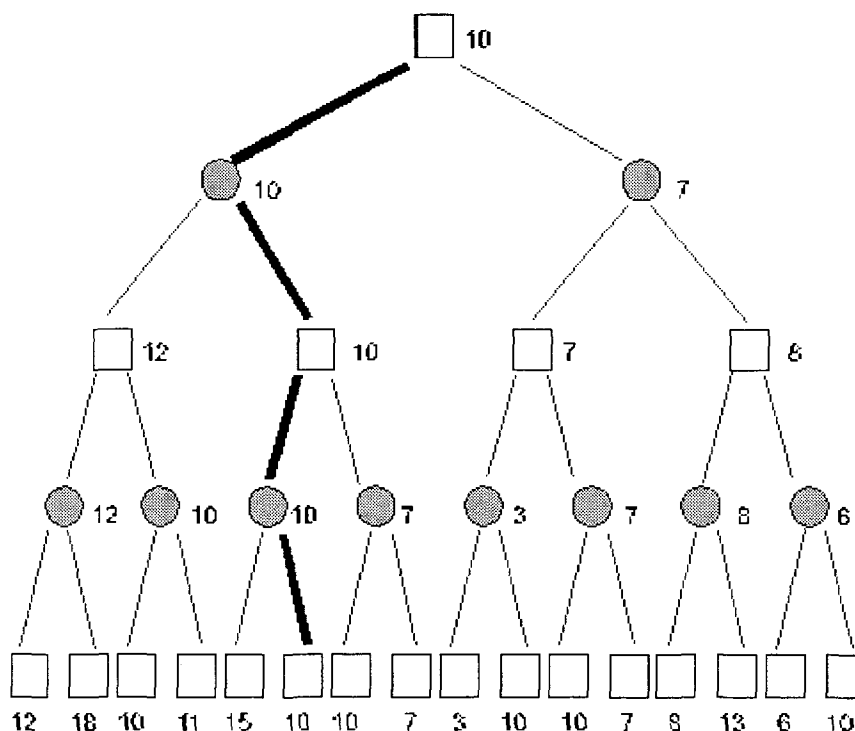


Figura 4.7 – A ÁRVORE DE PESQUISA CONSTRUÍDA PELO MINIMAX

mecanismo de explicação para a escolha efetuada através das funções de avaliação. Estas modificações acarretaram em mudanças substanciais no algoritmo original, pois para que o mesmo fosse capaz de apresentar uma explicação coerente, era necessário gerar não uma única lista de arranjos a partir da função de avaliação resultante da união de todas as sub-funções existentes, mas sim esta lista e mais uma lista para cada sub-função existente. Apenas a partir da criação desta lista de listas de arranjos é que o algoritmo pode explicar o porquê de um arranjo de solução em detrimento de outros.

Entre outros efeitos, estas modificações acarretaram em uma severa queda de desempenho do sistema, e tornou o processo de autoria extremamente complexo, pois a qualidade da seleção dos arranjos e das explicações eram fortemente limitadas pela qualidade das sub-funções de avaliação que eram desenvolvidas pelo autor.

Foi necessário então desenvolver uma abordagem original para o problema, diferente das comumente utilizadas. Esta nova abordagem está descrita na seção 4.2.2

4.2.2 Uma abordagem alternativa às técnicas clássicas de IA

Com base no exposto no item 4.2.1, outro fator determinante para o afastamento da solução clássica foi o fato de que todas as características visuais da perícia, definidas por Lesgold (vide Seção 2.1.1), não podiam ser diretamente manipuladas pelo Minimax. É certo que o especialista também se utiliza de uma abordagem hipotético-dedutiva semelhante à utilizada pelo Minimax, mas apenas para justificar ou explicar uma jogada. Na maior parte do tempo estes especialistas utilizam-se da sua enorme habilidade para a rápida comparação de padrões visuais, baseada em um volumoso estoque de “esquemas” mentais. Esquemas estes que são de difícil “tradução” para serem utilizados pelo Minimax.

Outra característica do algoritmo Minimax foi destacada por Beliner (BELINER, 1977): ao se implementar jogadores automáticos baseados no algoritmo Minimax, estes tornam-se incapazes de “apostar no erro” do adversário. Ou seja, o algoritmo é incapaz de selecionar jogadas que, em caso de erro do oponente, poderiam trazer uma efetiva vantagem ao jogador automático.

Tornou-se então necessário afastar-nos um pouco da IA clássica e desenvolver um novo meio através do qual o autor pudesse transferir o seu conhecimento pericial para o tutor e este para o aprendiz, sem basear-se em algoritmos de busca. A solução encontrada para este tutor foi a construção de primitivas visuais no Nível Conceitual (vide capítulo 5) e de um classificador de PGN's no Nível de Produção.

O termo PGN vem do inglês (*Portable Game Notation*), referindo-se ao fato dos registros dos jogos poderem ser arquivados de forma que qualquer pessoa possa consultá-los e até mesmo acompanhar as jogadas, observar os erros e feitos das partidas de Xadrez. O PGN é um arquivo do tipo texto, composto de caracteres em seqüência utilizando símbolos e rótulos pré-

```

[Event "?"]
[Site "Amsterdam Don mem"]
[Date "1995.07.14"]
[Round "3"]
[White "Piket,J"]
[Black "Seirawan,Y"]
[Result "0-1"]
[WhiteTitle "IM"]
[WhiteElo "2625"]
[WhiteCountry "NED"]
[BlackTitle "GM"]
[BlackElo "2625"]
[BlackCountry "USA"]
[ECO "D35"]

1. d4 Nf6 2. c4 e6 3. Nf3 d5 4. Nc3 Nbd7 5. cxd5 exd5 6. Qc2 c6 7. Bf4 Nh5 8. Bg3 g6 9. e3 Nxc3
10. hxg3 Bd6 11. Bd3 Nf6 12. O-O-O Be6 13. Kb1 Qe7 14. Ka1 O-O-O 15. Rc1 Kb8 16. Na4 h5 17.
Nc5 Bc8 18. b4 Ne4 19. Nxe4 dxe4 20. Bxe4 Rhe8 21. Bd3 Bxb4 22. Rb1 Ka8 23. Rb3 Bd6 24.
Rhb1 c5 25. a4 cxd4 26. exd4 Bc7 27. Qc3 Qd6 28. Bc4 Re7 29. Rb5 Bb6 30. Rd5 Qc7 31. a5
Rxd5 32. axb6 axb6 33. Kb2 Ra5 34. Qb4 Be6 35. Rc1 Bxc4 36. Rxc4 Re2+ 37. Kc3 Qe7 38. Qxe7
Rxe7 0-1

```

Figura 4.8 – EXEMPLO DE ARQUIVO TEXTO NO FORMATO PGN

definidos (Figura 4.8). Existem vários programas que podem ser utilizados para visualizar este tipo de arquivo, dentre os quais ressaltamos o *Arasan* (<http://www.internetchess.com/software.shtml>), programa este que pode, além de ler os arquivos PGN, jogar com o usuário.

Esta notação tornou-se o padrão para o registro e acompanhamento de jogos em campeonatos e disputas oficiais. Como cada anotação (por exemplo: 10. hxg3 Bd6) representa um lance com 2 jogadas (hxg3 e Bd6), pode-se facilmente, a partir de um PGN, recriar os arranjos correspondentes e confrontá-los visualmente com as classes de situações criadas pelo autor com a ferramenta do Nível Conceitual. Uma vez armazenados na base de dados do SAEX, estes arranjos podem ser facilmente recuperados, não a partir da aplicação de funções heurísticas de avaliação, mas através de suas características visuais, sendo as demais jogadas recuperadas a partir do próprio jogo selecionado.

Consegue-se assim resolver os dois problemas básicos da autoria: (a) o autor pode manipular visualmente as situações do tabuleiro e, (b) o sistema pode apresentar as explicações baseando-se nas regras pedagógicas

de cada classe de situações (vide Capítulo 5), bem como nas jogadas, observações e comentários dos próprios especialistas envolvidos nos jogos armazenados na base de dados. Apenas selecionando os jogos a serem armazenados na base do SAEX, o autor pode ainda criar diferentes “cursos”, com diferentes ênfases, graus de dificuldade ou estilos de jogadas. Isto é possível por que o classificador de PGN’s analisa apenas os arquivos PGN selecionados pelo autor, recriando para cada jogo, todos os arranjos desenvolvidos durante a partida, submetendo-os um a um às regras de classificação definidas no Nível Conceitual, para só então armazenar ou descartar os jogos em análise.

Estes arquivos PGN’s podem ser facilmente encontrados em diversos sites da Internet, específicos para a disseminação do Jogo de Xadrez (por exemplo: <http://www.chol.com.br/>), ou ainda nas diferentes sedes oficiais das entidades que promovem campeonatos e torneios (por exemplo: *Confederação Brasileira de Xadrez – CBX*). Outra fonte expressiva deste tipo de arquivos são os jogadores automáticos, programas comerciais que em sua maioria, disponibilizam uma grande variedade de partidas neste formato (por exemplo: *USF Chess*).

4.3 CONSTRUTOR DE CASOS PARA ESTUDO

Conforme explicitado na Seção 4.2, o Classificador de PGN’s permite ao autor transferir todo o trabalho de apropriação de exemplos de arranjos para a ferramenta DESCAs. Entretanto, existem situações em que o autor necessita criar os seus próprios exemplos: (a) quando o exemplo em questão não existe na base de dados do SAEX, ou quando o esforço computacional para encontrá-lo é superior à prioridade do autor; (b) quando se trata de criar um

contra-exemplo, que dificilmente será encontrado em jogos reais entre humanos e (c) quando, no ensino de estratégias, o autor necessitar explicitar não um único arranjo, mas uma seqüência de arranjos. Para permitir ao autor superar estas dificuldades, foi desenvolvido então o construtor de casos para estudo.

A interface do construtor de casos para estudo (vide Figura 4.4) é guiada por descrições de uma base de dados, as quais são construídas com base em primitivas de autoria. As referidas primitivas estão descritas na tabela 4.1. Esta tabela está organizada de forma a detalhar os principais elementos que estão associados na composição de uma animação idealizada pelo autor. Os referidos elementos são subordinados a um objeto computacional denominado “*board*”, o qual é uma estrutura fundamental, presente nas ferramentas (*shell's*) dos 3 níveis do ambiente.

Com estas primitivas é possível dar liberdade suficiente ao autor de cursos para que este possa expressar criações originais que eventualmente nunca surgiriam em um jogo real. Isto se dá porque a originalidade é a chave não só para a competição mas também para o ensino. Tal originalidade torna-se ainda mais marcante nos pontos mais centrais da partida, onde a explosão combinatória de posições é impossível de ser abordada de forma detalhada.

4.4 SEQUENCIALIZADOR DE CASOS E ARRANJOS

A abordagem fundamental para a ordenação ou seqüencialização dos casos para estudo e arranjos por meio da ferramenta DESCAs é baseada no conceito de Carga Cognitiva. Este conceito já foi explorado anteriormente por vários autores, porém poucos deles aplicaram as suas idéias para sistemas tutores inteligentes (PIMENTEL, 1997; KILLGARIFF, 1988).

OBJETO	PRIMITIVA	FORMA GERAL	DESCRIÇÃO
		EXEMPLO	
PIECE	ADD	BOARD.PIECE.ADD <tipo de peça>, <cor>, <coord>	Esta primitiva instancia uma nova peça no tabuleiro, na coordenada h8
		BOARD.PIECE.ADD Q, WHITE, H8	
	MOVE	BOARD.PIECE.MOVE <coord1>, <coord2>	Movimenta a peça existente na coordenada1 até a coordenada2
		BOARD.PIECE.MOVE D7, D5	
	PROMOTION	BOARD.PIECE.PROMOTION <tipo de peça>, <coord>	Promove o peão indicado pela coordenada para o tipo de peça indicado
		BOARD.PIECE.PROMOTION Q, F1	
	CAPTURE	BOARD.PIECE.CAPTURE <coord1>, <coord2>	Move a peça existente na coordenada1 para a coordenada2, capturando a peça ali existente
		BOARD.PIECE.CAPTURE D5, E7	
AXIS	ALIGN	BOARD.AXIS.ALIGN TOP BOTTOM	Indica se as brancas estão na parte superior ou inferior do tabuleiro
		BOARD.AXIS.TOP	
	SHOW	BOARD.AXIS.SHOW	Torna visíveis as coordenadas nas laterais do tabuleiro, respeitando a posição de partida das peças brancas
		BOARD.AXIS.SHOW	
	HIDE	BOARD.AXIS.HIDE	Torna invisíveis as coordenadas das laterais do tabuleiro
		BOARD.AXIS.HIDE	
ARROW	COORDINATE	BOARD.ARROW.COORDINATE <coord1>, <coord2>	Cria uma linha reta ligando a coordenada1 à coordenada2
		BOARD.ARROW.COORDINATE C6, G2	
	SHOW	BOARD.AXIS.SHOW	Torna visível a linha
		BOARD.AXIS.SHOW	
	HIDE	BOARD.AXIS.HIDE	Torna invisível a linha
		BOARD.AXIS.HIDE	
COMMENT	TEXT	BOARD.COMMENT.TEXT <linhalcoluna>, [<texto>]	Torna visível um balão com o texto apresentado, no lado esquerdo/labaixo do tabuleiro, com uma seta indicando a linhalcoluna indicada. Se o texto não for fornecido, torna o balão invisível
		BOARD.COMMENT.TEXT D, TESTE	

Tabela 4.1 – PRIMITIVAS DO CONSTRUTOR DE CASOS DE ESTUDO

Para conseguir uma medida que quantifique cognitivamente um exemplo ou uma imagem, no caso do ensino de conceitos visuais, a ser mostrado para um aprendiz, precisa-se analisar como este exemplo irá contribuir para a evolução do aprendizado. A Carga Cognitiva pode então ser definida como a capacidade que um exemplo tem em exercitar o aprendiz. A

carga cognitiva de um exemplo pode ser dividida em alguns componentes, onde cada um destes componentes irá medir um tipo de contribuição ao aprendizado. No caso do ensino de conceitos visuais para radiologia médica existem algumas medidas já bastante usadas, que são, segundo Cury (CURY, 1996), a frequência, a saliência e a confiabilidade.

Quando confrontados com a dinâmica do Jogo de Xadrez, poucos destes fatores se aplicam na determinação da carga cognitiva de um arranjo ou caso para estudo. Isto se dá porque as definições convencionais para a representação de semelhança, tipicidade e contraste com casos “normais” (sem anomalia) não parecem se aplicar ao domínio de Xadrez da mesma forma como se aplicam para a área de radiologia médica. Faz-se necessário encontrar outros fatores que, sendo mais específicos para o domínio do Xadrez, permitam mensurar corretamente a carga cognitiva dos conceitos visuais existentes nos diferentes arranjos e casos para estudo.

Na abordagem deste trabalho, a carga cognitiva é utilizada em duas modalidades. A primeira se destina à busca de casos e arranjos que motivem o aprendiz a reconhecer estratégias de jogadas do oponente. A segunda modalidade é utilizada para encontrar casos e arranjos que ajudem o aprendiz a construir formas de mascarar suas estratégias de jogo para que o seu oponente não perceba o seu objetivo. Apesar de serem duas modalidades distintas de ordenação, a medida cognitiva de cada caso ou arranjo é a mesma para ambas as modalidades.

Durante o processo de autoria, através da interface de sequencialização, o autor deverá definir uma série de parâmetros, dentre os quais a carga cognitiva dos diversos casos e arranjos. São estes os parâmetros:

- a) A seqüência considerada ideal pelo autor para a apresentação das diversas classes de situações e em que momento do curso elas serão apresentadas;

- b) Os parâmetros a serem usados pela *shell* de ensino para a adaptação da seqüência das classes ao desempenho do aprendiz;
- c) O número de arranjos e estudos de casos a serem analisados pelo aprendiz durante a apresentação de uma determinada classe;
- d) A seqüência, explícita ou não, de apresentação dos exemplos de uma classe de situações. Se a seqüência for explicitamente fornecida, a *shell* apresenta os exemplos segundo esta seqüência;
- e) A carga cognitiva de cada exemplo. Se a seqüência não for fornecida, ou estiver incompleta, a *shell* de ensino deverá complementá-la através da análise da medida cognitiva de cada um dos exemplos;
- f) O peso que cada uma das medidas cognitivas (aplicáveis) terá na ordenação dos arranjos dentro da classe.

Se os três primeiros itens (a, b, c), por sua vez, já foram alvo de diversos trabalhos, os demais, referentes à carga cognitiva dos arranjos e casos, necessitam de um estudo mais aprofundado para a determinação de quais são os fatores que devem necessariamente ser levados em consideração no momento da sequencialização dos exemplos, pela *shell* de ensino. Esta é uma área que neste trabalho não foi analisada com maior interesse, mas que suscita a necessidade de maiores pesquisas no futuro, o que acreditamos, será de grande valia.

CAPÍTULO 5

O NÍVEL CONCEITUAL

Tal como mencionado no capítulo 4, o processo de autoria foi desmembrado nos Níveis Conceitual e de Produção para permitir que o SAEX seja manipulado por dois tipos distintos de autores: (a) um especialista no domínio e (b) o especialista pedagógico. Este desmembramento pode ser observado através da Figura 1.2. Enquanto a ferramenta de autoria do Nível de Produção pode ser utilizada por autores (normalmente professores) com menor especialização no domínio do conhecimento, a ferramenta do Nível Conceitual deve ser utilizada por especialistas no domínio do Jogo de Xadrez.

Este capítulo descreve como o autor especialista, utilizando-se da ferramenta de autoria do Nível Conceitual, pode: (a) criar e manter classes de situações do domínio e (b) criar estruturas pedagógicas de ensino adequadas para a apresentação destas mesmas classes de situações pela *shell* de ensino.

Na Seção 5.1 apresenta-se a ferramenta de descrição de classes de situações (DECLASS), com sua arquitetura interna e interface. A Seção 5.2 explicita o mecanismo de descrição das classes de situações do domínio, a anatomia do tabuleiro, a linguagem de programação, as primitivas de enquadramento e a representação interna do domínio utilizadas pela ferramenta. O mecanismo de representação pedagógica, sua linguagem, primitivas e mecanismos internos de representação estão descritos na Seção 5.3.

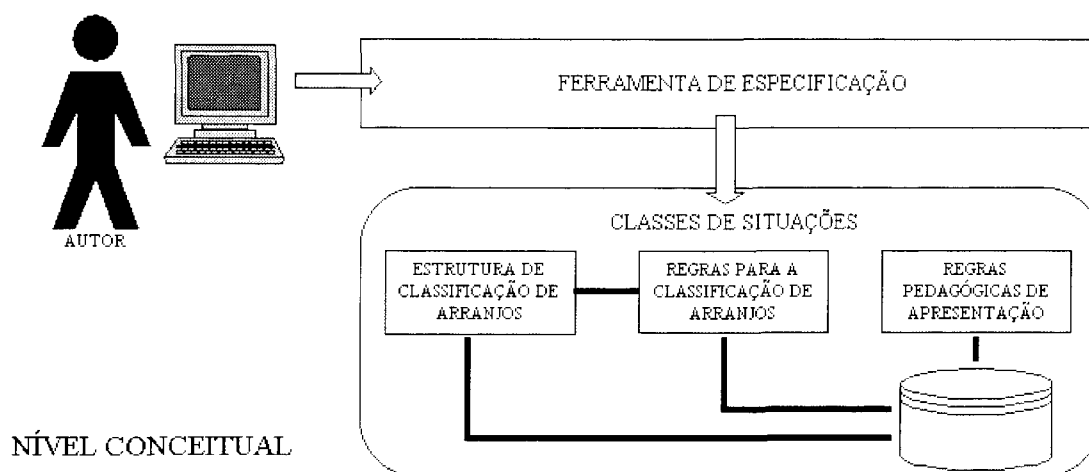


Figura 5.1 – O NÍVEL CONCEITUAL

5.1 FERRAMENTA DE DESCRIÇÃO DE CLASSES DE SITUAÇÕES

No Nível Conceitual o autor deve: (a) criar as classes de situações que julgar necessárias e (b) definir, para cada umas destas classes, tanto as regras de classificação quanto as de carácter pedagógico. Para tanto o autor deve utilizar a ferramenta de descrição de classes de situações (DECLASS) que permite criar as classes de situações e suas respectivas representações pedagógicas, posteriormente utilizadas pela ferramenta do Nível de Produção (vide Capítulo 4).

5.1.1 Arquitetura Básica

O Nível Conceitual é constituído de uma ferramenta de autoria e de uma base de dados. Esta ferramenta possui 3 (três) módulos distintos (Figura 5.1): (a) a Estrutura de classificação de arranjos, (b) o Editor de regras para a classificação de arranjos, e (c) o Editor de regras pedagógicas de apresentação.

É através desta ferramenta que são criadas as classes de situações que, no Nível de Produção (Figura 1.2), serão utilizadas para classificar os arranjos componentes dos jogos armazenados no formato PGN. E também é

através dela que serão criadas as representações pedagógicas de cada uma das classes de situações, que no Nível Instrucional (vide Figura 5.2 e Capítulo 3), serão utilizadas para apresentar visualmente ao aprendiz os arranjos pertencentes às diferentes classes de situações.

Neste tutor, as classes de situações são representadas por nós de uma árvore. Cada nó desta árvore é uma estrutura capaz de armazenar os procedimentos e chamadas de função utilizadas pelo Classificador de PGN's (vide Capítulo 4), bem como as regras pedagógicas utilizadas pela interface gráfica da *shell* de ensino (vide Capítulo 3) para a exibição dos arranjos. Os casos para estudo possuem suas próprias regras pedagógicas de apresentação e não utilizam as regras definidas no Nível Conceitual.

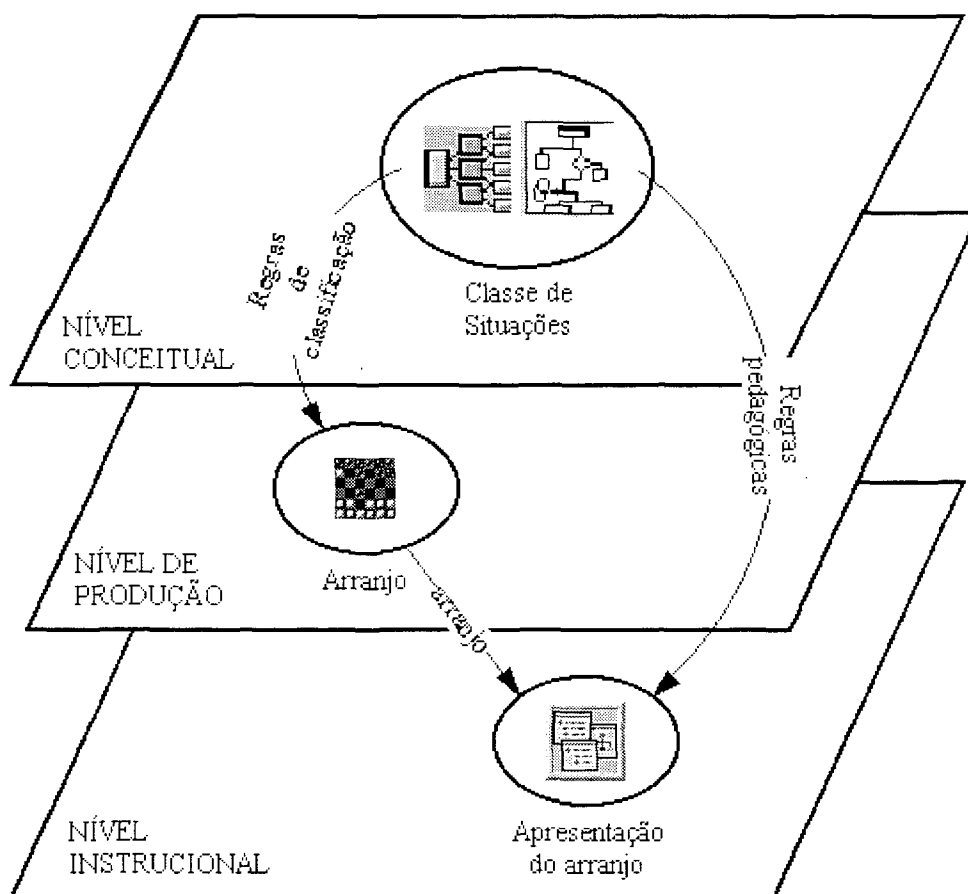


Figura 5.2 – O NÍVEL CONCEITUAL E O PROCESSO DE INTERAÇÃO COM OS DEMAIS NÍVEIS

5.1.2 Interface da Ferramenta

A ferramenta DECLASS é composta de uma única interface (Figura 5.3), que engloba os três módulos descritos anteriormente. É através desta interface que o autor constrói a árvore de classes de situações. A ferramenta implementa ainda, um mecanismo parcial de herança: os procedimentos criados para um determinado nó estão acessíveis para serem referenciados pelos seus correspondentes nós descendentes. Isto se dá porque a ferramenta DECLASS implementa, ainda que de forma insipiente, alguns dos fundamentos da programação orientada a objetos (OOP). A ferramenta DECLASS não dá suporte pleno à herança, não sendo possível através dela, a criação de uma classe que incorpore automaticamente todas as características de seus ancestrais. A ferramenta possibilita porém que o autor especialista especifique formalmente uma hierarquia de classes (através do posicionamento da classe dentro da árvore), indicando explicitamente através de chamadas de função, as características de suas classes ancestrais que deseja utilizar.

Tanto as regras de classificação quanto as regras pedagógicas são inseridas nas diversas classes através de um editor de texto próprio, disponível dentro da interface. Estas regras são compostas através da criação, pelo autor, de procedimentos e chamadas de função, a partir de primitivas previamente disponibilizadas e de uma linguagem aberta (*“open-ended”*), desenvolvida especialmente para este tutor. Durante o processo de criação dos diversos procedimentos que comporão as regras de classificação e pedagógicas, o autor possui também acesso aos procedimentos desenvolvidos em classes ancestrais e ao objeto *“board”* (vide Capítulo 4), através do qual pode manipular situações do tabuleiro.

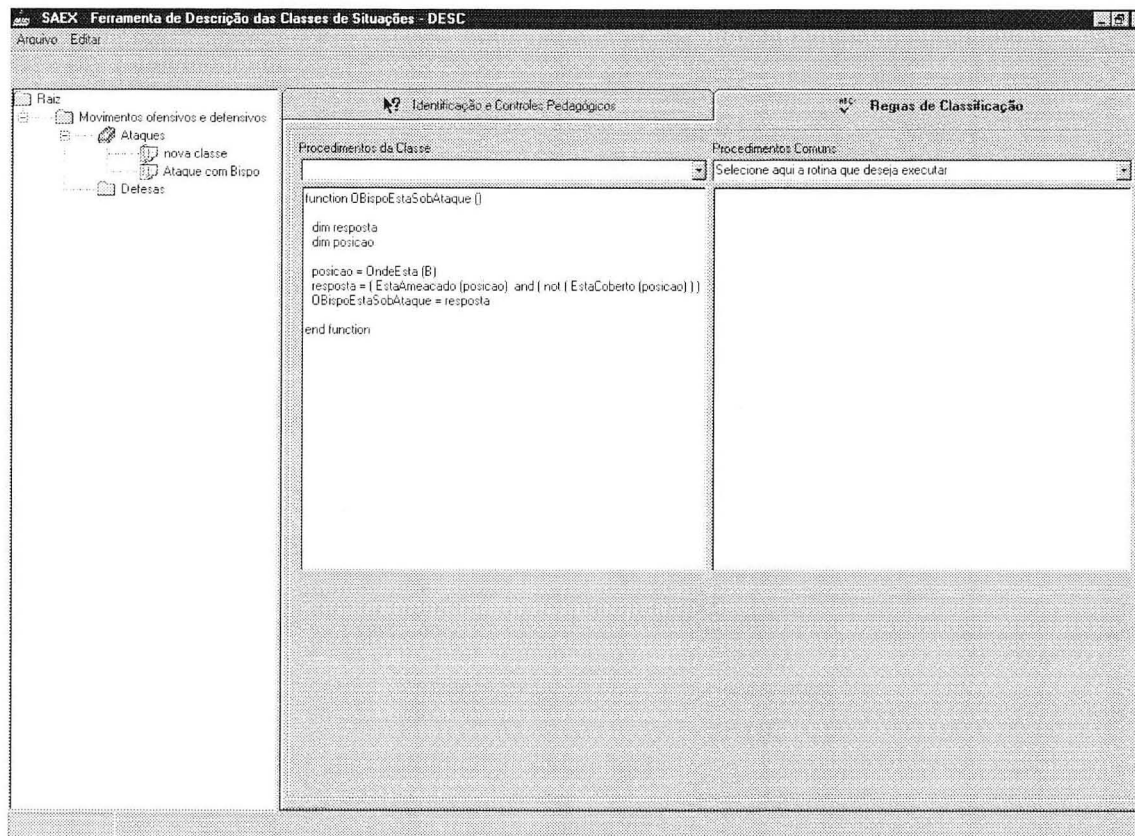


Figura 5.3 – A FERRAMENTA DECLASS

5.2 CLASSES DE SITUAÇÕES DO DOMÍNIO

Nesta seção analisaremos a forma pela qual o conjunto das classes de situações do Nível Conceitual é capaz de representar as características do processamento visual primário necessário para a formação de perícia no Jogo de Xadrez. São estas representações que permitem ao Classificador de PGN's, no Nível de Produção, selecionar os arranjos de tabuleiros que pertençam às respectivas classes de situações.

A construção do conjunto de classes de situações é uma tarefa delicada, pois cada classe tem que refletir um determinado detalhe da perícia que o especialista deseja transmitir. Através do uso da ferramenta DECLASS o processo de criação deste conjunto de classes dá-se sem a necessidade de uma ordem previamente estabelecida. Este aspecto modular da interação entre

o especialista e a ferramenta permite ao autor especialista criar e modificar facilmente as diferentes classes e descrições. Este processo de representação do conhecimento foi concebido para que o especialista possa descrever a sua perícia de uma forma estruturada, já durante a fase de modelagem das respectivas classes. Isto ajuda a aumentar o grau de legibilidade do conhecimento transferido para o tutor, permitindo inclusive que o mesmo seja facilmente inspecionado e complementado por outros especialistas.

Para permitir ao autor construir descrições visuais satisfatórias para o ensino de habilidades de reconhecimento visual, foi determinado que este conjunto de classes seria disposto em uma estrutura hierárquica (no caso presente, uma árvore). A árvore é uma estrutura hierárquica satisfatória para expressar a decomposição de características visuais em partes e sub-partes. A árvore provê os meios para: (a) permitir a visualização das características visuais em diferentes níveis de detalhamento, (b) criar uma ordem implícita para o tutor apresentar ao aprendiz cada uma das várias componentes da perícia, e (c) permitir ao autor especialista encapsular, de forma intuitiva, o seu conhecimento pericial dentro destas classes de situações componentes, na forma de características visuais e de operações pedagógicas específicas para o domínio em questão (Figura 5.3).

Para tornar este processo de autoria o mais transparente possível, esta estrutura foi concebida segundo os princípios da OOP. Isto permitiu desenvolver um ambiente dotado de um maior nível de abstração, no qual esconde-se do autor as informações que, ainda que necessárias para o funcionamento do tutor, não possuem relação direta com o conhecimento pericial a ser transmitido.

5.2.1 Composição de uma Classe de Situações

É através do conjunto de todas as classes de situações que o tutor é capaz de apresentar ao aprendiz os problemas e tarefas envolvidos na

aquisição de perícia no domínio do Jogo de Xadrez, segundo a visão do especialista. E cada uma das diferentes classes de situações é formada por um conjunto de descrições conceituais que representam de maneira formal, um componente pericial que o especialista deseja desenvolver no aprendiz. Por descrições conceituais entende-se neste trabalho como as características visuais que devem necessariamente existir ou não em uma determinada classe de situações de arranjos de tabuleiros, bem como as suas respectivas regras pedagógicas de apresentação.

Uma classe de situações deve descrever com precisão um conjunto de arranjos de tabuleiro, os quais apresentam entre si características visuais semelhantes e bem determinadas. Uma classe de situações apresenta, necessariamente, as seguintes partes:

1. Um nome;
2. A identificação das características visuais de interesse, através de regras específicas para o seu reconhecimento a partir de um arranjo de tabuleiro genérico;
3. A identificação das operações pedagógicas a serem desenvolvidas a partir de instruções dinâmicas.

A Figura 5.4 apresenta um exemplo de descrição das características visuais de uma classe de situações que serve para determinar se o jogador de posse das peças brancas está ou não sendo ameaçado por um ou mais bispos do adversário. As operações pedagógicas da classe descrita neste exemplo encontram-se na Seção 5.3.

Neste exemplo, criou-se uma função específica, denominada *Está_Sendo_Ameaçado*, que retornando *verdadeiro* ou *falso*, permite à classe *AtaqueDeBispos* armazenar ou descartar da base de dados do SAEX o arranjo em análise. Dentro da função *Está_Sendo_Ameaçado* criou-se uma variável denominada *Bispos*, que comporta-se como uma lista. Esta variável armazena-

```

Função Está_Sendo_Ameaçado ( Cor )

    Nova_Lista ListaDeBispos
    ListaDeBispos = OndeEstá( BISPOS, Oponente( Cor ))

    Se QuemÉAmeaçado( ListaDeBispos ) = VAZIO Então
        Está_Sendo_Ameaçado = FALSO
    Senão
        Está_Sendo_Ameaçado = VERDADEIRO
    Fim_Se

Fim_Função

Classe AtaqueDeBispos

    Se Está_Sendo_Ameaçado ( BRANCAS ) Então
        ArmazeneArranjo ( "Ataque de Bispos" )
    Senão
        DescarteArranjo
    Fim_Se

Fim_Classe

```

Figura 5.4 – UMA DESCRIÇÃO PARCIAL DAS REGRAS DE CLASSIFICAÇÃO DE UMA CLASSE DE SITUAÇÕES

rá as coordenadas do tabuleiro ocupadas pelos bispos do jogador indicado (no caso o oponente do jogador referenciado na chamada da função) ou permanecerá vazia, caso não haja nenhum. Através da utilização de três (3) primitivas (*OndeEstá()*, *Oponente()* e *QuemÉAmeaçado()*) esta função é capaz de detectar se uma determinada característica visual (a ameaça através de bispos) existe ou não em diferentes arranjos. Se o autor determinar que uma dada função pode ou deve ser utilizada em mais de uma classe de situações, o especialista pode incorporá-la na biblioteca de primitivas. A partir daí ela também passará a ser considerada pelo sistema como sendo uma primitiva, estando disponível a todas as classes de situações.

A criação de funções que serão responsáveis por determinar a existência ou não de determinadas características visuais dependem de alguns fatores específicos, tais como: (a) a anatomia do tabuleiro e (b) o formalismo

empregado (linguagem e primitivas) para enquadrar os arranjos dentro das classes de situações. Estes fatores estão descritos nas próximas Sub-seções.

5.2.2 Anatomia do Tabuleiro

O tabuleiro de Xadrez é constituído de uma malha quadriculada de 8x8, com 64 casas iguais, alternadamente claras e escuras. O tabuleiro de Xadrez é colocado entre os jogadores de forma a deixar a casa à direita de cada jogador de cor mais clara. O tabuleiro é formado por 3 linhas retas: (a) as colunas, (b) as fileiras e (c) as diagonais. As colunas (verticais) são designadas a partir da esquerda do jogador de posse das peças brancas, pelas letras minúsculas a, b, c, d, e, f, g, h. Já as fileiras (horizontais), também a partir do jogador de posse das peças brancas, são numeradas de 1 a 8. As diagonais também possuem denominação própria.

O tabuleiro de Xadrez comporta até 32 peças (16 brancas e 16 pretas), de 6 tipos distintos, a saber: (a) as Torres, que ficam nas extremidades da primeira fila, tendo ao seu lado (b) os Cavalos. Aos Cavalos seguem-se os (c) Bispos. No centro posicionam-se (d) o Rei e (e) a Dama. O Rei fica sempre na casa oposta à sua cor, isto é, o Rei branco se posiciona na casa preta. A Figura 5.5 apresenta o tabuleiro em seu arranjo com a disposição inicial. Cada tipo de peça se movimenta de uma maneira diferente, entretanto, sem distinção de cor. O início de uma partida sempre se dá com as peças brancas.

Para a determinação das coordenadas das diferentes peças dentro de um tabuleiro o ambiente SAEX utiliza o sistema algébrico inglês. Este sistema, utilizado para anotar partidas de Xadrez, é hoje o mais popular em todo mundo e assemelha-se ao utilizado no conhecido jogo de batalha naval. Neste sistema, o tabuleiro é referenciado através de um par ordenado (letra – número). Na anotação dos lances a letra maiúscula indica a peça (T – Torre, B – Bispo, N - Cavalo, K – Rei e Q – Dama) que se moveu (com exceção do peão, que não é representado por nenhuma letra). Neste tutor, quando neces-

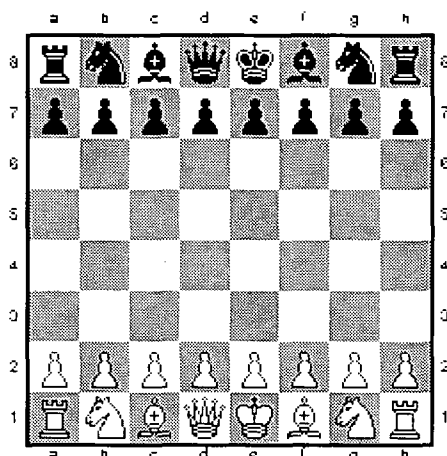


Figura 5.5 – A DISPOSIÇÃO INICIAL DO TABULEIRO DE XADREZ

sário, o peão pode ser explicitamente referenciado pela letra P. O par letra - número representa o ponto para onde a referida peça vai se mover. A Figura 5.6 indica todas as coordenadas existentes em um tabuleiro de Xadrez. Outros símbolos também são utilizados, para indicar a “qualidade” de um determinado lance (Tabela 5.1). Por exemplo, 7. Nxf2 Qb3+ indica que no sétimo lance das brancas o Cavalo saltou para “f2”, capturando a peça que lá se encontrava. Em resposta, as pretas deslocaram a dama para “b3”, com xeque. Nas jogadas de Peão anota-se apenas a casa de chegada. Quando um Peão efetua uma captura, indica-se apenas a sua coluna original e o destino onde vai capturar (11. exf5, por exemplo). No caso de duas peças idênticas poderem atingir a mesma casa (por exemplo o caso de dois Cavalos em “c6” e “f7” para a jogada Ne5), para não haver dúvidas sobre qual delas se moveu, a anotação contém

Peças pretas							
a8	B8	C8	d8	e8	f8	g8	h8
a7	B7	C7	d7	e7	f7	g7	h7
a6	B6	C6	d6	e6	f6	g6	h6
a5	B5	C5	d5	e5	f5	g5	h5
a4	B4	C4	d4	e4	f4	g4	h4
a3	B3	C3	d3	e3	f3	g3	h3
a2	B2	C2	d2	e2	f2	g2	h2
a1	B1	C1	d1	e1	f1	g1	h1
Peças brancas							

Figura 5.6 – O TABULEIRO DE XADREZ E SUAS COORDENADAS

Símbolos Utilizados	
!	Bom lance
!!	Lance excelente
?	Mau lance
??	Erro grave
!?	Lance interessante
?!	Lance de valor duvidoso
+	Xeque
++	Xeque – mate
O-O	Roque menor
O-O-O	Roque maior
X	Captura
1 : 0	Vitoria da brancas
0 : 1	Vitoria das pretas
½ : ½	Empate

Tabela 5.1 – OS SÍMBOLOS UTILIZADOS NO SISTEMA ALGÉBRICO PARA INDICAR A "QUALIDADE DE UMA JOGADA"

uma referência da casa inicial (5.Nce5 para o "Nf7"). Se as duas peças estão na mesma coluna o diferenciador será a fileira de onde sai a peça.

No decorrer da partida poderão ocorrer lances os quais podem receber denominações especiais, e que nos convêm aqui destacar. São eles: (a) o Xeque, (b) o Roque e (c) a Promoção do peão. Além dos lances, outros tipos de jogadas podem ocorrer durante uma partida, e devido à sua intencionalidade, também acabam por receber denominações próprias. Como exemplo, citamos três tipos de defesas: (a) a Fuga, (b) a Cobertura e (c) a Captura.

Durante o processo de criação das diferentes classes de situações, estes elementos anatômicos de uma partida de Xadrez devem ser considerados e utilizados para referenciar as características visuais desejadas. A forma como se dá esta utilização está descrita na Sub-Seção 5.2.3.

5.2.3 A construção das Regras e as Primitivas de Enquadramento

Para representar os diferentes tipos de lance ou de comportamentos possíveis para as peças presentes foi necessário dotar a ferramenta do Nível

Conceitual de uma linguagem de alto nível para a construção das regras, um interpretador para a mesma e de uma biblioteca de funções, as quais foram denominadas “primitivas”. Isto foi necessário para permitir ao autor especialista, ao criar as regras de classificação, “traduzir” os conceitos visuais que deseja exercitar para um formato que o tutor possa manipular e compreender.

Para este processo de tradução foram analisados diferentes tipos de soluções: (a) o desenvolvimento de uma nova linguagem especificamente construída com a finalidade de facilitar a programação das regras, (b) estender uma linguagem de programação já existente, provendo funcionalidades que dessem suporte ao desenvolvimento destas mesmas regras e (c) prover o tutor de um compilador ou interpretador de uma linguagem convencional, fornecendo, na forma de uma biblioteca, funções que facilitem o desenvolvimento destas regras.

Em favor da primeira opção, tem-se o fato de, ao se desenvolver uma nova linguagem, poder-se esconder, de forma mais efetiva a, complexidade inerente à interpretação visual dos arranjos de tabuleiros, oferecendo funcionalidades de mais alto nível. Em contraposição a isto, pode-se argumentar que as demais opções possuem a vantagem de normalmente utilizarem como base linguagens de programação bastante conhecidas e que permitem ao autor especialista, caso já as domine, assimilar mais rapidamente o processo de construção das regras. Já a segunda solução resultaria em uma alteração sintática da linguagem escolhida de forma a prover novas construções específicas para o suporte às regras específicas do Xadrez. Isto implicaria na alteração do compilador da linguagem ou na construção de um pré-processador que resolveria as construções específicas introduzidas.

A abordagem escolhida neste trabalho é uma variante destas três opções: a ferramenta do Nível Conceitual permite que o autor utilize uma linguagem criada especialmente para este tutor, denominada ChessCode. Além desta ferramenta prover um interpretador adequado para esta linguagem,

o sistema disponibiliza ainda uma biblioteca de funções já pré-programadas, denominadas primitivas. O ambiente do tutor é composto desta biblioteca e de um núcleo que dá suporte à linguagem ChessCode e às regras desenvolvidas em tempo de execução pelo autor.

As primitivas previamente desenvolvidas estão divididas em três grupos: (a) o grupo de primitivas booleanas (que retornam como resultado apenas duas variações: Verdadeiro ou Falso), (b) o grupo de primitivas posicionais (que retornam coordenadas de um tabuleiro de Xadrez) e (c) o grupo de primitivas identificadoras (que retornam o tipo de peça presente em determinada coordenada do tabuleiro). A tabela 5.2 discrimina cada um destes grupos. O autor especialista tem total liberdade para utilizá-las ou não, modificá-las, estendê-las ou ainda de criar novas primitivas. É a construção das regras através da linguagem ChessCode e utilização das primitivas que possibilitam a seleção ou descarte dos arranjos armazenados no formato PGN, a partir da interpretação visual dos mesmos.

Este conjunto de primitivas surgiu do interesse do ambiente em “esconder” do autor especialista o mecanismo utilizado pela linguagem ChessCode para extrair do arranjo de tabuleiro em análise as suas características visuais e o processo de armazenamento e descarte do mesmo. A extração das características visuais do arranjo é possível através do objeto *board*, descrito nos capítulos 3 e 4. Já o armazenamento e descarte de tabuleiros é realizado pelo Classificador de PGN's quando da análise dos PGN's, através de objetos internos à ferramenta do Nível de Produção.

5.3 REPRESENTAÇÃO PEDAGÓGICA DAS CLASSES DE SITUAÇÕES

Na Seção anterior analisamos a forma pela qual as classes de situações do Nível Conceitual são capazes de descrever características visuais distintas de um arranjo do Jogo de Xadrez de forma tal a permitir ao Classifica-

Tipo	Nome	Finalidade
Booleanas	EstáEmXeque(X)	Retorna Verdadeiro ou Falso se o Rei indicado pela coordenada X está ou não em xeque
	EstáProtegida(X)	Retorna Verdadeiro ou Falso se a peça indicada por X está protegida ou não por outra peça de mesma cor
	EstáProtegendo(X)	Retorna Verdadeiro ou Falso se a peça indicada pela coordenada X está ou não protegendo outra peça de mesma cor
	EstáAmeaçando(X)	Retorna Verdadeiro ou Falso se a peça indicada pela coordenada X pode ou não capturar uma peça inimiga
	EstáAmeaçada(X)	Retorna Verdadeiro ou Falso se a peça indicada por pela coordenada pode ou não ser capturada por uma peça inimiga
Posicionais	QuemÉ(X)	Retorna a peça indicada pela coordenada X
	OndeEstá(X, COR)	Retorna a coordenada da peça indicada por X, da cor indicada por COR
Identificadoras	QuemProtege(X)	Retorna a coordenada da peça que está protegendo a peça indicada pela coordenada X
	QuemEstáProtegida(X)	Retorna a coordenada da peça que está sendo protegida pela coordenada X
	QuemAmeaça(X)	Retorna a coordenada da peça que está ameaçando a peça indicada pela coordenada X
	QuemÉAmeaçado(X)	Retorna a coordenada da peça que está ameaçada pela peça indicada pela coordenada X

Tabela 5.2 – ALGUMAS DAS PRIMITIVAS DISPONIBILIZADAS PELA FERRAMENTA DECLASS

dor de PGN's selecionar e classificar os diferentes arranjos segundo estas características.

Nesta seção analisaremos como o ambiente SAEX utiliza estas mesmas classes de situações para instruir a *shell* de ensino quanto a como apresentar estas características visuais a partir dos diferentes arranjos de tabuleiro selecionados (vide Capítulo 3).

5.3.1 Regras Pedagógicas

As regras de classificação de arranjos são antes de tudo, genéricas. Quaisquer arranjos, com as mais diversas configurações de peças, que satisfaçam as regras definidas para uma determinada classe serão armazenadas. Entretanto, como o tutor pode ressaltar e explicar estas mesmas características, sendo os diferentes arranjos tão diversos? – Através das regras pedagógicas. A definição da forma como a *shell* de ensino deverá transmitir o

conhecimento pericial através de um determinado arranjo de tabuleiro constitui uma tarefa crucial. No nível instrucional, o aprendiz somente é capaz de perceber o que o autor deseja transmitir, através da visualização do arranjo que lhe é apresentado e das operações pedagógicas a que este arranjo e o próprio ambiente da *shell* são submetidos.

Uma operação pedagógica nada mais é do que uma ação que o objeto *board* está capacitado a executar dentro da *shell* (vide capítulo 3). As regras pedagógicas, de forma semelhante às regras de classificação, também são construídas através da linguagem ChessCode e de primitivas. E estas primitivas também estão associadas com o objeto *board*, descrito com maiores detalhes nos Capítulos 3 e 4.

A Figura 5.8 apresenta um exemplo de regra pedagógica de apresentação. Dado um determinado arranjo, quando a *shell* de ensino aplica regra pedagógica inserida na função TrocaDePeças, recebe como resposta as operações apresentadas na Figura 5.7. O resultado, visualizado pelo aprendiz através da *shell* de ensino é então o arranjo, tal como apresentado na Figura 5.9.

Através destas regras o autor pode transmitir o seu conhecimento pericial com o aprendiz, independentemente de qual será o arranjo selecionado pela *shell*. O aprendiz poderá inclusive verificar através de diferentes arranjos e da aplicação das mesmas regras qual efetivamente é a intenção do autor.

```
Board.Arrow.Coordinate e4, c6  
Board.Arrow.Show  
Board.Comment.Text 4, "o CAVALO da coordenada c3 está  
protegido pela TORRE da coordenada a3. Sem uma estratégia  
específica, isto resultará apenas em uma troca de peças."
```

Figura 5.7 – EXEMPLO DE INSTRUÇÕES APLICÁVEIS AO OBJETO *BOARD*

```

Função TrocaDePeças (Cor)
  Nova_Variável Texto
  Nova_Variável Peça1
  Nova_Variável Peça2
  Nova_Lista Coordenada1
  Nova_Variável Coordenada2

  Coordenada1 = OndeEstá (BISPO, Cor)
  Se EstáAmeaçando (Coordenada1) Então
    Coordenada2 = QuemÉAmeaçado (Coordenada1)
    Peça2 = QuemÉ (Coordenada2)
    Texto = "Board.Comment.Text " & Linha & ", Article ( Peça1 ) & " " &
    Peça1 & " da coordenada " & Coordenada1 & " está protegido pel" &
    Article ( Peça2 ) & " " & Peça2 & " da coordenada " & Coordenada2 &
    ". Sem uma estratégia específica, isto resultará apenas em uma troca
    de peças."
    Shell.Execute = "Board.Arrow.Coordinate " & Coordenada2 & ", " &
    Coordenada1
    Shell.Execute = "Board.Arrow.Show"

    Se Linha (Coordenada1) <> Linha (Coordenada2) Então
      Shell.Execute = "Board.Comment.Text " &
      Linha(Coordenada1) & ", " & Texto
    Senão
      Shell.Execute = "Board.Comment.Text " &
      Coluna(Coordenada1) & ", " & Texto
    Fim_Se
  Fim_Se
Fim_Função

```

Figura 5.8 – EXEMPLO DE REGRA PEDAGÓGICA

Há várias vantagens em utilizar regras para a definição de operações pedagógicas específicas para cada uma das classes de situações. Primeiramente, as regras pedagógicas têm um padrão de sintaxe claro e bastante pouco exigente para classificar o conhecimento. Isto pode ser particularmente importante quando o autor possui pouca habilidade de programação. Finalmente, aproximações alternativas, como metas e planos, provavelmente exigiriam esforço redundante alcançar o mesmo grau de especialização de contexto. Este mecanismo de regras pedagógicas não permite ao ambiente SAEX efetivamente interagir com o aprendiz, tal como o faz o sistema RUI (Direne, 1993), através das suas “*default teaching actions*”.

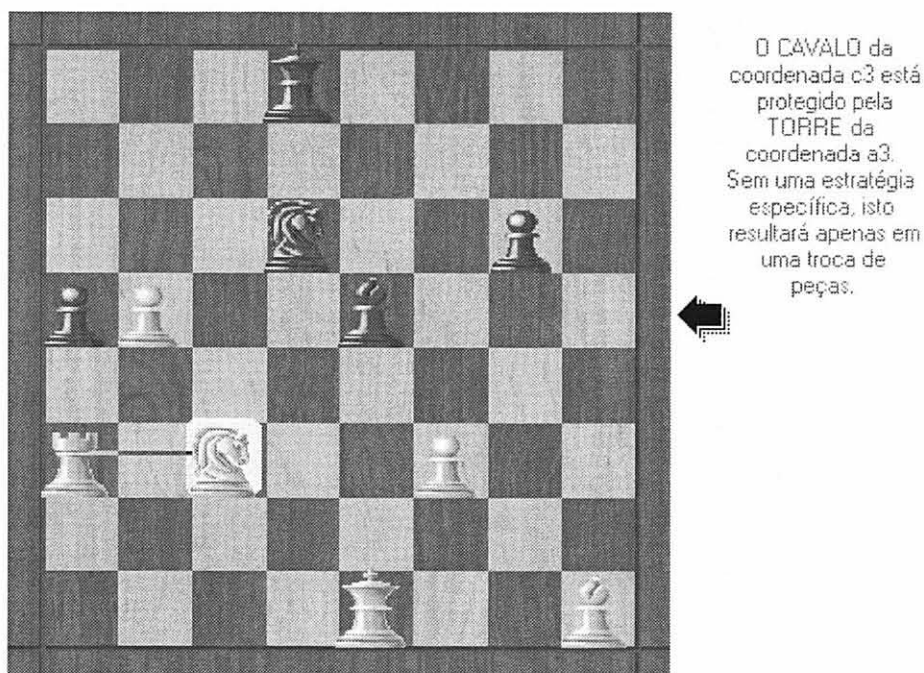


Figura 5.9 – O RESULTADO DA APLICAÇÃO DA REGRA PEDAGÓGICA DA FIGURA 5.7

O sistema RUI implementa um mecanismo de diálogo, que entre outras características, possui uma espécie de “agenda”. Através dela o tutor pode “colocar à prova” a capacidade do aprendiz em efetivamente discriminar e descrever aspectos visuais das imagens a ele apresentadas.

O sistema SAEX não apresenta mecanismo equivalente. A linguagem ChessCode é potente o suficiente para implementar soluções parciais para este tipo de deficiência, entretanto acreditamos que em futuro próximo será interessante dotar a ferramenta do Nível Conceitual do SAEX também de um mecanismo equivalente de regras. Isto permitirá ao SAEX se tornar mais adaptável ao perfil do aprendiz, ao mesmo tempo que desobrigará o autor especialista a explicitamente descrever as ações a serem executadas pela *shell* do Nível Instrucional.

CAPÍTULO 6

CONCLUSÃO

Conforme verificado no Capítulo 2, o único exemplo de sistema tutor para o ensino de estratégias de jogos publicado que se tem notícia é o QUIZ. Entretanto nem ele é capaz de abordar a componente visual necessária para a formação de perícia para o Jogo de Xadrez. Os poucos protótipos de sistemas tutores inteligentes conhecidos para jogos em geral são versões reduzidas e requerem ainda pesquisas a respeito de formas mais precisas para a representação de jogadas e de extensos catálogos de posições construídas e testadas em ambientes realistas da prática dos jogos. É necessário que mais estudos sobre o processo de ensino-aprendizagem de conceitos visuais, sobretudo voltados para os jogos de estratégia, sejam levados a bom termo.

6.1 A CONTRIBUIÇÃO

O ambiente SAEX, desenvolvido e descrito neste trabalho é a primeira tentativa que se tem conhecimento, de construir-se um ITS específico para o ensino de conceitos visuais para o Jogo de Xadrez. Por utilizar uma base de dados baseada quase que totalmente em arranjos de jogos reais, e por ser capaz de representar internamente as características visuais de cada um destes arranjos, o ambiente SAEX supre a lacuna até então não coberta pelos protótipos e sistemas comerciais existentes.

Ainda que a ferramenta do Nível Instrucional apresente mecanismos primários de interação com o aprendiz, pode-se desde já verificar que este ambiente possui boas possibilidades de utilização e de aprimoramento, justamente por abordar o aspecto iminentemente visual da formação de perícia. Ao utilizar o método pedagógico da descoberta-guiada, o ambiente permite que

o aprendiz tanto possa pesquisar a base de dados do sistema quanto assistir a uma sessão de ensino. Este processo dinâmico de interação permite que o aprendiz dite suas próprias regras de conduta que, respeitadas pela ferramenta do Nível Instrucional, acreditamos propiciam um maior conforto e eficácia na utilização do ambiente.

Acreditamos ser no processo de editoração onde está a maior contribuição deste trabalho. Através dos Níveis Conceitual e de Produção o autor pode idealizar e implementar um sem número de cursos, cada qual com diferentes níveis de especialização. Através da identificação das características visuais a serem estudadas e de suas respectivas regras de classificação e ordenação, torna-se possível quantificar o potencial que cada classe e cada exemplo tem em exercitar o aprendiz em uma determinada capacidade que ele deve desenvolver para se tornar perito. Isto propicia, entre outras coisas, à individuação do ensino, ao permitir a aplicação da estratégia pedagógica que mais se adapte ao aprendiz e permitem mudanças para corrigir falhas do aluno.

Devido a inexistência de outros trabalhos publicados na área, todas as ferramentas, linguagens de criação de regras de classificação e de regras pedagógicas descritas neste trabalho são originais. A criação de funções “primitivas”, tanto para as regras de classificação quanto para a de apresentação (regras pedagógicas), aliada à possibilidade de carregar-se a base de conhecimento do ambiente com os registros de partidas reais são as principais contribuições deste trabalho. Devido à originalidade dos conceitos desenvolvidos, os mesmos tiveram de necessariamente, serem testados, o que foi realizado através da implementação de cada uma das ferramentas. Cada ferramenta foi desenvolvida separadamente, mas tendo uma base de dados comum. No momento estas ferramentas estão em fase final de integração, o que resultará efetivamente em um ambiente único de editoração e ensino.

6.2 TRABALHOS FUTUROS

No momento presente, estão sendo realizados contatos com instituições que se prestam ao ensino do Jogo de Xadrez, para *in loco*, validarmos os conceitos e o ambiente desenvolvido neste trabalho. Esta etapa é necessária para o aprimoramento, principalmente da ferramenta do Nível Instrucional.

Como trabalho futuro, propomos o projeto e implementação de um modelo do aprendiz, para que a ferramenta do Nível Instrucional possa também levar em conta o grau de desenvolvimento e particularidades do aprendiz, durante a sessão de ensino. A interseção entre os modelos do aprendiz e o tutorial é muito grande num ITS. Um modelo tutorial precisa das informações que o modelo do aprendiz obtém sobre o desempenho e, até mesmo, sobre o estado motivacional do aprendiz.

REFERÊNCIAS BIBLIOGRÁFICAS

- ANDERSON, J. R. (1983). *The Architecture of Cognition*. Harvard University Press, Cambridge, Massachusetts.
- AUSUBEL, D. P. (1968) *Educational Psychology: A Cognitive View* Holt, Rinehart and Winston, New York, NY
- BAUDET, G. M. (1978) On the Branching Factor of the Alpha-Beta Pruning Algorithm, *Artificial Intelligence*, Vol. 10, nº 2.
- BEILER, A., STRUBE DE LIMA, V. (1996) Permitir ao leitor desempenhar o papel de criador de significado: autoria e aprendizado numa interface de hipertexto. *VII Simpósio de Informática na Educação*. Belo Horizonte.
- BERLINER, H. J. (1977) Search and Knowledge, *In Proc. IJCAI 5, 1977*.
- BINDER, F.; DIRENE, Alexandre I. (1999). Conceitos e ferramentas para apoiar o ensino de lógica de programação imperativa. *Anais do X Simpósio Brasileiro de Informática na Educação – SBIE 99*. Curitiba - PR
- BLESSING, S. B. (1997). A programming by demonstration authoring tool for model-tracing tutors. *International Journal of Artificial Intelligence in Education*, (8):223-261.
- BRUNER, J. S. (1968). *Toward a Theory of Instruction*. New York: W. W. Norton & Company, Inc.
- DIRENE, A.I. (1993). *Methodology and tools for designing concept tutoring systems*. Brighton: Doctor's thesis, School of Cognitive and Computing Sciences - Univ. of Sussex.
- DIRENE, A. I. (1997). Designing intelligent systems for teaching visual concepts. *International Journal of Artificial Intelligence in Education*, 8(1):44-70.

- FREIRE, M. E. P.; NUNES, M. G. V. (1997) SIATE: Sistema Inteligente de Apoio ao Treinamento e Ensino: Proposta de uma arquitetura. *I Encontro Nacional de Inteligência Artificial (ENIA)*, Brasília, p. 194-199.
- GAGNÉ, R. M. (1985) *The Conditions of Learning and Theory of Instruction*. Fourth Edition Holt, Rinehart and Winston, New York, NY
- LABAT, J. (1991). QUIZ, a Distributed Intelligent System. *International Conference on the Learning Sciences*, AACE, Evaston.
- LABAT, J. (1992). QUIZ, a Distributed Intelligent Tutoring System. *Proceedings of the International Conference Computer Aided Learning*.
- LABAT, (1993). Dynamic and Interactive Pedagogical Planning, The QUIZ Example. *Proceedings of AI-ED 93. World Conference on Artificial Intelligence in Education*.
- LESGOLD, A. M. (1984). Acquiring expertise. In Anderson, J. R. and Kosslyn, S. M., editors, *Tutorials in Learning and Memory: Essays in Honor of Gordon Bower*. W. H. Freeman.
- LESGOLD, A. M., Rubinson, H., Glasser, P. F. R., Klopfer, D., and Wang, Y. (1989). Expertise in a complex skill: Diagnosing x-ray pictures. In Chi, M., Glasser, R., and Farr, M., editors, *The Nature of Expertise*. Lawrence Erlbaum.
- MERVIS, C. B., & ROSCH, E. (1981). Categorisation and Natural Objects. *Annual Review of Psychology*, 32, 89-115.
- MURRAY, T. (1999). Authoring intelligent tutoring systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education*, (10):98-129.
- NASCIMENTO, Elisa; Direne, Alexandre I. (1997). Uma shell inteligente para apoiar o treinamento da programação de dispositivos digitais. *Anais do Workshop de Sistemas de Tutoria Inteligente aplicados à Educação e Treinamento do SBIE'97*. São José dos Campos -SP

- NWANA, H. S. (1990) Intelligent Tutoring Systems: An Overview. *Artificial Intelligence Review* (1990) 4, 251-277
- PEARL, J. (1982) The solution for the Branching Factor of the Alpha-Beta Pruning Algorithms and its the Optimality, *Communications of the ACM*, Vol. 25, nº 8, p. 559-564.
- PFROMM NETO, S. (1987). Psicologia da Aprendizagem e do ensino. Ed. *Pedagógica e Universitária – Universidade de São Paulo*.
- PIMENTEL, A., Direne, Alexandre I. (1998). Cognitive Measures for Visual Concept Teaching with Intelligent Tutoring Systems. *Anais da International Conference on Intelligent Tutoring Systems (ITS'98)*, San Antonio - USA
- RICH, E., Knight, K. (1993). Artificial Intelligence, *Mc Graham-Hill*
- SANTOS, G. H. R.; VIEIRA, F. M. C.; HASEGAWA, R.; NUNES, M. G. V. (1997) SASHE: A autoria de aplicações hypermídia para o ensino. *Simpósio Brasileiro de Informática na Educação*, ITA, São José dos Campos, p. 425-440.
- SHARPLES, M. (1991). Computer-based tutoring of visual concepts: from novice to expert. *Journal of Computer Assisted Learning*, 7:123-132.
- SHARPLES, M., JEFFERY, N., TEATHER, B., and du BULAY, G. (1997). A socio-cognitive engineering approach to the development of a knowledge-based training system for neuroradiology. In du Boulay, B. and Mizoguchi, R., editors, *Artificial Intelligence in Education*, pages 402-409. IOS Press.
- TEATHER, D., TEATHER, B. A., WILLS, K. M., du BOULAY, G. H., et al (1988). Evaluation of computer advisor in the interpretation of CT images of the head. *Neuroradiology*, 30, 511-577.