

ALEXANDRE RÔMOLO MOREIRA FEITOSA

**DEFINIÇÃO FORMAL DE TÁTICAS DE XADREZ POR MEIO DA
AUTORIA INCREMENTAL DE CONCEITOS HEURÍSTICOS**

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

Orientador: Prof. Dr. Alexandre Ibrahim Direne

CURITIBA

2006

ALEXANDRE RÔMOLO MOREIRA FEITOSA

**DEFINIÇÃO FORMAL DE TÁTICAS DE XADREZ POR MEIO DA
AUTORIA INCREMENTAL DE CONCEITOS HEURÍSTICOS**

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:

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

Prof. Dr. André Luiz Pires Guedes – DINF/UFPR – Membro Interno

Prof. Dr. Fabiano Silva – DINF/UFPR - Suplente

Prof. Dr. Luiz Arthur Pagani - DELIN/UFPR – Membro Externo

Curitiba, 22 de Setembro de 2006.

AGRADECIMENTOS

Muitas pessoas me ajudaram na realização deste trabalho. Dando-me apoio, conforto, e incentivo para superar todos os obstáculos que apareceram, no mestrado e na minha vida, durante este período conturbado. Mas ao final de tudo, depois da tormenta, e da batalha, posso me sentir aliviado, realizado e vitorioso. Agradeço a todos, verdadeiramente, do fundo do meu coração.

Minha memória fraca e seletiva, e o limite de uma página me impedem de colocar o nome de cada pessoa que me ajudou, mas mesmo assim, faço questão de citar e agradecer algumas delas neste momento.

Primeiro agradeço a Deus, meu grande parceiro, que sempre iluminou o meu caminho, sempre me deu esperança, e sentido ao que faço. Espero poder usar mais este conhecimento que recebi para contribuir na melhora da vida das pessoas de bem, mesmo que seja com um pouco.

Agradeço aos meus pais, José e Elizabeth, aos meus irmãos Alessandra e Emerson, e a minha querida vó Tereza. Pois no seio desta minha louca, agitada e Grande Família sempre tenho um refúgio, um porto seguro, repleto de apoio, carinho, e amor.

A todos os meus companheiros de trabalho da Sercomtel e da Unopar, o meu muito obrigado. Hoje considero todos meus grandes amigos. A força de vocês foi essencial. Deixo um agradecimento especial para o Sebastião Custódio, o Roberto Nishimura, e a Iolanda Sanches. Sem a compreensão deles nesta reta final este trabalho não seria finalizado.

Agradeço especialmente a Marília Amaral, a Viviane Prado, a Lia & Luis Patricio, a Cristiane & Eidy Guandeline, e a Tatiana & Fernando Serenato. Vocês são demais!

E por último, deixo para falar do meu amigo Alexandre Direne, o meu grande orientador. Devo e dedico a ele este trabalho. Pois a paciência, competência, compreensão, e obstinação dele me fizeram prosseguir e concretizar este estudo. Nunca conheci um professor tão comprometido com o ensino e pesquisa quanto ele. A todos o meu muito obrigado! E a você uma boa leitura.

SUMÁRIO

| | |
|---|------|
| LISTA DE FIGURAS | v |
| LISTA DE TABELAS | vi |
| LISTA DE ABREVIATURAS | vii |
| RESUMO | viii |
| ABSTRACT | ix |
| 1 INTRODUÇÃO..... | 1 |
| 1.1 Definição do Problema..... | 1 |
| 1.2 Objetivos do Trabalho..... | 3 |
| 1.2.1 Geral..... | 3 |
| 1.2.2 Secundário..... | 3 |
| 1.2.3 Específico..... | 4 |
| 1.3 Contexto do Projeto..... | 5 |
| 1.4 Estrutura da Dissertação..... | 6 |
| 2 TRABALHOS CORRELATOS..... | 7 |
| 2.1 Representação Externa..... | 7 |
| 2.1.1 Representação de Domínios de Conhecimento..... | 8 |
| 2.1.2 Representações Externas em Ambientes de Modelagem e Colaboração..... | 10 |
| 2.2 Uso de Aprendizagem Colaborativa em Aplicações de I.A. na Educação..... | 12 |
| 2.3 Uso de Jogos eletrônicos no processo de Ensino..... | 13 |
| 3 CONCEITOS E LINGUAGEM DE DEFINIÇÃO DE COMPONENTES HEURÍSTICOS..... | 15 |
| 3.1 Conceitos Fundamentais para Definição de Heurísticas..... | 15 |
| 3.1.1 Relação Material, Espaço e Tempo..... | 15 |
| 3.1.2 Valor inicial de peça..... | 17 |
| 3.1.3 Valor relativo de peça..... | 17 |
| 3.1.4 Identificação de Contextos..... | 18 |

| | | |
|---------|---|----|
| 3.1.4.1 | Quantitativo..... | 18 |
| 3.1.4.2 | Qualitativo..... | 19 |
| 3.1.4.3 | Posicional..... | 19 |
| 3.1.4.4 | Temporal..... | 20 |
| 3.1.4.5 | Comportamental..... | 21 |
| 3.1.5 | Simetria..... | 22 |
| 3.1.5.1 | Simetria de Jogo..... | 23 |
| 3.1.5.2 | Simetria Heurística..... | 23 |
| 3.1.5.3 | Simetria da Interpretação Heurística..... | 26 |
| 3.1.6 | Componentes Incrementais..... | 27 |
| 3.1.7 | Expressão de Cálculo Heurístico..... | 27 |
| 3.2 | Linguagem para Definição de Heurísticas de Jogos - DHJOG..... | 28 |
| 3.2.1 | Regras de Produção..... | 29 |
| 3.2.1.1 | HTE – Heurística de Transição de Etapa..... | 29 |
| 3.2.1.2 | HVP – Heurística de Valor de Peça..... | 30 |
| 3.2.1.3 | HVJ – Heurística de Valor de Jogo..... | 30 |
| 3.2.2 | Máquina de Estados Finito..... | 30 |
| 3.2.3 | Desenvolvimento Orientado a Funções..... | 32 |
| 3.2.4 | Primitivas Lógico Matemáticas..... | 34 |
| 3.2.4.1 | Aritmética..... | 34 |
| 3.2.4.2 | Geométricas..... | 34 |
| 3.2.4.3 | Lógicas..... | 35 |
| 3.2.4.4 | Algébricas..... | 35 |
| 3.2.5 | Nível de Expressividade..... | 36 |
| 3.2.6 | Apoio a Autoria e Colaboração..... | 38 |
| 3.2.7 | Fluxo de Execução de um Conjunto Heurístico..... | 38 |
| 3.2.8 | Limitações da Linguagem..... | 39 |
| 4 | ANÁLISE DO AMBIENTE DE ENSINO PROPOSTO - CACAREJE..... | 41 |
| 4.1 | Arquitetura Funcional..... | 42 |
| 4.1.1 | Aplicação Servidora..... | 42 |
| 4.1.1.1 | Base de Conhecimento Heurístico..... | 43 |
| 4.1.1.2 | CGDH – Comparador Genérico de Definições Heurísticas..... | 43 |
| 4.1.1.3 | EGPA – Escalonador Genérico de Partidas Automáticas..... | 44 |

| | |
|--|----|
| 4.1.2 Aplicação Cliente..... | 44 |
| 4.1.2.1 Ferramenta para Autoria de Heurísticas de Jogos..... | 45 |
| 4.1.2.2 Ferramenta para Definição de Situações de Jogo..... | 45 |
| 4.1.2.3 Repositório de Objetos Heurísticos..... | 46 |
| 4.1.2.4 Ferramentas de Comunicação entre Usuários..... | 46 |
| 4.1.2.5 Módulo para Criação/Participação em Competições Heurísticas..... | 46 |
| 4.1.2.6 Módulo para Controle da Execução de Partidas..... | 47 |
| 4.1.2.7 Módulo para Definição de Permissões de Acesso | 47 |
| 4.2 Exemplo de Cenário Completo de Utilização..... | 48 |
| 4.2.1 Descrição dos Atores..... | 48 |
| 4.2.2 Sequência de Ações..... | 49 |
| 4.3 Protótipo da Ferramenta de Autoria de Heurísticas..... | 51 |
| 4.3.1.1 Adaptação de Interface ao Nível de Conhecimento do Usuário..... | 52 |
| 4.3.1.2 Utilização de Facilitadores de Tarefas - Wizards..... | 54 |
| 4.3.1.3 Definição Visual de Regiões..... | 55 |
| 4.3.1.4 Agrupamento de Funções por Categoria..... | 58 |
| 4.3.1.5 Diagrama de Transição entre Etapas..... | 58 |
| 4.3.1.6 Geração Automática de Codificação Heurística..... | 60 |
| 4.3.1.7 Uso de Múltiplas Representações Externas..... | 62 |
| 4.3.1.8 Conceito de Caixa Preta e Caixa de Vidro integradas..... | 62 |
| 4.3.1.9 Combinação Automática de Componentes..... | 63 |
| 4.3.1.10 Limitações da Ferramenta de Autoria de Heurísticas..... | 64 |
| 4.4 Próximas etapas do desenvolvimento do sistema..... | 65 |
| | |
| 5 CONCLUSÃO E TRABALHOS FUTUROS..... | 66 |
| 5.1 Trabalhos Futuros..... | 67 |
| | |
| A ESPECIFICACAO DA LINGUAGEM <i>DHJOG</i> | 69 |
| | |
| B EXTENSÃO DA LINGUAGEM <i>DHJOG</i> PARA O JOGO DE XADREZ..... | 91 |
| | |
| REFERÊNCIAS BIBLIOGRÁFICAS..... | 96 |

LISTA DE FIGURAS

| | |
|---|----|
| Figura 2.1.1 – Sistema MER Calques 3D..... | 9 |
| Figura 2.1.2-1 – SimForest B – Exemplo de uso da técnica Caixa Preta..... | 11 |
| Figura 2.1.2-2 – SimForest G – Exemplo de uso da técnica Caixa de Vidro..... | 11 |
| | |
| Figura 3.1.3.3 – Exemplo de Definição de Centro de um Tabuleiro..... | 20 |
| Figura 3.1.3.5 – Exemplo de Tipos de Jogada Roque do Xadrez..... | 22 |
| Figura 3.1.4.2 -1 – Exemplo de Definição de Região para Proteger Roque..... | 24 |
| Figura 3.1.4.2-2 – Região Espelho de Proteção de Roque Pequeno..... | 25 |
| Figura 3.2.2 – Exemplo de Diagrama de Transição entre Etapas..... | 32 |
| Figura 3.2.4.4 – Exemplo de definição de Heurística de Valor de Jogo para Xadrez..... | 36 |
| | |
| Figura 4.1 – Arquitetura Cliente-Servidor do Sistema CACAREJE..... | 42 |
| Figura 4.3.1.1 – Nível de Complexidade de um Conjunto Heurístico..... | 53 |
| Figura 4.3.1.2 – Tela de criação de um novo Conjunto Heurístico..... | 55 |
| Figura 4.3.1.3-1 – Tela de Definição de Regiões..... | 56 |
| Figura 4.3.1.3-2 – Tela de Edição de Etapas e de Visualização de Regiões..... | 57 |
| Figura 4.3.1.5 – Diagrama de Transições entre Etapas..... | 59 |
| Figura 4.3.1.6-1 – Tela de Edição de Heurística - Aba principal..... | 60 |
| Figura 4.3.1.6-2 – Tela de Edição de Heurística – Aba de visualização de código gerado..... | 61 |
| | |
| Figura A – Diagrama de Pacotes da Linguagem DHJOG..... | 69 |
| Figura A.1 – Classes para Representação do Conhecimento Heurístico..... | 70 |
| Figura A.2 – Representação de uma Situação de Jogo..... | 77 |
| Figura A.3.9-1 – Formalização da DHJOG em BNF (Parte 1)..... | 89 |
| Figura A.3.9-2 – Formalização da DHJOG em BNF (Parte 2)..... | 90 |
| | |
| Figura B.1 – Elementos para a Representação de uma Situação de Jogo do Xadrez..... | 91 |
| Figura B.1.1 – Sistema Algébrico e Sistema Cartesiano..... | 92 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 3.2.5 – Níveis de Expressividade da Linguagem DHJOG | 37 |
| Tabela A.1 – Legenda de Cores Utilizadas..... | 70 |
| Tabela A.3.6-1 – Operadores Matemáticos..... | 83 |
| Tabela A.3.6-2 – Operadores Lógicos..... | 83 |
| Tabela A.3.6-3 – Operadores Relacionais..... | 83 |
| Tabela A.3.6-4 – Operadores de Atribuição..... | 83 |
| Tabela A.3.6-5 – Operadores de Precedência..... | 84 |
| Tabela A.3.6-6 – Operadores de Manipulação de Conjuntos..... | 84 |
| Tabela A.3.6-7 – Operadores de Manipulação de Textos..... | 84 |
| Tabela A.3.6-8 – Operadores para Definição de Comentários..... | 84 |

LISTA DE ABREVIATURAS

| | |
|---------------|--|
| IA..... | Inteligência Artificial |
| PROTEX..... | Projeto de Tipificação do Ensino de Xadrez |
| CACAREJE..... | Colaboração Alternada com Competição na Aprendizagem Referenciada por Jogos Educativos |
| DHJOG..... | Linguagem de definição de heurísticas de Jogos |
| RE..... | Representação Externa <i>(ER – External Representations)</i> |
| MER..... | <i>Multiple External Representations</i> (Sistemas com Múltiplas Representações Externas) |
| CAI..... | <i>Computer Aided Instruction</i> (Sistema de Instrução Assistida por Computador) |
| STI..... | Sistemas Tutoriais Inteligentes <i>(ITS – Intelligent Tutoring Systems)</i> |
| FIDE..... | <i>Fédération Internationale des Échecs</i> (Federação Internacional de Xadrez) |
| CEX..... | Centro de Excelência de Xadrez |
| HTE..... | Heurística de Transição de Etapa |
| HVP..... | Heurística de Valor de Peça |
| HVJ..... | Heurística de Valor de Jogo |
| LCH..... | Linguagem de Codificação de Heurísticas |

RESUMO

Neste trabalho são apresentados os resultados da pesquisa sobre a definição formal e incrementável de conhecimentos heurísticos de jogos. Este estudo serve de base para o projeto e desenvolvimento de um sistema de aprendizagem referenciada por jogos educativos, que defende a alternância da colaboração e da competição no processo de ensino, o sistema *CACAREJE*. Neste sistema o aprendiz irá formalizar conhecimento heurístico do jogo de Xadrez, inscrever as heurísticas definidas em competições automáticas entre heurísticas, acompanhar o desempenho delas nestes torneios, e trocar conhecimento formalizado com os demais aprendizes. Este sistema está inserido no projeto *PROTEX*, de apoio computacional ao ensino de Xadrez nas escolas brasileiras, apresentado em (Direne *et al.* 2004). A pesquisa deste trabalho foi conduzida em três etapas. Primeiramente, foi realizada uma análise crítica dos conceitos existentes e inerentes à formalização de conhecimentos heurísticos de jogos, incluindo fundamentos como vantagem material, temporal, espacial, e simetria, os quais foram cuidadosamente observados. Na segunda etapa foi especificada uma linguagem para definição de heurísticas de jogos, chamada *DHJOG*. Esta linguagem permite a formalização completa destes conhecimentos heurísticos, e o uso da especificação criada por um jogador automático para a simulação de partidas. Esta linguagem tem como base o uso de conceitos como Regras de Produção, Máquina de Estados Finitos, e o Desenvolvimento Orientado a Funções. Na última etapa foi realizado o projeto e implementação de um protótipo da Ferramenta de Autoria de Heurísticas. Para isto, inicialmente foram especificados os módulos das aplicações cliente e servidora do ambiente *CACAREJE*, às quais a ferramenta *DHJOG* será integrada. Esta tarefa foi executada em conjunto com Daniel Martineschen (2006). Posteriormente, foi feito um estudo sobre o uso de técnicas de Representação Externa do Conhecimento em ambientes de ensino. Esta análise serviu de base para o projeto da interface da aplicação, e visou diminuir o esforço cognitivo necessário ao uso da ferramenta. O desenvolvimento deste protótipo serviu como primeira validação do uso da linguagem *DHJOG*, e será a base para a implementação final desta aplicação.

ABSTRACT

This paper presents the research results about formal and incremental definition of gaming heuristics knowledge. This study is the design and development foundation of a learning system referenced by educational games which stimulates the alternate use of collaboration and competition in the educational process: the CACAREJE system. This system allows the user to formally describe Chess gaming heuristic knowledge, test the user defined heuristics in heuristics matches, analyze their performance in these matches and exchange formalized knowledge with other users. It is part of the *PROTEX* Project, presented in (Direne *et al.* 2004) which supports computer aided Chess teaching in brazilian schools. The research involved is split in three stages. Firstly, a critical analysis of the existing concepts intrinsic to gaming heuristics knowledge formalization, including material, temporal and spatial advantage, and also symmetry, took place. Secondly, a gaming heuristics definition language, called *DHJOG*, was proposed. This language allows de complete formalization of heuristics knowledge and then the use of these knowledge in an automated player that plays a simulated match. This language is based on Production Rules, Finite State Machines and Function Oriented Design. Lastly, an Heuristics Authoring Tool prototype was designed and developed. The design process started with the specification of the server and client modules for the *CACAREJE* environment in which the *DHJOG* tool will be integrated. This initial stage was done with Daniel Martineschen (2006). Later, an study about using techniques for External Representation of Knowledge in teaching environments was made and used in the application interface design to lower the cognitive effort required to use the tool. The application prototype development was the first validation of the *DHJOG* language and will be the foundation for the final version of the tool.

1 CAPÍTULO

INTRODUÇÃO

Neste capítulo apresentamos uma descrição geral da dissertação. Definimos o problema central abordado, especificamos os objetivos a alcançar, e descrevemos o contexto no qual este estudo está inserido. Ao final fazemos uma breve descrição dos tópicos abordados pelos demais capítulos deste trabalho.

1.1 Definição do Problema

O ensino de jogos é realizado normalmente através de aulas expositivas, onde o instrutor apresenta as regras do jogo e analisa estudos de casos (partidas), e depois os aprendizes são submetidos a competições para validarem as técnicas aprendidas. Podemos citar como exemplo o ensino de Xadrez. Neste processo o aluno é instigado a aprender a identificar, relacionar, e valorar os possíveis tipos de vantagens que podem ocorrer durante uma partida. Estas vantagens podem ser materiais, espaciais (posicionais), ou temporais (de desenvolvimento) (Veja Seção 3.1), são alcançadas pelo jogador através das jogadas realizadas. A adoção de táticas específicas pelo jogador pode acelerar a conquista de determinadas vantagens durante uma partida.

Podemos identificar vários problemas nesta abordagem de ensino. Primeiramente, muitos conceitos, vantagens e táticas são tratadas de forma totalmente subjetiva, e sem um formalismo claro. Por exemplo, no Xadrez existe a tática de dominar o centro do tabuleiro na abertura do jogo, mas não existe nenhum formalismo que deixe claro para um aprendiz iniciante quando uma partida está na abertura, quais casas constituem o centro do tabuleiro, e de que forma uma região pode ser considerada dominada.

Segundo, mesmo que o aprendiz analise registros de partidas onde tal técnica foi adotada, ele terá dificuldade em identificar os parâmetros que o jogador adotou mentalmente durante o jogo para executar tal tática. A perícia desta percepção de estratégias só é alcançada depois da análise exaustiva de várias partidas, e do uso da tática em si em diversas competições.

Terceiro, a validação do aprendizado apenas através da participação em competições é um processo impreciso pois, ao final apenas se constata a vitória ou derrota do aluno. Um instrutor poderá, acompanhando a partida, perceber quais técnicas o aprendiz está tentando utilizar, mas não conseguirá, de forma precisa, visualizar todos os parâmetros que ele adotou mentalmente. Fica difícil então saber qual o parâmetro que o aprendiz analisou de forma imprecisa, e retornar a ele uma correção de postura.

E por último, este processo de ensino não estimula o aprendiz a entender o conceito de heurísticas em si, e não o força a formalizar os processos mentais adotados por ele durante a execução das partidas.

Estes erros se devem em muito pela falta de mecanismos que permitam aos especialistas, instrutores, e aprendizes a formalizarem os seus conhecimento heurísticos de um jogo, e que facilitem a troca destes conhecimentos especificados entre a comunidade. Foi constatada a escassez de estudos científicos que discutam o processo de formalização destes conhecimentos heurísticos, e o desenvolvimento de ferramentas de apoio ao ensino destes conceitos. Não é uma tarefa simples tornar fácil a usuários leigos (sem conhecimentos de Inteligência Artificial) o uso e aprendizado destes conceitos.

Nos trabalhos que precederam o contexto deste trabalho (Schäfer 2000; Direne *et al.* 2004; Martineschen *et al.* 2006) são apresentadas ferramentas, conceitos e arquiteturas de sistemas que visam atender esta necessidade. Este presente trabalho vem a complementar estes estudos, e propor uma solução para a formalização de conhecimento heurístico de jogos, especificamente do Xadrez.

1.2 Objetivos do Trabalho

1.2.1 Geral

O objetivo principal deste estudo é criar conceitos e mecanismos para a definição formal e incrementável de conhecimentos heurísticos de jogos. As ferramentas definidas deverão propiciar que pessoas leigas em conceitos de Inteligência Artificial possam iniciar seus estudos sobre o uso e a formalização de heurísticas. Na literatura científica existe uma escassez de trabalhos que abordam este tema. Este estudo também servirá de base para o projeto e desenvolvimento de um sistema de aprendizagem referenciada por jogos educativos, que defende a alternância da colaboração e da competição no processo de ensino. Neste sistema o aprendiz irá formalizar conhecimento heurístico de um jogo, inscrever as heurísticas definidas em competições automáticas entre heurísticas, acompanhar o desempenho delas nestes torneios, e trocar conhecimento formalizado com os demais aprendizes para melhorar sua heurística, redefinindo-a e entrando em novas competições para dar continuidade e novos ciclos.

1.2.2 Secundário

Podemos definir como objetivo secundário o desenvolvimento de uma linguagem de formalização de conhecimentos heurísticos de jogos. Esta linguagem deverá possibilitar a especificação de heurísticas, e permitir que as mesmas sejam: incrementadas, analisadas, comentadas, pelo próprio autor e por outros, e utilizadas por um jogador automático para a simulação de partidas entre heurísticas.

A linguagem será validada através da prototipagem de uma ferramenta de Autoria de

Heurística. Esta aplicação deverá suportar a linguagem de formalização, e utilizar técnicas de representação externa para diminuir o esforço cognitivo necessário ao usuário na realização das tarefas. Este protótipo ainda servirá como base para a implementação da versão definitiva desta ferramenta, que fará parte do ambiente de ensino *CACAREJE*, e estará integrada ao servidor do CEX (Centro de Excelência de Xadrez).

1.2.3 Específico

Nesta seção definimos as atividades que foram realizadas neste trabalho, e que forneceram os subsídios necessários para o processo de análise e definição dos conceitos e ferramentas propostos nesta dissertação. São elas:

- Analisar os conceitos existentes e inerentes à formalização de conhecimentos heurísticos de jogos. Fundamentos como vantagem material, temporal, espacial, e simetria devem ser cuidadosamente observados.
- Entrevistar enxadristas para obter conceitos genéricos sobre heurísticas de Xadrez, e saber quais funcionalidades são esperadas por eles em uma ferramenta de software de apoio ao ensino destes conceitos.
- Especificar uma linguagem para a formalização de heurísticas de Jogos. Esta linguagem deve permitir que as heurísticas especificadas possam ser incrementadas, anotadas, e utilizadas por um jogador automático para a simulação de partidas entre heurísticas.
- Complementar a especificação do ambiente de ensino *CACAREJE* realizada por Daniel Martineschen (2006), especificando os módulos da aplicação cliente e as características da Ferramenta de Autoria de Heurísticas.
- Estudar o uso de técnicas de Representação Externa de Conhecimento em ambientes de

ensino, e a partir deste estudo projetar a interface da Ferramenta de Autoria de Heurísticas, minimizando o esforço cognitivo do usuário na execução das tarefas.

- Desenvolver o protótipo da Ferramenta de Autoria de Heurísticas. Neste processo será utilizada a linguagem de definição de heurísticas e o projeto de interface elaborado.

1.3 Contexto do Projeto

A pesquisa realizada nesta dissertação está inserida no projeto *PROTEX*, de apoio computacional ao ensino de Xadrez nas escolas brasileiras, apresentado em (Direne *et al.* 2004). Neste projeto é delineado um ambiente para ensino, autoria e competição de heurísticas. O Sistema a ser desenvolvido será integrado ao servidor de Xadrez do CEX (Centro de Excelência de Xadrez), e possibilitará que as disputadas realizadas por meio de partidas simuladas entre as heurísticas possam ser acessadas, e utilizadas como material de estudo pelos aprendizes.

O primeiro trabalho realizado visando este objetivo foi um estudo sobre a definição de uma linguagem para a representação interna de heurísticas do Xadrez (Souza *et al.* 2004). Este estudo serviu como estímulo para as pesquisas realizadas nesta dissertação, que especificou uma linguagem de formalização de heurísticas mais ampla e genérica.

O presente trabalho foi realizado em paralelo com a dissertação de mestrado do aluno Daniel Martineschen (2006). A arquitetura do ambiente de ensino de conceitos heurísticos é definida em conjunto pelos dois estudos. Nesta dissertação o foco é a aplicação cliente e as interfaces de usuário, ao passo que no trabalho de Martineschen são definidos os conceitos e componentes da aplicação servidora.

Martineschen também apresenta uma análise crítica sobre o uso da Colaboração alternada com a Competição em ambientes de aprendizagem, e discute uma implementação do protótipo do

algoritmo de busca heurística, que utilizou o algoritmo Minimax com Podas Alfa-Beta (Rich 1983). O artigo de Martineschen *et al.* (2006) apresenta uma análise sucinta sobre este trabalho realizado.

1.4 Estrutura da Dissertação

A presente dissertação está organizada da seguinte forma: Após esta introdução, apresentamos no Capítulo 2 a análise de trabalhos correlatos às áreas de Representação Externa, uso de Aprendizagem Colaborativa, e sobre a aplicação de Jogos na Educação. No Capítulo 3 são discutidos os fundamentos relacionados à formalização de heurísticas de jogos, e a partir desta pesquisa é proposta a linguagem para definição de conceitos heurísticos *DHJOG*. No Capítulo 4 é formalizada a arquitetura do sistema *CACAREJE*, que é um ambiente de aprendizagem referenciado por jogos educativos que alterna colaboração e competição no processo de ensino, e são analisadas as técnicas utilizadas na implementação do protótipo da ferramenta de autoria de heurísticas, que foi desenvolvida para suportar a linguagem *DHJOG*. No Capítulo 5 é apresentada a conclusão deste estudo, e são propostos alguns trabalhos futuros. No Apêndice A é apresentada a especificação da linguagem *DHJOG*, e no Apêndice B a extensão realizada nela para o jogo de Xadrez. Ao final do documento segue a lista de referências bibliográficas utilizadas durante a elaboração deste trabalho.

2 CAPÍTULO

TRABALHOS CORRELATOS

Neste capítulo realizamos uma análise crítica de alguns trabalhos relacionados com o problema exposto neste trabalho. Esta abordagem tem o objetivo de ressaltar quais aspectos são relevantes para a metodologia utilizada e para substanciar a solução do complexo problema em questão.

2.1 Representação Externa

O conceito de Representação Externa (*ER – External Representations*) pode ser definido como o uso de técnicas para representar, organizar e apresentar conhecimento ao usuário. A utilização de figuras, textos e gráficos são exemplos clássicos de uso de técnicas de representação Externa. Autores como Zhang (1997 e 2001) mostram como é importante o uso adequado de representações externas para facilitar a resolução de problemas complexos.

O benefício mais evidente do uso de representações externas (RE) é o de facilitar o processo de memorização, reduzindo a carga cognitiva necessária para se realizar uma tarefa. Todavia, esta ainda não é sua característica fundamental. Uma RE pode possuir as seguintes propriedades (Zhang, 1994):

- prover informações que podem ser usadas diretamente sem nenhuma interpretação ou formulação explícita;
- fixar comportamento cognitivo;
- mudar a natureza das tarefas a serem realizadas, podendo torná-las mais fáceis ao

usuário.

Com a finalidade de demarcar os resultados das técnicas relevantes, criadas em pesquisas passadas, apresentaremos a seguir alguns estudos sobre novas abordagens de uso de RE, principalmente aquelas associadas a software de ensino e aprendizagem.

2.1.1 Representação de Domínios de Conhecimento

A seguir serão apresentados estudos que mostram a importância do uso de REs na descrição do domínio de conhecimento para aprendizes, instrutores e autores.

O autor Swidersk (2001) propõe a criação de uma ferramenta de controle de conteúdo multimídia em sistemas hipermídia. Esta ferramenta possibilita ao usuário controlar quais tipos de mídias são mostrados (vídeo, som, texto, etc.) e qual o nível de detalhamento das representações externas é apresentado na aplicação. Desta forma, o aprendiz pode acomodar a interface ao seu nível de conhecimento e à suas preferências. Se ele for um usuário iniciante, pode escolher acessar menos detalhes sobre um conteúdo e utilizar mídias mais simples, como vídeo. A partir do momento que ele se sinta confortável com o tema, ele pode querer aprofundar os estudos. Para isto ele altera a interface para apresentar mais informações e para utilizar REs mais complexas, como diagramas, textos e fórmulas. O uso desta técnica pode tornar sistemas hipermídia mais eficazes em termos de facilidade de acesso aos seus conteúdos.

No estudo feito por Labeke (2001) é discutida a importância do uso de várias formas de RE em ambientes de Ensino. Estes sistemas são chamados MER (*Multiple External Representations*). É apresentado neste trabalho o sistema Calques3D para ensino de conceitos geométricos. Neste sistema existem diferentes formas de representações externas que são utilizadas para apresentar o mesmo conhecimento, só que por abordagens diferentes. É possível criar uma figura geométrica e

analisá-la através de uma visão 3D, uma visão 2D, uma lista de figuras geométricas que a compõem, e por fórmulas matemáticas que a descrevem. Isto permite que o aprendiz tente utilizar a RE que mais se adapte à sua capacidade cognitiva atual. A Figura 2.1.1 apresenta algumas das REs disponíveis no sistema Calques 3D.

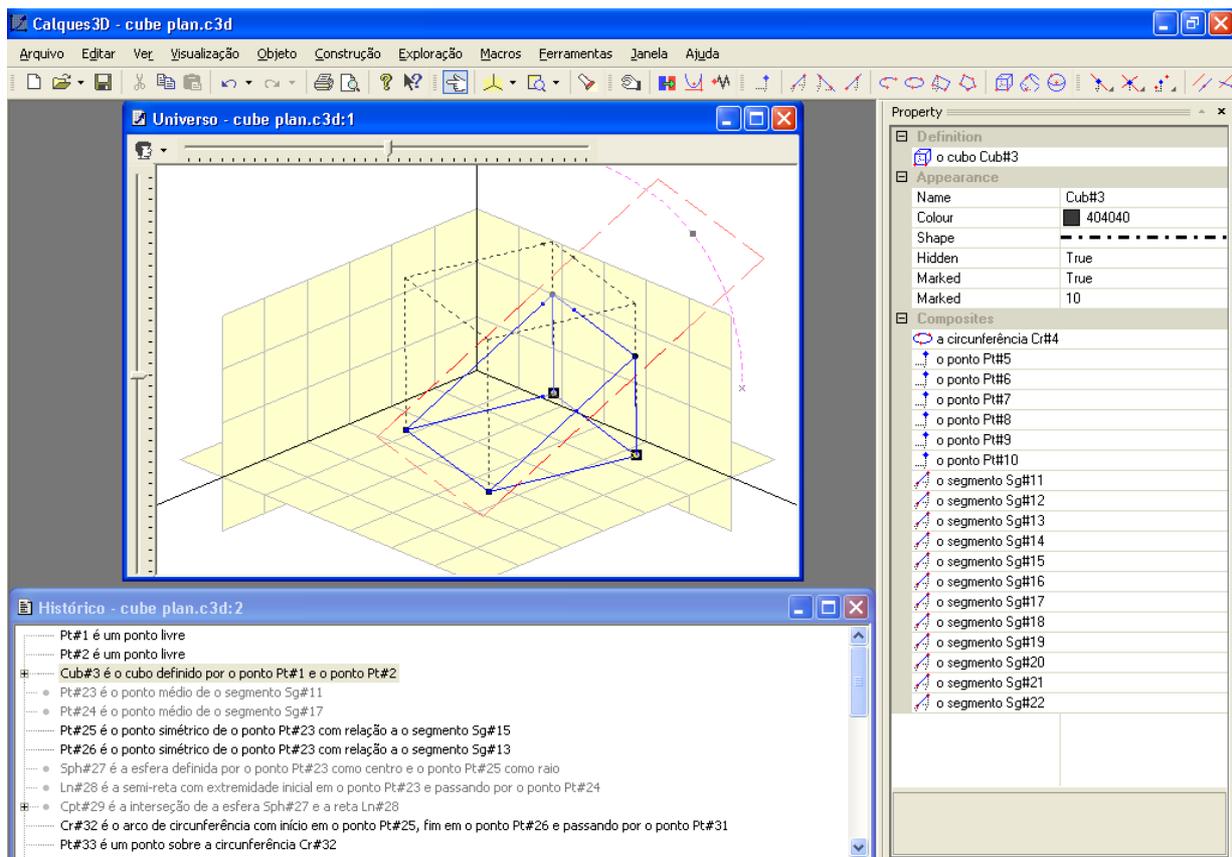


Figura 2.1.1 – Sistema MER Calques 3D

Desenvolver sistemas MERs eficazes não é uma tarefa simples. Aisnworth (1999) apresenta um estudo aprofundado de como construir sistemas MER de Ensino realmente eficazes.

No estudo realizado por Willians (2001) é demonstrada a importância da utilização de REs adequadas em ferramentas de autoria para STIs (Sistemas Tutoriais Inteligentes), a fim de facilitar as tomadas de decisões locais em um domínio complexo e que envolve um contexto global. Neste

estudo são apresentadas melhorias ao sistema de autoria de STIs REDEEM com a incorporação de novas REs na forma de mapas de navegação, sumários e estruturas de conteúdo. Neste estudo, Willians verifica que o uso de REs em sistemas de autoria de STIs pode: (i) diminuir a carga cognitiva necessária para usar a ferramenta; (ii) tornar as informações imediatamente acessíveis; (iii) representar o conhecimento de forma estruturada e relacionada (demonstrando a relação existente entre os vários elementos do ambiente de autoria).

2.1.2 Representações Externas em Ambientes de Modelagem e Colaboração

Os estudos analisados a seguir mostram como o uso adequado de representações externas em sistemas de modelagem, colaboração e simulação, são cruciais para motivarem o aprendiz a realizar as tarefas propostas.

Murray apresenta o sistema SimForest (Murray *et al.* 2001) que é um ambiente de simulação de crescimento de árvores em florestas. Neste projeto ele utiliza os conceitos de caixa preta (*black box*) e caixa de vidro (*glass box*), sendo cada um deles usado em uma versão específica do programa. Na versão caixa preta o aprendiz pode alterar os parâmetros utilizados pelo modelo interno de simulação, e acompanhar os resultados. Por exemplo, aumenta-se a taxa de crescimento de uma espécie de árvore e ela acaba dominando a floresta. Esta versão é direcionada para aprendizes iniciantes que poderão usá-la para realizar experimentos. Já um aprendiz experiente pode querer entender como o modelo de simulação funciona, e quais são as variáveis que se relacionam entre si. Neste caso ele deverá utilizar a versão caixa de vidro. Nesta versão o aprendiz pode através de outras Representações Externas acessar os detalhes internos do modelo usado para a simulação, editar as formulas matemáticas que o regem, e propor novas dependências e parâmetros.

Com o uso destas duas abordagens (*black box* e *glass box*) o aprendiz consegue utilizar o

sistema desde o início dos estudos e, após consolidar os conhecimentos, utilizar a ferramenta mais avançada. O aprendiz consegue através do sistema acompanhar situações concretas, entender o relacionamento entre variáveis, explicar sobre os princípios que existem por trás das equações de simulação e, por último, gerar novos modelos. Estes novos modelos podem ser trocados entre aprendizes e instrutores e podem ser usados para formular novas teorias do problema. A Figura 2.1.2-1 apresenta a tela principal da versão caixa preta, já a Figura 2.1.2-2 apresenta a tela de edição de equações da versão caixa de vidro.

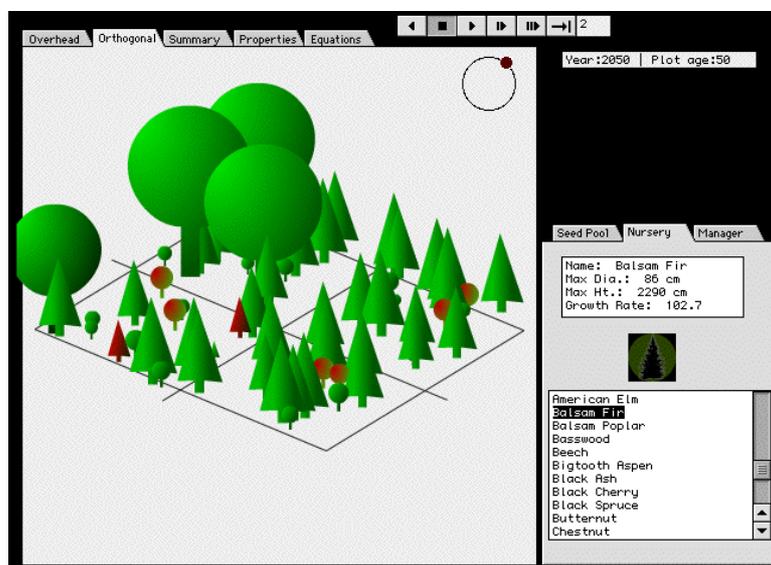


Figura 2.1.2-1 – SimForest B – Exemplo de uso da técnica Caixa Preta

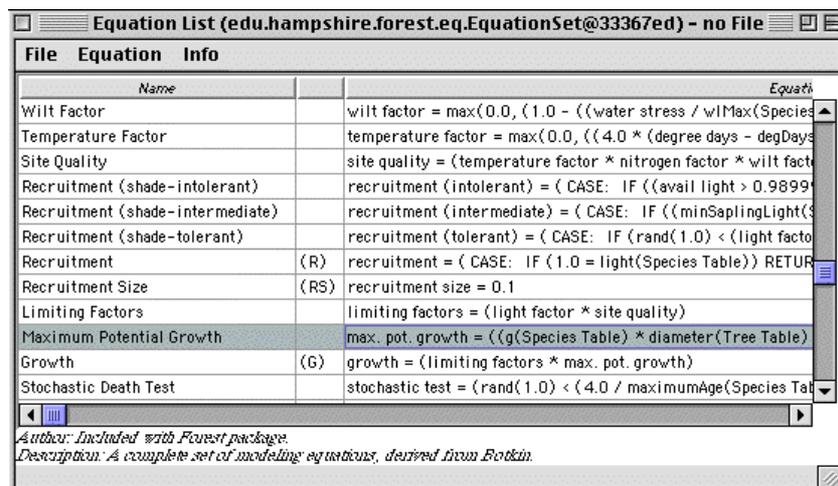


Figura 2.1.2-2 – SimForest G – Exemplo de uso da técnica Caixa de Vidro

O autor Joolingen (2001) apresenta um estudo realizado sobre o uso de REs em um ambiente colaborativo de ensino, onde os aprendizes podem modelar e simular modelos matemáticos. Ele utilizou neste sistema três tipos de REs: gráfica (na forma de um grafo); textual (fórmulas matemáticas); dinâmico-visual (um gráfico de execução do modelo). Durante os experimentos realizados a RE de natureza dinâmico-visual estava acessível a todos os aprendizes para que pudessem validar os modelos criados. Apenas os aprendizes que utilizaram a representação textual confrontaram a simulação esperada com o gráfico da representação dinâmico-visual durante o processo de modelagem.

Após este estudo Joolingen chegou às seguintes considerações iniciais: (i) a RE utilizada na modelagem pode motivar ou inibir o aprendiz a refinar o modelo criado; (ii) o aprendiz iniciante que utiliza apenas a representação gráfica consegue rapidamente criar um modelo, só que o aprendiz não se preocupa em refiná-lo e aprofundar suas especificações. (iii) o aprendiz que utiliza apenas a representação textual despende um tempo inicial buscando entender de forma mais ampla a relação entre o conhecimento necessário para a tarefa, e leva mais tempo para finalizar um novo modelo, mas este fica mais específico e refinado.

2.2 Uso de Aprendizagem Colaborativa em Aplicações de I.A. na Educação

Os primeiros sistemas de Instrução Assistida por Computador (CAI) tinham a preocupação apenas de apresentar o conteúdo ao aluno, eram desenvolvidos apenas com as informações referentes ao domínio do instrutor, apresentavam o conhecimento de forma linear, influenciados pela Teoria Psicológica Behaviorista (Skinner 1958). Constatou-se que esse processo era ineficiente e que aprendizes poderiam responder de formas diferentes ao mesmo estímulo. Os alunos possuem particularidades que devem ser levadas em consideração durante a definição da estratégia a ser

adotada para o processo de ensino e de aprendizagem.

Já os Sistemas Tutores Inteligentes (STI) surgiram com o intuito de agregar técnicas de Inteligência Artificial para que o processo de ensino se adapte às características específicas de cada aprendiz. Atualmente a área de Inteligência Artificial aplicada à Educação busca agregar novas tecnologias para tornar a experiência de aprendizagem mais rica e eficaz. No momento ocorre uma evolução nos STIs que passam a utilizar maciçamente técnicas de aprendizagem colaborativa, uso de recursos multimídias e de realidade virtual (Rosatelli 2000). Sistemas de ensino como o LeCS (Rosatelli *et al.* 2000), o MArCo (Tedesco 1999) e o MCOE, (Giraffa 1999) são exemplos de aplicações de I.A na educação que usam conceitos de colaboração no processo de aprendizagem.

O projeto de um ambiente de colaboração que seja realmente eficaz depende de um bom planejamento inicial da ferramenta. Em trabalhos como (Ramos 1999) e (Arriada 2001) são discutidas características e diretrizes importantes que devem ser avaliadas durante o processo de projeto de sistemas, que pretendem facilitar, ou mesmo permitir simplesmente, a condução de tarefas de aprendizagem colaborativa. Todavia, nenhuma das pesquisas anteriores abordou o conceito de alternância da colaboração e da competição entre os aprendizes, e como estas duas modalidades de atividades poderiam ser coordenadas pelo instrutor da comunidade de aprendizagem.

2.3 Uso de Jogos eletrônicos no processo de Ensino

O uso de jogos no processo de ensino proporciona ao aprendiz uma maneira lúdica, ainda que desafiadora, de abordar indiretamente a precisão e o encadeamento lógico exigidos no tratamento de certos campos específicos do conhecimento humano. Existe a falta de um processo mais criterioso de desenvolvimento de aplicações educacionais comerciais que utilizem jogos no

processo de ensino. Isto faz com que muitos deles não sejam mais do que meros itens de “passa-tempo” em formato eletrônico.

Várias pesquisas e sistemas foram desenvolvidos utilizando jogos no processo de ensino. Sistemas como o WEST e o QUEST utilizam os jogos como ferramentas para o ensino de conceitos específicos de domínio (*e.g.*, Álgebra), conseguindo ótimos resultados em experimentos realizados com fins de avaliação (Wenger 1987). Todavia, ainda existe uma grande lacuna de conhecimento na área de ensino dos conceitos de natureza heurísticas, e de estratégias de jogos em si. É preciso aprofundar estudos sobre os fundamentos desta tarefa, e no desenvolvimento de ferramentas que auxiliem o ensino destes conceitos. Os estudos realizados em Direne *et al* (2004), Schäfer (2000), Souza (2004), e Martineschen (2006) vão de encontro a este objetivo. São exemplos de trabalhos realizados no ensino de conceitos heurísticos de jogos, especificamente do jogo de Xadrez.

As ferramentas existentes atualmente para o ensino de jogos se preocupam apenas em apresentar as regras básicas do jogo e, logo em seguida, simular adversários para o aprendiz humano. Este é o caso do *Fritz*, do *Chessbase* e do *Chessmaster*, os quais já são considerados clássicos do mundo comercial para o ensino de Xadrez. No entanto, nenhum deles se preocupa em abordar explicitamente qualquer parâmetro heurístico, ou estratégias que tenham maior alcance do que o planejamento em um ou dois lances à frente da configuração atual. Fora do mundo comercial, também não foram encontrados na literatura de pesquisa trabalhos sobre conceitos e ferramentas de ensino de jogos, que enfoquem a aprendizagem de estratégias através da formalização e validação de conceitos heurísticos.

3 CAPITULO

CONCEITOS E LINGUAGEM DE DEFINIÇÃO DE COMPONENTES HEURÍSTICOS

Neste capítulo iremos primeiramente analisar os conceitos relacionados a heurísticas de jogos. Quais elementos existem em uma heurística, a relação e dependência entre eles, e como podem ser utilizados no processo de formalização. Este estudo foi a base para a especificação dos fundamentos da linguagem de definição de heurísticas de Jogos (*DHJOG*). Iremos apresentá-la e discuti-la na Seção 3.2. Vale salientar que apesar das aplicações deste presente trabalho serem destinadas ao jogo de Xadrez, os conceitos envolvidos são razoavelmente genéricos em relação a outros jogos por turno¹.

3.1 Conceitos Fundamentais para Definição de Heurísticas

Nesta seção iremos apresentar alguns conceitos fundamentais para a definição de heurísticas de jogos. Estes conceitos são características, comportamentos e elementos que estão diretamente relacionados aos conceitos envolvidos no cálculo heurístico.

3.1.1 Relação Material, Espaço e Tempo

O jogo de Xadrez é possivelmente o jogo mais analisado em termos de estrutura do jogo, de eficácia de estratégias, e até de efeitos pedagógicos do seu uso em ambientes de ensino. Sempre despertou fascínio entre jogadores e pesquisadores. Uma das análises fundamentais realizadas sobre

¹ Um jogo por Turno possui a alternância constante de lances entre os jogadores.

este jogo e que pode ser estendida facilmente a outros jogos é a relação Material, Espaço e Tempo.

Vamos definir estas grandezas e as características que ocorrem quando se possui vantagem durante uma partida em cada uma delas:

- *Material*. Conjunto de peças disponíveis ao jogador. Quando se possui uma vantagem material é que o conjunto de peças de um jogador é mais forte do que o do adversário. O termo força está associado diretamente a cada tipo de peça do jogo e as jogadas possíveis a ela. Por exemplo, no Xadrez uma Dama é mais forte que um Peão.
- *Espaço*. Posições do jogo ocupadas pelas peças do jogador. É dito que um jogador possui uma vantagem posicional (espacial) quando ele ocupa mais posições do que o adversário ou, quando ocupa posições mais estratégicas do jogo.
- *Tempo*. Tempo utilizado ou restante ao jogador para executar suas jogadas. Um jogador pode possuir vantagem temporal quando ele avança com seu conjunto de peças a um determinado objetivo com menos jogadas realizadas do que o adversário.

Durante uma partida o jogador pode priorizar determinada vantagem em relação a outra de acordo com a estratégia adotada. Determinadas vantagens abrem novas possibilidades durante o jogo. Possuir vantagem material pressupõe mais possibilidades de ataque. Vantagem posicional pode significar ter mais espaço para movimentar as peças e montar estratégias. Dispor de vantagem temporal indica que o jogador possui jogadas sobrando depois de ter realizado um objetivo, e pode gastá-las contra o oponente. Toda esta alternância de estratégias de um jogador pode acontecer durante uma única partida.

Deve-se incluir na linguagem de definição de heurísticas de jogos elementos para identificar estas grandezas (vantagens), e posteriormente no processo de formalização da heurística de priorizá-las.

3.1.2 Valor inicial de peça

Uma peça é um elemento de jogo que é utilizado pelos jogadores durante a dinâmica de uma partida. Nos jogos, as peças podem ser de diferentes tipos, possuírem diferentes características e comportamentos distintos. Cartas de um baralho (Três de Paus, Reis de Ouro,...), Peças de um jogo de Xadrez (Dama, Peão, Torre, etc.), e elementos de um jogo de *JokenPo* (Papel, Pedra e Tesoura) são exemplos de peças de jogo.

No contexto de um jogo cada tipo de peça poderá ter uma importância maior ou menor para um jogador, de acordo com a visão que ele tem de como esta peça pode ajudá-lo a conquistar a vitória. Esta importância é definida atribuindo a cada tipo de peça um valor inicial. Por exemplo, Peão vale 1, Torre vale 2, Cavalo vale 4, etc. Este valor escolhido deve levar em consideração as características e comportamentos da peça comparando-os com os demais tipos existentes, de forma independente de situações específicas do jogo. No Xadrez um Peão sempre é considerado uma peça mais fraca do que uma Dama, pelas suas limitações de ação.

O valor inicial definido para cada tipo de peça será utilizado durante o cálculo heurístico de uma situação de jogo para verificar o quanto esta situação é favorável ou não ao jogador.

3.1.3 Valor relativo de peça

Como definido no item anterior cada tipo de peça terá um valor inicial que indica sua importância em termos conceituais dentro das regras e mecanismos do jogo. Mas uma peça pode em determinadas situações de jogo se apresentar como mais ou menos vantajosa ao jogador. É o caso de um Quatro de Ouros em um jogo de Truco. Na maioria das rodadas ela será considerada a carta mais fraca da mesa, mas quando a virá é um Três ela se torna a quarta maior carta do baralho.

Devido a esta peculiaridade é fundamental que possamos em determinados momentos alterar a importância que determinadas peças terão no cálculo do valor heurístico da situação de jogo. Ou seja, seu valores iniciais são momentaneamente substituídos por valores relativos.

3.1.4 Identificação de Contextos

Uma heurística deve apontar o quanto uma situação de jogo é favorável ou não ao jogador. Este apontamento é feito de acordo com o valor numérico retornado do cálculo heurístico. O primeiro passo antes de determinar este cálculo é verificar se determinados contextos são encontrados na situação de jogo em análise. E de acordo com a interpretação definida na heurística estes contextos podem significar maior ou menor vantagem ao jogador. Vejamos a seguir quais são as características de contextos elementares que são encontrados em uma situação de jogo.

3.1.4.1 Quantitativo

A relação mais simples a ser identificada em uma situação de jogo é a questão de quantidade de peças dos jogadores. Uma heurística simples de Xadrez por exemplo poderia verificar apenas a diferença entre a quantidade de peças de um jogador em relação a quantidade do seu adversário. Esta análise não leva em consideração os tipos de peças que cada jogador possui, o que importa unicamente é a quantidade. Logicamente isso não se mostra eficiente em jogos onde os comportamentos das peças diferem bruscamente. Possuir três Peões não é melhor do que possuir dois Bispos. Já em jogos onde só existam peças idênticas ou com comportamentos similares, talvez a relação quantitativa de peças seja suficiente para verificar vantagem ou desvantagem em determinada situação de jogo.

3.1.4.2 Qualitativo

É possível avaliar uma situação de jogo de acordo com a qualidade das peças disponíveis naquele momento. O termo qualidade é algo subjetivo e está relacionado ao valor inicial atribuído a cada tipo de peça pelo próprio jogador (Veja Seção 3.1.1). Desta forma um jogador pode definir uma heurística que lhe indique a relação de “poder” de ataque de suas peças comparadas com as do oponente. Vale ressaltar que como discutido na Seção 3.1.2 o valor inicial de uma peça indica uma avaliação de qualidade isenta de situações específicas de jogo, em determinados casos uma peça terá o seu valor heurístico incrementado, o que indicará que seu comportamento padrão já não é válido naquela situação.

3.1.4.3 Posicional

Uma outra característica que pode ser identificada e avaliada em uma situação de jogo é a questão da distribuição das peças pelas posições disponíveis no jogo. Onde pode-se buscar identificar num dado momento quais peças estão em determinada região ou, em quais regiões estão determinadas peças. Um exemplo a ser mencionado é a estratégia de Xadrez de, na abertura do jogo, o jogador buscar dominar o centro do tabuleiro. Este domínio teoricamente garantirá a ele melhores possibilidades de jogadas no decorrer da partida. Toda esta definição heurística é bem subjetiva e depende da visão particular que o jogador possui a respeito do que é para ele o centro do tabuleiro, e como ele entende que uma região está dominada. Neste trabalho é utilizado a notação algébrica para identificar as casas de um tabuleiro de Xadrez². Na figura abaixo vemos um exemplo de definição da região central de um tabuleiro de Xadrez entre as casas C6,F6,C3 e F3.

² A notação Algébrica divide um tabuleiro de Xadrez em colunas de “a” a “h” e linhas de 1 a 8. É a notação oficial recomendada pela FIDE (Fédération Internationale des Échecs) - Federação Internacional de Xadrez.

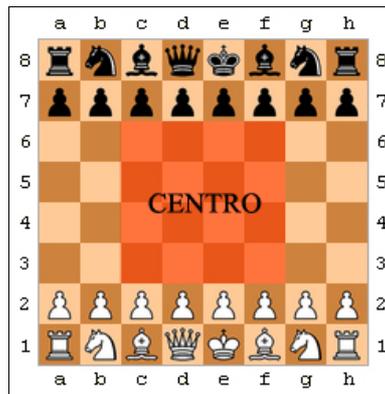


Figura 3.1.3.3 – Exemplo de Definição de Centro de um Tabuleiro

Um outro exemplo é que para um determinado jogador dominar o centro do tabuleiro pode ser identificado através da diferença entre a quantidade de peças dele e a quantidade de peças do adversário (Veja a Seção 3.1.3.1). Já para outro jogador dominar o centro depende da qualidade das peças naquela região, e pode ser identificada através da diferença entre a soma dos valores iniciais das peças que ele possui menos a soma das peças do adversário (Veja a Seção 3.1.3.2).

3.1.4.4 Temporal

Outro fator que pode ser identificado em uma situação de jogo é a questão do tempo decorrido até aquele momento, em termos de segundos decorridos na partida ou de quantidades de lances realizados. Pode-se conferir esta informação de forma geral analisando a partida como um todo, ou individualmente para cada jogador.

Um dos usos desta característica é poder utilizá-la para identificar estágios existentes dentro de uma partida, e adaptar as heurísticas utilizadas ao estágio corrente. No Xadrez é comumente definido três estágios principais: Abertura, Meio de Jogo, e Finalização. Só que a definição de quando se está em um estágio e quando se está em outro não é clara nem formalizada, e cabe a cada

jogador criar suas próprias regras de identificação. Por exemplo, um jogador poderia utilizar o critério de que a Abertura é um estágio que vai do primeiro ao décimo lance jogado, e neste período focar em dominar o centro do tabuleiro.

3.1.4.5 Comportamental

É possível em uma situação de jogo identificar comportamentos em determinadas peças de acordo com as possíveis aplicações de regras de transformações disponíveis naquele momento. Pode-se por exemplo verificar que uma peça está sendo atacada, atacando, livre, bloqueada e etc. Um jogador pode utilizar desta informação e construir uma heurística onde a vantagem em uma situação de jogo é calculada pela quantidade de peças atacando menos a quantidade de peças sob ataque.

Outra questão relacionada ao comportamento das peças é a possibilidade ou não de aplicar regras de transformação que dependam de pré-condições para ocorrerem. Por exemplo, no jogo de Xadrez a jogada “*Roque*” que consiste em movimentar o Rei e uma Torre simultaneamente para proteger o Rei só pode ocorrer se os seguintes pré-requisitos forem satisfeitos:

- O Rei nunca foi movido.
- A Torre a ser usada no Roque nunca foi movida.
- O Rei não está em Xeque.
- As casas entre o Rei e a Torre estão desocupadas.
- Nenhuma das casas pelas quais o Rei irá passar ou ficar está sob ataque.

A jogada de Roque pode ter duas variações de acordo com a Torre que será utilizada. Desta

forma podemos dizer que o Rei está em determinado momento do jogo passível de “*Roque Grande*”, ou passível de “*Roque Pequeno*”, ou de ambos, ou de nenhum. A Figura 3.1.3.5 demonstra os possíveis tipos de *Roque* do Xadrez.

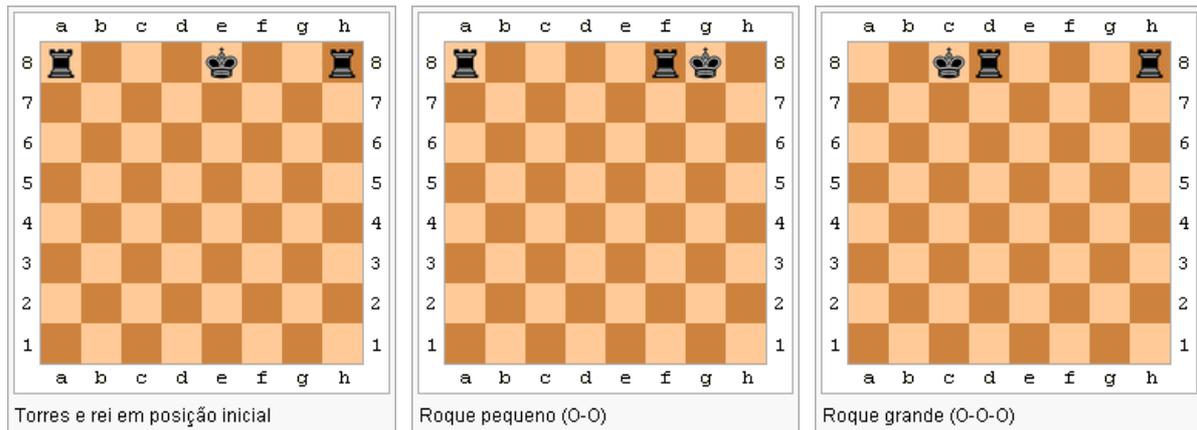


Figura 3.1.3.5 – Exemplo de Tipos de Jogada Roque do Xadrez

Um outro exemplo do Xadrez é a jogada de captura de Peões chamada “*en-passant*”, onde um Peão pode ser capturado por outro Peão na mesma linha, caso o primeiro tenha saltado duas casas na jogada imediatamente anterior. Da mesma forma que no *Roque*, podemos dizer então que em determinado momento um Peão pode estar passível de “*captura en-passant*” ou não.

3.1.5 Simetria

Simetria é um conceito fundamental em jogos e sua compreensão se tornou importante no processo de análise realizado durante este trabalho de mestrado, proporcionando-nos subsídios para a criação de mecanismos que garantam maior legibilidade e compreensão no formalismo de heurísticas de jogo.

Dentro do escopo deste trabalho podemos identificar os seguintes tipos de Simetria: Simetria de Jogo, Simetria Heurística e Simetria da Interpretação Heurística. Vamos a seguir definir e exemplificar cada um deles.

3.1.5.1 Simetria de Jogo

A simetria do jogo é o princípio de que todos os jogadores tenham inicialmente a mesma probabilidade de vitória. Nenhuma pessoa gostaria de jogar um jogo onde pelo simples fato dela iniciar o jogo com um conjunto de peças específicos, ou por iniciar o jogo em determinada posição ela já esteja prejudicada. A área de estudo sobre projeto de jogos chamada “*Game Design*” define este conceito de simetria como sendo balanceamento de jogo, “*Game Balance*” (Rouse 2001).

No Xadrez os jogadores possuem inicialmente a mesma quantidade de cada tipo de peça e o tabuleiro é idêntico em ambos os lados. Mas podemos identificar uma pequena assimetria no jogo de Xadrez, que é o fato de que as peças brancas por executarem a primeira jogada levam uma pequena vantagem inicial. A forma de se contornar está assimetria é realizar mais de uma partida entre os mesmos jogadores alternando-se quem joga com as brancas.

3.1.5.2 Simetria Heurística

A simetria heurística vem da seguinte observação: Uma heurística de jogo irá identificar a existência de determinado contexto em uma situação de jogo (Veja a Seção 3.1.3), e de acordo com a interpretação pessoal definida pelo jogador irá qualificar aquela situação como mais ou menos vantajosa a ele. Pressupondo-se que o jogo em questão seja simétrico (como é o caso a ser considerado sobre o Xadrez), podemos constatar que a mesma vantagem que pode ocorrer para um

jogador poderá acontecer para o outro.

Vejamos o seguinte exemplo: Um jogador define uma heurística em que caso não exista nenhuma Peça do adversário na região compreendida pelas casas {e2,f2,g2 e h2} isto é uma vantagem para ele, pois ele considera que o “*Roque Pequeno*” está protegido, e por isso atribui um valor +100 ao valor heurístico calculado para esta situação de jogo. A Figura 3.1.4.2-1 demonstra a região definida.

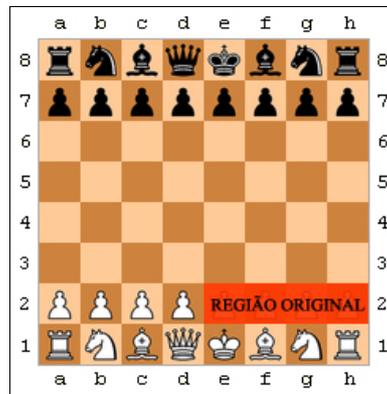


Figura 3.1.4.2 -1 – Exemplo de Definição de Região para Proteger Roque

Claramente está vantagem definida na heurística pode acontecer para o outro jogador. O adversário também poderá ter a região a frente do seu “*Roque Pequeno*” livre de ameaças. Para encontrarmos a região do adversário equivalente a definida pelo jogador basta rotacionarmos o tabuleiro em 180°. Desta forma a região espelho a definida pelo jogador é a região compreendida pelas casas {a7,b7,c7 e d7}. A Figura 3.1.4.2-2 apresenta esta nova região.

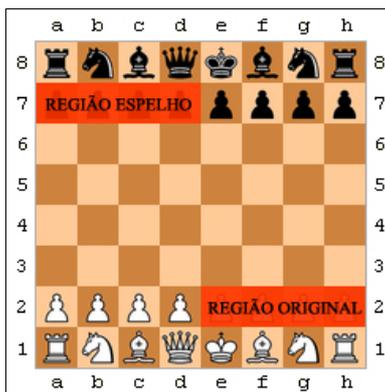


Figura 3.1.4.2-2 – Região Espelho de Proteção de Roque Pequeno

Pode-se então em determinada situação de jogo existir a vantagem definida de proteção de “*Roque Pequeno*” acontecendo para ambos os jogadores, neste caso não é prudente que um jogador afirme que possui esta vantagem. Uma vantagem de jogo só é realmente uma vantagem real se a mesma não se aplica ao adversário. Pois caso contrário, ambas se anularão, já que cada jogador ganhará +100 para a situação de jogo atual.

O correto então seria definir a heurística da seguinte forma: Caso não exista nenhuma Peça do adversário na região compreendida pelas casas {e2,f2,g2 e h2}, e caso não exista nenhuma Peça minha na região compreendida pelas casas {a7,b7,c7 e d7}, somente eu possuo a região de “*Roque Pequeno*” protegida, devido a isto o valor heurístico calculado para esta situação de jogo será incrementado em +100.

Toda está análise feita para a especificação de uma vantagem também se aplica à especificação de uma desvantagem de jogo. Uma desvantagem só é uma desvantagem real se a mesma não se aplica ao adversário.

A fim de que o Autor de heurísticas possa compreender este conceito e melhore a legibilidade do formalismo criado, a própria ferramenta de edição de conhecimento heurístico deverá questionar o autor sobre a existência ou não de simetria a cada definição de vantagem ou desvantagem, e embutir automaticamente na especificação criada a não ocorrência da equivalente

da mesma para o adversário.

3.1.5.3 Simetria da Interpretação Heurística

A Simetria da Interpretação Heurística está relacionada ao fato de que as heurísticas formalizadas pelo autor na ferramenta de edição serão posteriormente utilizadas por um jogador automático, que utilizará uma variação do algoritmo de Minimax. Veja a dissertação de mestrado de Daniel Martineschen (2006) para uma análise completa do algoritmo escolhido.

O algoritmo de Minimax a partir da situação de jogo atual cria um grafo E-OU com as próximas possíveis jogadas. Ele utiliza a função heurística de um jogador para prever as jogadas de ambos, sempre procurando fazer a jogada que traga a melhor situação de jogo tanto para um quanto para o outro. Ao final da geração deste grafo ele escolherá a melhor jogada para o jogador atual buscando maximizar a vantagem dele e minimizar a vantagem do adversário (Rich 1983).

Neste processo como ele utiliza a mesma função heurística, a vantagem que ele identificar para o jogador caso exista para o adversário também será identificada, no momento de prever a jogada do adversário. Isto fará que ambas se anulem no momento de calcular a melhor situação heurística dentre todas as jogadas encadeadas. Ou seja, a forma de avaliação é simétrica para ambos os jogadores. Desta forma, o conceito de que uma vantagem só é uma vantagem real se não ocorrer para o adversário será tratada automaticamente pelo jogador automático.

A recomendação de que no momento de formalizar a heurística explicita-se que a vantagem só será computada caso não ocorra para o adversário (Veja a Seção 3.1.4.2), vem com o objetivo de tornar claro para o autor este conceito, evitando que o mesmo fique escondido dentro do processo de funcionamento do jogador automático, funcionamento este que é complexo para autores iniciantes.

3.1.6 Componentes Incrementais

Um componente incremental é um elemento utilizado durante o processo de avaliação de uma situação de jogo que pode ter o seu valor incrementado (para mais ou para menos). Desta forma, pode-se alterar a peso que o componente terá no resultado final heurístico calculado. Podemos identificar como elementos incrementais as Peças de Jogo (Veja a Seção 3.1.2), e o próprio valor heurístico inicialmente calculado.

3.1.7 Expressão de Cálculo Heurístico

A expressão de cálculo heurístico (ou função heurística) defini a forma como uma situação de jogo será avaliada, e procura identificar o quanto esta situação de jogo é favorável ou não a um determinado jogador. O grau de vantagem encontrado na situação de jogo e representado matematicamente através de um valor numérico retornado. Comumente valores positivos indicam vantagem e valores negativos indicam desvantagem ao jogador.

Alguns exemplos de expressões de cálculo heurísticos são:

- Calcular a diferença entre a soma dos valores das peças minhas menos a soma das peças do adversário.
- Calcular a diferença da quantidade de peças minhas que estão atacando menos a quantidade de peças minhas sob ataque.

3.2 Linguagem para Definição de Heurísticas de Jogos - *DHJOG*

A partir dos estudos realizados neste trabalho sobre os componentes heurísticos existentes em uma situação de jogo e a relação entre eles (Ver Seção 3.1), foi definida a linguagem para definição de heurísticas de jogos chamada *DHJOG* e que será apresentada nesta Seção. Esta linguagem é formada por vários componentes distintos (Etapas, Heurísticas, Primitivas, etc..) que de forma integrada possibilitam uma completa formalização do conhecimento heurístico de um jogo, que ainda poderá posteriormente ser utilizado por um jogador automático para a escolha de próximas jogadas.

O autor a formalizar um conhecimento heurístico de jogo estará criando um componente que chamaremos de Conjunto Heurístico. O termo conjunto é utilizado pois este elemento irá agrupar várias heurísticas distintas que serão utilizadas durante a avaliação das situação de jogo ao longo de uma partida. Onde pode-se ainda especificar quais heurísticas estarão disponíveis em determinado momento, e quais não. Adaptando-se desta forma a avaliação heurística ao contexto temporal atual do jogo (Veja Seção 3.1.3.4).

O Conhecimento Heurístico de Jogo a ser formalizado nesta linguagem pode ser compreendido como:

- Identificar em qual estágio de uma partida o jogo se encontra no momento, e quais heurísticas deverão ser aplicadas neste contexto temporal.
- Definir a relação de importância entre os elementos do jogo. O valor inicial e relativo das Peças, (Veja Seção 3.1.1 e 3.1.2).
- Identificar contextos em uma determinada situação de partida (Veja Seção 3.1.3), e especificar o quanto estes contextos contribuem como vantagem ou desvantagem ao jogador.

- Especificar de que forma o cálculo heurístico será realizado.

Vejamos a seguir quais foram os conceitos utilizados na definição dos componentes da linguagem, para que ela suporte a formalização dos conhecimentos heurísticos.

3.2.1 Regras de Produção

O uso de Regras de Produção é uma técnica para representação de conhecimento muito utilizada no desenvolvimento de sistemas especialistas. Consiste simplesmente em definir regras no formato de SE (Condição) ENTÃO (Ação). Pode-se ainda relacionar várias condições através de conectivos lógicos (E, OU e NÃO) para a criação de condições mais complexas. Vejamos algumas das vantagens do uso desta técnica segundo (Bratko, 1990):

- *Modularidade*: Cada regra pode ser considerada um elemento de conhecimento independente das demais.
- *Facilidade de Edição*: Devido a natureza modular das regras é fácil acrescentar, editar e excluir regras da base de conhecimento definida, com certa independência entre elas.
- *Transparência do Sistema*: Esta técnica garante maior legibilidade da base de conhecimentos definida.

Utilizaremos esta técnica para definir os componentes da linguagem *DHJOG* responsáveis por identificar contextos de uma situação de jogo (Veja Seção 3.1.3), são eles:

3.2.1.1 HTE – Heurística de Transição de Etapa

É uma heurística que a partir de um contexto de jogo identificado irá configurar uma nova

Etapa como sendo a Etapa atual do conjunto heurístico. Isto irá implicar em quais heurísticas estarão disponíveis para serem avaliadas no estágio atual da partida.

3.2.1.2 HVP – Heurística de Valor de Peça

É uma heurística que a partir de um contexto de jogo identificado irá incrementar o valor inicial de um ou mais tipos de Peças. Desta forma, será possível alterar a influência que estas Peças terão no valor heurístico calculado para a situação de jogo atual.

3.2.1.3 HVJ – Heurística de Valor de Jogo

É uma heurística que a partir da identificação de um contexto irá incrementar o valor heurístico já previamente calculado para a situação de jogo atual. Valor este que é gerado pela Expressão de cálculo heurístico (Veja Seção 3.1.6). Isto possibilita que determinado contexto tenha influência direta no valor heurístico total da situação de jogo.

3.2.2 Máquina de Estados Finito

Como foi discutido na Seção 3.1.3.4 podemos identificar em um jogo contextos temporais. Estes contextos podem indicar estágios (formais ou subjetivos) que acontecem durante uma partida. No caso do Xadrez existe a definição informal de Abertura, Meio de Jogo e Finalização. Um jogador normalmente utiliza técnicas distintas de acordo com o estágio em que o jogo se encontra. Por exemplo, na abertura de um jogo de Xadrez pode-se priorizar o domínio do centro do tabuleiro,

já no estágio de Meio de Jogo pode-ser focar em eliminar os Peões do adversário. Visto isto, é fundamental que a linguagem de formalização de heurísticas de jogos (*DHJOG*) permita ao Autor formalizar os estágios existentes na partida, e relacionar quais heurísticas deverão ser utilizadas em cada momento. Podemos constatar então que o processo de avaliação heurística de uma partida se comporta como uma Máquina de Estados Finitos.

Uma máquina de estados finitos é uma modelagem do comportamento de um sistema ao longo do tempo, é composto de estados, transições e ações. Um estado indica as mudanças que ocorreram desde a inicialização do sistema até o presente momento. Já uma transição indica uma possível mudança de estado, e é descrita por uma condição que precisa ser satisfeita para que a mudança efetivamente ocorra. E uma ação é a descrição de uma atividade que será realizada em um determinado momento. Esta técnica de especificação é amplamente utilizada na modelagem de comportamento de aplicativos, no projeto de sistemas digitais, na engenharia de software e no estudo de linguagens de computação.

A fim de implantar o conceito de máquina de estados no processo de avaliação heurística definimos um componente chamado Etapa, que representa um determinado estágio em uma partida. Cada Etapa possui os seguintes elementos:

- Um Conjunto de Heurísticas de Transição de Etapa (Veja Seção 3.2.1.1).
- A definição dos Valores iniciais de cada tipo de Peça (Veja Seção 3.1.1).
- Um Conjunto de Heurísticas de Valor de Peça (Veja Seção 3.2.1.2).
- Uma Expressão de Cálculo Heurístico (Veja Seção 3.1.6).
- Um Conjunto de Heurísticas de Valor de Jogo (Veja Seção 3.2.1.3).

Podemos relacionar cada um dos componentes apresentados acima com os elementos de um Máquina de Estados Finitos. Uma Etapa é um estado, uma Heurística de Transição de Etapa é uma Transição, e os demais elementos são Ações que serão realizadas enquanto a partida estiver nesta

determinado etapa. Isto permite que o Conjunto Heurístico tenha comportamentos diferenciados ao longo da execução de uma partida. A Figura 3.2.2 apresenta um exemplo de diagrama de transição de Etapas utilizando a notação UML³.

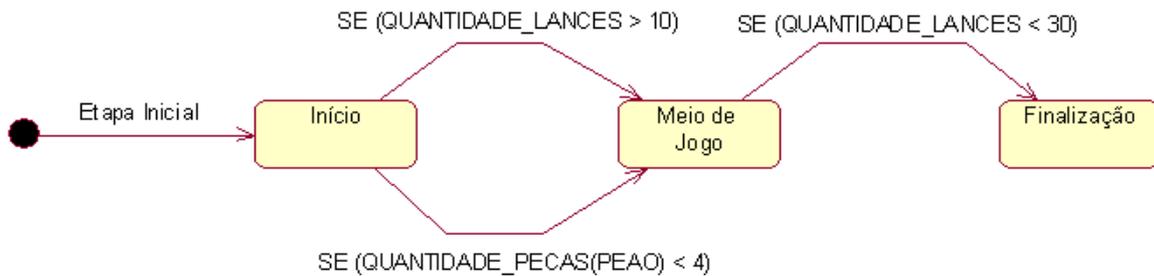


Figura 3.2.2 – Exemplo de Diagrama de Transição entre Etapas

Além da definição do componente Etapa também foi incluído no Conjunto Heurístico um atributo que indica qual será a Etapa inicial. Podemos entender agora que um Conjunto Heurístico é formado por uma ou mais Etapas. No contexto do processo de Avaliação Heurística de Jogo não se faz necessário definir estados finais, pois quem irá identificar o final da partida será o próprio jogador automático, antes de invocar o Conjunto Heurístico para avaliar uma situação de Jogo.

3.2.3 Desenvolvimento Orientado a Funções

A identificação de contextos em uma situação de jogo (Veja Seção 3.1.3) será utilizada dentro da linguagem *DHJOG* como condição de ativação das regras de produção, sejam elas Heurísticas de Transição de Etapa, Heurísticas de Valor de Peça, ou Heurísticas de Valor de Jogo (Veja Seção 3.2.1). Esta condição é formalizada como uma expressão matemática que retorna um valor lógico (Verdadeiro ou Falso), comumente chamada de expressão condicional.

³ *Unified Modeling Language* – Consiste em uma linguagem livre para modelagem orientada a objetos.

Foi identificada a necessidade de criar mecanismos que facilitem o Autor no processo de definição de elementos para a construção das expressões condicionais. A fim de atender esta necessidade a linguagem foi concebida para estimular o Desenvolvimento Orientado a Funções (Pressman 1995). Seguindo os princípios deste paradigma de projeto o Autor definirá funções capazes de identificar características de contextos em uma situação de jogo, e funções mais complexas deverão ser construídas utilizando funções mais simples previamente especificadas. Este processo possibilitará as seguintes vantagens:

- A ferramenta de edição proverá um **conjunto de funções elementares** para a identificação de contextos em uma situação de jogo.
- O Autor poderá utilizar este conjunto de funções elementares para iniciar o processo de definição de heurísticas. Isto fará com que ele necessite de um **menor esforço cognitivo** para aprender a desempenhar a tarefa.
- A definição de funções de escopo limitado permite um maior **encapsulamento** dos conceitos abordados.
- Ficará **transparente** para usuários iniciantes como determinados contextos são identificados. O Autor poderá então focar apenas em definir o quanto estes contextos são relevantes em uma situação de jogo quando ocorrem. Por exemplo, para um aprendiz iniciante de Xadrez não é importante entender como uma Harmonia de Peões é identificada em uma situação de jogo (a implementação desta função seria de alta complexidade), e sim entender o quanto esta característica é vantajosa ou não para o jogador.
- O Autor poderá manter uma **coleção particular de funções** que o ajude a entender os possíveis contextos de uma situação de jogo, e a definir novas heurísticas.
- O acesso às coleções permitirá uma grande **reutilização** das funções definidas, e

propiciará maior **produtividade** ao Autor.

- Autores mais experientes poderão definir novas funções para serem utilizadas por eles ou por toda a comunidade de aprendizes. Isto permite que a identificação de contextos seja um **processo em contínuo aprimoramento e expansão**.

3.2.4 Primitivas Lógico Matemáticas

A linguagem de definição de heurísticas para jogos *DHJOG* possui um ferramental vasto de primitivas lógico matemáticas que possibilitam uma grande expressividade e formalismo a ela. Vejamos a seguir cada um dos tipos de primitivas disponíveis.

3.2.4.1 Aritmética

Aritmética é o ramo da matemática que trata das propriedades elementares de certas operações sobre numerais. Estão incluídas na linguagem *DHJOG* as operações básicas de adição, subtração, multiplicação e divisão. Estes operadores podem ser utilizados na definição de uma expressão de cálculo heurístico, na alteração dos valores iniciais das peças, na alteração do valor heurístico calculado de uma situação de jogo, ou mesmo no código interno de uma função definida.

3.2.4.2 Geométricas

A Geometria é uma área da matemática dedicada ao estudo das formas planas e espaciais e a análise de suas propriedades. É possível no processo de formalização de heurísticas de jogos definir

regiões espaciais a serem utilizadas como entradas para funções de identificação de contextos. Uma região é formada por um conjunto de posições existentes no jogo as quais podem estar ocupadas por peças. Podemos definir uma região através do uso de primitivas geométricas tais como retângulo, elipse ou ponto. Por exemplo, um jogador de Xadrez pode especificar que a região centro do tabuleiro é definida pelo retângulo formado pelos pontos (Casas) $\{c6,f6,c3,f3\}$. A Figura 3.1.3.3 apresenta a região definida.

3.2.4.3 Lógicas

A lógica matemática consiste no uso da lógica formal para o estudo do raciocínio matemático, ou como proposto por Alonzo Church é o estudo da lógica tratada pelo método matemático (Church 1956). Na linguagem *DHJOG* estão inseridos primitivas de uma sub-área da lógica matemática chamada Lógica de Predicados. Esta sub-área permite formalizar e identificar aspectos de relacionamento entre conhecimentos expressos como sentenças.

O conceito de Regras de Produção é o fundamento da especificação de heurísticas para identificação de contextos em uma situação de jogo (Veja Seção 3.2.1). Este conceito usa o operador lógico SE-ENTÃO como chave essencial na formalização do conhecimento tratado.

3.2.4.4 Algébricas

Na matemática a Álgebra é o ramo que estuda as generalizações dos conceitos e operações de aritmética. Esta linha de estudo possui diversas sub-áreas tais como a Álgebra Elementar⁴ e a Álgebra Abstrata⁵. Iremos buscar dentro da Álgebra Abstrata o conceito de Álgebra Booleana e o

⁴ Área da Álgebra que estuda as equações e os métodos de resolvê-las.

⁵ Área da Álgebra que estuda estruturas algébricas como grupos, anéis e corpos.

utilizaremos no processo de construção das expressões condicionais. Estas expressões são as responsáveis pela ativação das regras de produção (Veja Seção 3.2.1 e 3.2.3).

A Álgebra Booleana é um conjunto de estruturas algébricas que definem o comportamento das operações lógicas E, OU e NÃO em expressões que retornam apenas o valor Verdadeiro ou Falso. Esses operadores serão utilizados como conectivos entre diferentes avaliações relacionais dentro de uma mesma expressão condicional. Vejamos na Figura 3.2.4.4 um exemplo do uso de operadores booleanos na definição da expressão condicional de uma Heurística de Valor de Jogo. Especificada para o Xadrez e formalizada como uma Regra de Produção.

```

HEURISTICA VALOR_JOGO "DOMÍNIO DE CENTRO"
SE
  (QUANTIDADE_LANCES() < 10) E
  (SOMA_PECAS(EU, CENTRO) > SOMA_PECAS(OPONENTE,CENTRO)) E
  NÃO (ESTA_AMEACADA(DAMA,EU))
ENTÃO
  TABULEIRO.VALOR <- TABULEIRO.VALOR * 1,2
FIM HEURISTICA

```

Figura 3.2.4.4 – Exemplo de definição de Heurística de Valor de Jogo para Xadrez

3.2.5 Nível de Expressividade

As funcionalidades de formalização de heurísticas de jogos disponíveis na linguagem *DHJOG* foram agrupadas em níveis de expressividade. Cada nível representa uma faixa de conhecimento sobre: conceitos heurísticos, sobre a forma de processamento das heurísticas em um jogador automático, e sobre os elementos da linguagem. Além de cada faixa representar um nível de conhecimento necessário, indo do básico ao avançado, também podemos associar a cada faixa o esforço cognitivo necessário para desempenhar as tarefas de formalização de heurísticas. Quanto

maior o nível, mais conhecimento é necessário ao Autor e mais recursos da linguagem estarão disponíveis para serem utilizados. Vejamos as vantagens resultantes desta abordagem:

- Permitiu-se realizar uma análise profunda sobre os elementos da linguagem, a relação entre eles, o grau de complexidade, e de expressividade de cada um.
- Um autor iniciante poderá desde o primeiro momento utilizar a linguagem para a criação de heurísticas simples de um jogo, e acompanhar o desempenho destas heurísticas em partidas automáticas.
- A ferramenta de edição de heurísticas poderá permitir ao usuário acomodar a interface ao seu conhecimento e experiência na criação de heurísticas. Isto possibilitará melhor utilização da aplicação tanto para usuários iniciantes quanto para avançados.

Vejamos na Tabela 3.2.5 os níveis de expressividade criados para a linguagem *DHJOG*.

| Nível | | Descrição da Expressividade Permitida |
|-------|---------------|--|
| 1 | Iniciante | Neste nível o aprendiz praticará a alteração dos valores iniciais de cada tipo de Peça. |
| 2 | Básico | Neste nível o aprendiz poderá definir regiões e criar Heurísticas de Valor de Peça. |
| 3 | Intermediário | Neste nível o aprendiz poderá criar Heurísticas de Valor de Jogo. |
| 4 | Pleno | Neste nível o aprendiz poderá definir novas Etapas e criar os critérios de transições entre elas (Heurísticas de Transição de Etapas). |
| 5 | Avançado | Neste nível o aprendiz poderá alterar as Expressões de Cálculo Heurísticos de cada Etapa criada. |
| 6 | Especialista | Neste nível o autor poderá alterar e codificar novas funções. |

Tabela 3.2.5 – Níveis de Expressividade da Linguagem *DHJOG*

3.2.6 Apoio a Autoria e Colaboração

No processo de definição da linguagem *DHJOG* além dos aspectos relacionados aos conceitos sobre heurísticas de jogos, e de funcionamento de um jogador automático, foi analisada também a questão de que o conhecimento heurístico formalizado será compartilhado entre grupos de aprendizes, através da arquitetura de ensino proposta focada na alternância entre colaboração e competição (Martineschen 2006). Foi constatada então a necessidade de equipar a linguagem com mecanismos que permitam: o controle da autoria dos componentes criados; e o registro textual de anotações, comentários, e explicações criados pelo Autor.

Cada um dos componentes da *DHJOG* (Conjunto Heurístico, Etapa, Heurística de Transição de Etapa, Heurística de Valor de Peça, Heurística de Valor de Jogo, Expressão de Cálculo Heurístico, Região e Função) possuem atributos que indicam quem foi o Autor do elemento, datas de criação e modificação, controle de versões, e podem ainda possuir anotações textuais relacionadas a eles.

As anotações textuais permitem uma maior expressividade do Autor no momento de formalizar o conhecimento heurístico, e ao leitor da heurística formalizada uma melhor compreensão sobre os conceitos abordados. Já o controle de versões permite que instrutores e autores analisem a utilização dos elementos criados pela comunidade de aprendizes. Isto possibilita implantar na arquitetura de ensino funcionalidades para a promoção/estímulo dos autores mais influentes.

3.2.7 Fluxo de Execução de um Conjunto Heurístico

A avaliação de uma situação de jogo por um conjunto heurístico segue os seguintes passos:

- 1) O jogador automático passa a representação completa de uma possível situação de jogo para o conjunto heurístico qualificar numericamente.
- 2) Verifica-se qual é a Etapa atual do conjunto Heurístico.
- 3) Realiza-se a avaliação seqüencial das Heurísticas de Transição de Etapa associadas à Etapa corrente. Caso uma destas heurísticas seja ativada, uma nova Etapa será configurada como ativa, e volta-se para o passo 2.
- 4) São inicializados os valores de cada tipo de peça do jogo para a Etapa corrente.
- 5) Realiza-se a avaliação seqüencial das Heurísticas de Valor de Peça. Para cada heurística deste tipo ativada, um ou mais tipos de peças terão seu valor heurístico incrementado.
- 6) Realiza-se o cálculo do valor heurístico da situação atual de jogo através da execução da Expressão de Cálculo Heurístico.
- 7) Realiza-se a avaliação seqüencial das Heurísticas de Valor de Jogo. Para cada heurística deste tipo ativada, o valor heurístico previamente calculado (passo 6) será incrementado.
- 8) Devolve-se para o jogador automático o valor heurístico final para a situação de jogo avaliada.

A partir do valor heurístico retornado pelo Conjunto Heurístico o jogador automático pode verificar o quanto esta situação de jogo é vantajosa ou não para o jogador. E analisando-se um conjunto de situações de jogo possíveis, pode-se então concluir qual a melhor jogada a ser realizada. Sempre buscando maximizar sua vantagem e minimizar a vantagem do oponente.

3.2.8 Limitações da Linguagem

A linguagem *DHJOG* é um primeiro passo na formalização de conhecimento heurístico de

jogos. Ela propicia ao autor o ferramental necessário para a especificação de heurísticas para jogos por turno. A heurística especificada poderá posteriormente ser utilizada por um jogador automático na simulação de partidas. O escopo da linguagem nesta primeira versão é a análise de apenas uma situação de jogo dentro de uma partida. Ela qualificará como vantajosa ou não ao jogador esta situação através do valor heurístico retornado.

Durante o decorrer de uma partida um jogador pode adotar estratégias que para serem realizadas serão necessárias várias jogadas seguidas. E a questão é: Será possível identificar uma estratégia de médio ou longo prazo analisando apenas uma situação de jogo? Na maioria dos casos não. Tanto isto é verdade que os materiais de ensino de táticas de jogo sempre apresentam pedaços de partidas (vários lances) para ensinar estes tipos de estratégias. Ou seja, é necessário acrescer à linguagem *DHJOG* em suas próximas versões o suporte a este tipo de análise, que no primeiro momento foi desconsiderada por tratar conceitos heurísticos complexos para aprendizes iniciantes. Vejamos outros fatores que não são suportados pela linguagem, e que poderão ser incluídos em versões futuras:

- Arriscar uma jogada ou estratégia. Contar com o elemento sorte.
- Simular um erro para desviar a atenção do adversário.
- Criar heurísticas baseadas no comportamento do adversário. Poder analisar um histórico dele de jogadas e partidas e tentar prever próximos lances.
- A heurística não consegue realizar a análise suficiente para poder desistir de uma partida, ou mesmo forçar um empate.
- Possibilidade de criação de heurísticas para a identificação de estratégias adotadas pelo adversário, para poder contra-atacá-las antecipadamente.

No Apêndice A é apresentada a especificação da linguagem *DHJOG*, e no Apêndice B a extensão realizada nela para adaptá-la ao contexto do jogo de Xadrez.

4 CAPÍTULO

ANÁLISE DO AMBIENTE DE ENSINO PROPOSTO - CACAREJE

Neste capítulo iremos analisar o projeto e implementação de um sistema de ensino de conceitos heurísticos de Xadrez, focado na alternância da colaboração e da competição. O sistema CACAREJE (Colaboração Alternada com Competição na Aprendizagem Referenciada por Jogos Educativos) é composto por uma base de dados de heurísticas de jogos, um comparador genérico de heurísticas (CGDH), de um escalonador de partidas (EGPA), e de interfaces de acesso ao sistema. Neste ambiente de ensino o aprendiz irá formalizar conhecimento heurístico do jogo de Xadrez, inscrever as heurísticas definidas em competições automáticas entre heurísticas (normalmente criadas por instrutores), acompanhar o desempenho delas nestes torneios, e trocar conhecimento formalizado com os demais aprendizes. Este sistema está inserido no projeto *PROTEX*, de apoio computacional ao ensino de Xadrez nas escolas brasileiras, apresentado em (Direne *et al.* 2004).

Os estudos realizados neste trabalho de mestrado estão integrados à definição e ao desenvolvimento do sistema *CACAREJE*. Principalmente na formalização dos conhecimentos heurísticos de jogos, e na definição das representações externas mais apropriadas à ferramenta de edição. O trabalho de mestrado de Daniel Martineschen (2006) realiza uma análise completa sobre o conceito da alternância da colaboração e da competição no ensino de conceitos heurísticos. Recomendamos a leitura deste estudo.

Na seção 4.1 iremos apresentar a arquitetura funcional do sistema, os módulos que o compõem e a relação entre eles. Na seção 4.2 iremos apresentar um cenário completo da utilização do ambiente de ensino. Na seção 4.3 iremos analisar o protótipo da ferramenta de edição de heurísticas, com foco nas técnicas de representações externas adotadas. E por fim na seção 4.4 iremos discutir os próximos passos do desenvolvimento.

4.1 Arquitetura Funcional

O sistema *CACAREJE* foi concebido seguindo o modelo Cliente-Servidor. Sua arquitetura é formada por módulos internos, responsáveis pelas tarefas inerentes ao próprio sistema, e por módulos externos, responsáveis pela comunicação entre o sistema e os usuários. Iremos analisar nas próximas seções cada um dos itens desta arquitetura. A Figura 4.1 apresenta a estrutura completa do sistema.

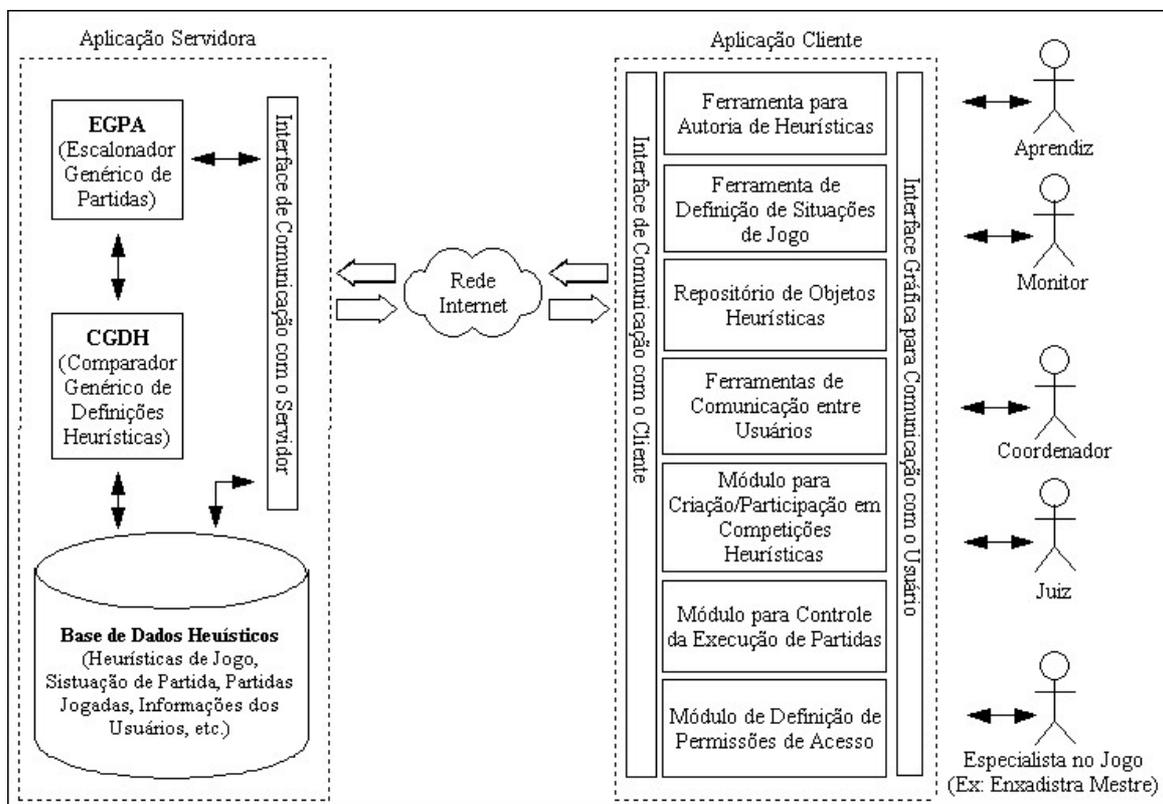


Figura 4.1 – Arquitetura Cliente-Servidor do Sistema *CACAREJE*

4.1.1 Aplicação Servidora

O servidor será responsável por centralizar o armazenamento dos dados, a troca de comunicações entre os usuários, gerenciar e executar as competições entre heurísticas, e controlar

as permissões de acesso dos usuários as tarefas e objetos do sistema (Martineschen 2006). Estará integrado com a infra-estrutura do servidor de Xadrez do CEX⁶, permitindo que os usuários deste servidor acessem as funcionalidades do ambiente *CACAREJE*.

4.1.1.1 Base de Conhecimento Heurístico

Este item corresponde à base de dados de todo o sistema, onde serão armazenadas as heurísticas e situações de tabuleiro formalizadas pelos aprendizes e instrutores, as informações completas das partidas realizadas nas competições automáticas, e as informações de preferências e de permissões de cada usuário.

4.1.1.2 CGDH – Comparador Genérico de Definições Heurísticas

Este módulo será o responsável por efetuar a comparação entre conhecimento heurístico de jogo definido pelos aprendizes. Isto será realizado através da simulação de partidas entre os Conjuntos Heurísticos formalizados. Ele utilizará uma variação do Algoritmo Minimax com podas Alfa-Beta para gerar um grafo de próximas possíveis jogadas, e as heurísticas definidas serão utilizadas como funções estáticas deste algoritmo para a escolha da melhor jogada a ser realizada. Será possível também configurar parâmetros relacionados à execução do algoritmo de simulação de jogo, tais como a altura da árvore, modalidade de partida, tempo de jogo, ordem de início da partida, etc. Veja o trabalho de Martineschen (2006) para maiores detalhes.

⁶ CEX – Centro de Excelência de Xadrez (www.cex.org.br)

4.1.1.3 EGPA – Escalonador Genérico de Partidas Automáticas

O módulo EGPA será o responsável pelo pareamento das heurísticas participantes de uma competição automática. É também o módulo que permitirá ao Coordenador e ao Juiz alterarem indiretamente os parâmetros de execução e até interromperem uma partida (Adjudicações⁷). Estas alterações são realizadas no módulo CGDH pelo EGPA.

Este módulo deverá suportar vários modelos de pareamento para partidas entre dois competidores. Na primeira versão estará disponível apenas o “*Pareamento Suíço*” e o “*Pareamento Simples*” (ou todos contra todos). Existem também diversas maneiras de condução de partidas (*Wild*, Sem limite de tempo, Relâmpago, etc.), mas inicialmente será implementado apenas o “Sem limite de tempo”.

4.1.2 Aplicação Cliente

Esta aplicação será a responsável por centralizar todo o acesso às funções e objetos do ambiente de ensino proposto *CACAREJE*. A mesma aplicação será utilizada pelos aprendizes, monitores, instrutores e juízes. No início de sua execução os usuários deverão entrar com seu identificador pessoal (*login*) e senha, e de acordo com as permissões definidas para este usuário e que a interface se adaptará para mostrar-lhe as funcionalidades permitidas.

As ferramentas de interface previstas para o sistema foram idealizadas com o objetivo de permitir e incentivar os aprendizes a aprender conceitos heurísticos de jogo, de forma colaborativa e competitiva. E possibilitarão ainda que estes possam verificar a evolução da aprendizagem a cada etapa, através da validação das heurísticas definidas em um ambiente real de competição. Os

⁷ Um exemplo de caso necessário de Adjudicação e quando as heurísticas geram jogadas cíclicas.

módulos previstos na aplicação cliente serão detalhados a seguir.

4.1.2.1 Ferramenta para Autoria de Heurísticas de Jogos

Este é o módulo principal de todo o ambiente proposto, permitirá que o usuário (aprendiz ou instrutor) formalize várias representações do seu conhecimento heurístico de um jogo, através do uso da linguagem de especificação de heurísticas *DHJOG* (Veja a Seção 3.2). Este editor utilizará recursos de Representação Externa para adaptar a interface ao nível de conhecimento do usuário, e permitirá que ele próprio altere a profundidade de detalhes que deseja definir na sua heurística (Veja Seção 3.2.5). O usuário ainda poderá realizar anotações nas heurísticas formalizadas por ele, ou por outros usuários. Desde que possua permissão para isto. A Seção 4.3 apresenta um primeiro protótipo desenvolvido deste módulo, especificamente para o jogo de Xadrez.

4.1.2.2 Ferramenta para Definição de Situações de Jogo

Neste módulo o usuário poderá formalizar situações específicas de um jogo a serem utilizadas na avaliação do comportamento das heurísticas. Esta avaliação será realizada de forma estática, onde apenas se verifica o valor retornado pela heurística para o momento de jogo definido. Será possível ao usuário posicionar as peças de cada jogador dentro do espaço do jogo, e realizar anotações que ajudem a explicar o contexto especificado.

4.1.2.3 Repositório de Objetos Heurísticos

Cada usuário poderá manter uma coleção particular de objetos heurísticos. Estes objetos podem ser: heurísticas de jogo formalizadas, situações específicas de jogo, partidas jogadas entre heurísticas, e funções definidas pelo aprendiz. Estes objetos serão utilizados como material de estudo e servirão de base para novas definições heurísticas. Todos estes itens armazenados poderão ser trocados entre os aprendizes. Com isto pretende-se estimular a colaboração entre usuários do sistema.

4.1.2.4 Ferramentas de Comunicação entre Usuários

É previsto que o ambiente de ensino implemente futuramente mecanismos para a comunicação entre os usuários do sistema. Eles poderão trocar mensagens entre si, de forma direta para um determinado usuário, ou aberta à comunidade através de grupos de discussão. Este estímulo à comunicação visa fomentar dentro da comunidade de aprendizes a colaboração e discussão dos temas abordados durante o processo de ensino.

4.1.2.5 Módulo para Criação/Participação em Competições Heurísticas

Através desta interface um usuário poderá inscrever suas heurísticas em competições automáticas entre heurísticas, e acompanhar o desempenho delas em relação às dos outros competidores. Alternativamente, o próprio aprendiz poderá criar um novo campeonato, exclusivamente seu, a ser realizado entre as heurísticas de sua própria coleção, ou também aberto a outros usuários (isto depende das permissões cedidas pelo instrutor e pelo aprendiz). A partir deste

módulo de competição é que instrutores e aprendizes poderão de forma objetiva avaliar e validar os conhecimentos heurísticos aprendidos, a cada etapa do processo de ensino.

4.1.2.6 Módulo para Controle da Execução de Partidas

Neste módulo o juiz poderá acompanhar e controlar o andamento das partidas em uma competição automática entre heurísticas. Este tipo de usuário será o responsável por zelar pelo bom andamento das competições, e poderá intervir em partidas em que ocorram anomalias durante a realização das mesmas. Como no caso de ciclicidade de jogadas. Ele terá o poder de suspender ou adjudicar uma partida.

4.1.2.7 Módulo para Definição de Permissões de Acesso

O coordenador poderá definir as permissões de acesso às tarefas do sistema pelos aprendizes de forma coletiva, ou individualizada. Por exemplo, o coordenador pode definir se aprendizes poderão realizar anotações em heurísticas de outros alunos, se poderão criar competições abertas aos demais usuários, ou se poderão definir eles mesmos as permissões de acesso que os outros usuários terão sobre os seus objetos. Por outro lado, o coordenador poderá também retirar permissões de alguns aprendizes que utilizarem as ferramentas colaborativas de forma inapropriada.

O coordenador poderá nomear usuários como monitores de uma turma de aprendizes. Um monitor é um aprendiz com um bom conhecimento do processo de definição de heurísticas, e no uso das ferramentas do sistema de ensino. Ele deverá realizar atividades repassadas pelo coordenador, e auxiliar os aprendizes nas atividades de ensino.

4.2 Exemplo de Cenário Completo de Utilização

Nesta seção será descrito um cenário completo de utilização do sistema *CACAREJE* e os atores envolvidos neste processo. Será abordado o ensino de conceitos heurísticos do jogo de Xadrez, que deverá ser o primeiro jogo suportado pelo ambiente de ensino proposto. Esta simulação visa exemplificar o uso das funcionalidades projetadas para a ferramenta.

4.2.1 Descrição dos Atores

Descrição dos atores definidos no cenário:

- **Coordenador:** Pessoa responsável por ensinar Xadrez a uma turma específica de aprendizes, e que utilizará o Ambiente de Ensino de Heurísticas *CACAREJE* neste processo.
- **Aprendiz:** Aluno que está participando do curso de Xadrez e que está utilizando o Ambiente de Ensino de Heurísticas, sobre orientação de um Coordenador.
- **Monitor:** Aluno com maior experiência da definição de heurísticas e no uso das ferramentas do sistema, e que foi nomeado pelo Coordenador como seu auxiliar no processo de ensino.
- **Juiz:** Usuário responsável pelo bom andamento da execução das partidas em uma competição automática. Pode suspender e até adjudicar uma partida.
- **Internauta:** Pessoa que através do servidor público de Xadrez do projeto CEX (Centro de Excelência de Xadrez) (CEX 2005) está acompanhando a realização de uma partida de Xadrez entre Heurísticas.

4.2.2 Sequência de Ações

É mostrado a seguir uma hipotética seqüência de ações realizadas por usuários do sistema *CACAREJE*.

- 1) **Coordenador1**: Acessa a aplicação cliente e entra com seu identificador e senha. Cria algumas heurísticas de Xadrez na Ferramenta de Autoria para serem usadas como exemplo pelos Aprendizes, e algumas situações de tabuleiro para que os alunos possam testar estas heurísticas.
- 2) **Coordenador1**: Configura as permissões necessárias para que os Aprendizes possam incluir anotações nas heurísticas criadas por ele, e copiá-las para os repositórios particulares de objetos.
- 3) **Coordenador1**: Cria um novo campeonato entre heurísticas definindo a forma de execução do campeonato (por exemplo, “mata-mata”, eliminatória dupla, etc.), e os parâmetros para a execução do algoritmo de Minimax (por exemplo, a altura da árvore de busca). Especifica a data limite de inscrições no campeonato, a data da sua realização, quem poderá participar (por exemplo, aberto a todos os aprendizes), redige uma descrição sobre este evento, e finalmente publica estas informações automaticamente para todos os Aprendizes.
- 4) **Aprendiz1**: Acessa a aplicação cliente e entra com seu identificador e senha. Lê o aviso do campeonato aberto. Manda uma mensagem para o Monitor da sua turma pedindo ajuda para começar a formalizar uma Heurística.
- 5) **Monitor1**: Lê a mensagem enviada pelo Aprendiz1 e entra em contato direto com ele lhe passando um roteiro de como acessar mais material de estudo, e quais passos deve fazer para definir uma nova heurística de jogo.

- 6) **Aprendiz1**: Segue o roteiro passado pelo Monitor1. Primeiramente abre o repositório de objetos do Coordenador1, escolhe uma das heurísticas ali colocadas e a copia para poder analisá-la na Ferramenta de Autoria de Heurísticas. Inclui nesta heurística algumas anotações suas e a testa em umas das situações de tabuleiro já criadas pelo Coordenador1.
- 7) **Aprendiz2**: Acessa a aplicação cliente e entra com seu identificador e senha. Lê o aviso do campeonato aberto. Abre a ferramenta de Autoria de Heurísticas, configura o nível de profundidade que deseja utilizar da linguagem de modelagem (isto afetará as Representações Externas apresentadas). Cria uma nova heurística de Xadrez e define algumas situações particulares de tabuleiro. Testa a heurística com estas situações e, por fim, a inscreve no campeonato aberto pelo Coordenador1.
- 8) **Aprendiz3**: Cria várias heurísticas distintas de Xadrez. Define vários campeonatos de Xadrez entre suas próprias heurísticas. Durante estes campeonatos ele vai alterando configurações do Minimax e da ordem de início das partidas para poder verificar quais heurísticas estão ganhando independente destes atributos. Altera algumas heurísticas combinando características e reinicia o processo de teste. Ao final ele consegue identificar qual é a sua melhor heurística e a inscreve no campeonato aberto pelo Coordenador1.
- 9) **Coordenador1, Aprendiz1, Aprendiz2, Aprendiz3, Monitor1 e Juiz1**: Acompanham as partidas realizadas durante o campeonato através da aplicação cliente.
- 10) **Internauta1**: Acompanha ao vivo a realização de uma partida de Xadrez entre heurísticas, através do servidor público de Xadrez do CEX.
- 11) **Juiz1**: Identifica que em uma das partidas as heurísticas estão realizando jogadas cíclicas. Decide reinicia-la alterando a heurística que jogará com as brancas.

- 12) **Coordenador1, Monitor1**: Após o encerramento do campeonato realiza anotações nas heurísticas participantes, e utiliza estas heurísticas e as informações das partidas realizadas como material expositivo para uma aula presencial com os Aprendizes.
- 13) **Internauta2**: Analisa as jogadas realizadas durante uma partida de Xadrez entre heurísticas através do servidor público de Xadrez do CEX.
- 14) **Aprendiz2**: Abre o repositório de objetos do Coordenador1, do Aprendiz1 e do Aprendiz3 para estudar as características das melhores heurísticas criadas por eles, e que se saíram melhor do que a do Aprendiz2 no campeonato. Realiza anotações nelas, e copia algumas para o seu repositório particular.
- 15) **Aprendiz2**: Cria uma nova heurística combinando as melhores características estudadas, valida ela em uma competição interna, realizada apenas com as heurísticas do seu repositório de objetos.
- 16) **Aprendiz2**: Cria um novo campeonato aberto a todos os aprendizes, e já inscreve nele a sua nova heurística de Xadrez.

4.3 Protótipo da Ferramenta de Autoria de Heurísticas

Nesta seção iremos discutir as técnicas e representações externas adotadas no desenvolvimento do protótipo da Ferramenta de Autoria de Heurísticas. Esta ferramenta permitirá ao aprendiz formalizar e acessar conhecimento heurístico de um jogo. Nesta primeira implementação o jogo alvo é o Xadrez. O conhecimento será especificado através da linguagem para definição de conhecimentos heurísticos *DHJOG* (Veja a Seção 3.2).

No projeto desta ferramenta pretende-se atender aos seguintes objetivos:

- Validar o uso da linguagem *DHJOG* no processo de formalização de conhecimento heurísticos de jogos.
- Permitir que esta implementação sirva de base para o desenvolvimento da versão final do editor de conhecimento heurístico.
- Prover a interface do editor de mecanismos que auxiliem o autor durante o processo de formalização de heurísticas. Diminuindo o esforço cognitivo necessário para desempenhar as tarefas.
- Permitir a adaptação da interface ao nível de conhecimento e experiência do usuário sobre o processo de formalização de heurísticas. Isto propiciará uma maior Usabilidade⁸ da ferramenta, tanto para usuários iniciantes quanto para experientes.

Vamos analisar a seguir cada uma das técnicas utilizadas no desenvolvimento.

4.3.1.1 Adaptação de Interface ao Nível de Conhecimento do Usuário

Na linguagem *DHJOG* foram especificados níveis de expressividade. Cada nível indica uma faixa de conhecimento heurístico, de esforço cognitivo, e de elementos de formalização heurística. Quanto mais alto o nível, maior conhecimento e experiência são necessários ao autor, e maior será o poder de representação heurística disponível. (Veja a Seção 3.2.5).

O autor, no momento da criação de um novo Conjunto Heurístico, deverá definir qual o nível de expressividade pretende utilizar na edição. A interface da ferramenta irá adaptar-se para prover apenas funcionalidades compatíveis ao nível de expressividade selecionado. Esta adaptação incluirá ações como ocultar, exibir ou adaptar as funções do sistema e as representações externas

⁸ Usabilidade indica o grau de efetividade, eficiência e satisfação, que um produto propicia a ser utilizado por um grupo específico de usuários para desempenhar uma tarefa específica. (ISO 9241-11)

utilizadas.

É possível fazer uma analogia entre a utilização dos níveis de expressividade e a técnica apresentada por Swidersk (2001). Ele propõe em seu estudo a criação de uma ferramenta de controle de conteúdo multimídia em sistemas hipermídia. (Veja a Seção 2.1.1). Onde o objetivo é disponibilizar ao usuário apenas representações externas e funcionalidades confortáveis ao nível de conhecimento atual do usuário, dentro do domínio abordado pela aplicação.

A interface do editor apresenta no seu canto superior direito um item chamado “Nível de Complexidade”, este item exibe o nível de expressividade da linguagem *DHJOG* atualmente em uso. O termo complexidade foi utilizado em detrimento do termo expressividade por ser mais compreensível à usuários iniciantes. A Figura 4.3.1.1 apresenta a interface do Editor de Heurísticas e o item discutido.

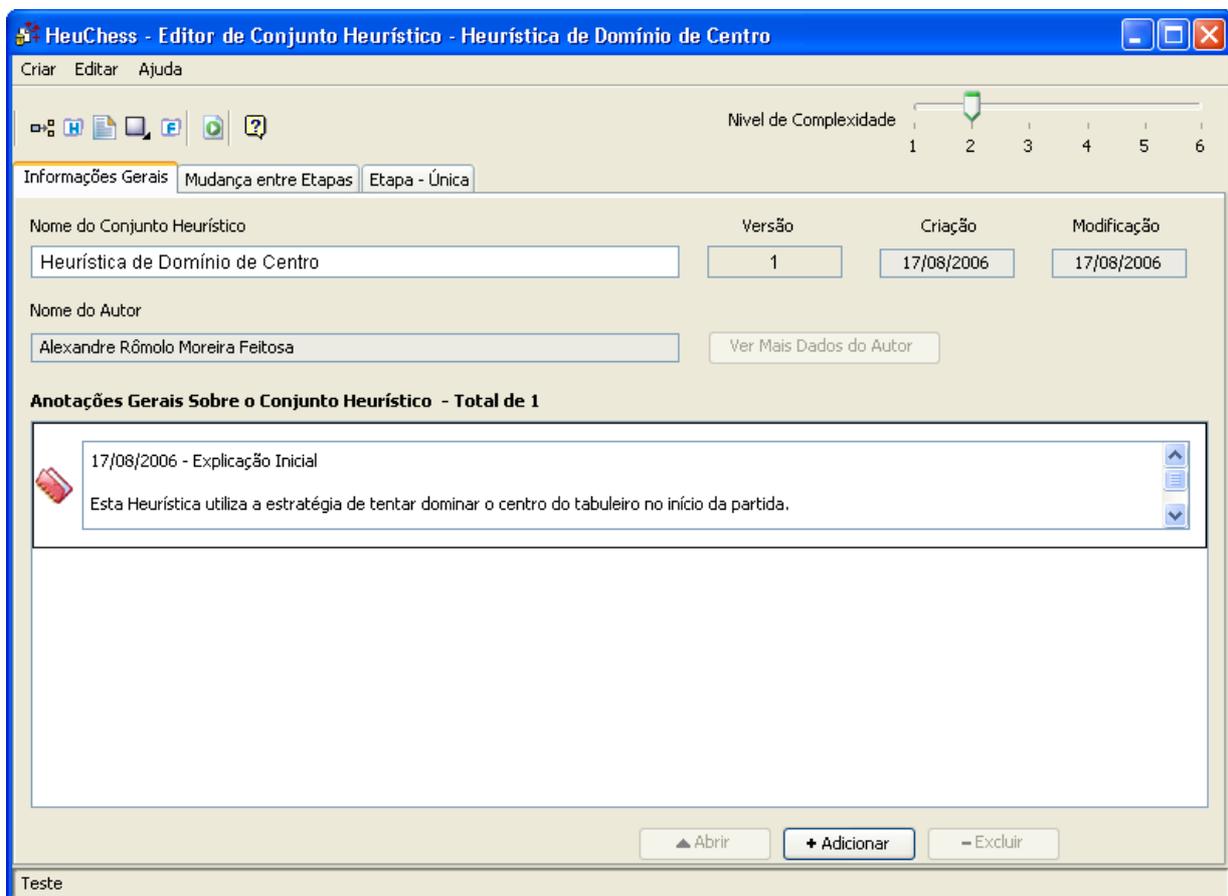


Figura 4.3.1.1 – Nível de Complexidade de um Conjunto Heurístico.

O autor poderá a qualquer momento alterar o nível de complexidade da heurística atualmente aberta no editor. O simples fato de alterar este nível fará primeiramente com que a interface questione o usuário sobre esta ação, e caso ele confirme a intenção de fazê-la a interface será adaptada para contemplar o novo nível de expressividade selecionado. Isto possibilita que o aprendiz possa avançar para o próximo nível no instante em que se sentir confortável com o conhecimento estudado até o presente momento.

4.3.1.2 Utilização de Facilitadores de Tarefas - *Wizards*

A formalização de Heurísticas de Jogos usando a linguagem *DHJOG* é um processo que compreende a especificação de vários elementos, tais como: Conjunto Heurístico, Etapa, Heurística, Anotação, Função e Região. A ferramenta de autoria de Heurísticas possui interfaces para auxiliar o autor no processo de criação, alteração e exclusão destes itens. Estas interfaces foram projetadas com o enfoque de direcionar o usuário durante toda a realização da tarefa, especificando um caminho a ser seguido, oferecendo apenas as opções de ações válidas para o contexto atual, e validando automaticamente as entradas feitas pelo usuário. Interfaces que seguem esta linha de desenvolvimento são comumente chamadas de “*Wizards*”. A Figura 4.3.1.2 apresenta a tela de criação de um novo Conjunto Heurístico.

O uso desta abordagem no projeto das interfaces de criação de itens propicia uma maior segurança ao usuário na realização da tarefa, a necessidade de menor esforço cognitivo no processo, e o entendimento dos passos a serem realizados. Podemos constatar que a própria interface de edição funciona como uma abstração do conteúdo a ser editado. Sendo uma representação externa do elemento, que deixa transparente ao usuário como aquele elemento será realmente codificado na representação interna⁹ do sistema.

⁹ A representação interna compreende os mecanismos utilizados para especificar a informação dentro da aplicação.



Figura 4.3.1.2 – Tela de criação de um novo Conjunto Heurístico

4.3.1.3 Definição Visual de Regiões

Na linguagem *DHJOG* é possível especificar uma lista de posições de jogo chamada Região. Esta lista pode ser utilizada como parâmetro para funções de identificação de contextos. Na ferramenta de autoria de heurísticas foi criada uma interface específica para a definição visual de Regiões. Nesta interface o autor poderá utilizar ferramentas que seguem o padrão de aplicações de editoração gráfica, tais como:

- *Retângulo*: Permite desenhar um retângulo no tabuleiro onde as casas que estão inclusas nesta figura serão incluídas na região.
- *Lápis*: Permite marcar uma a uma casas do tabuleiro como se fossem pontos, cada casa marcada será incluída na região.
- *Borracha*: Apaga uma casa marcada pela ferramenta lápis ou um retângulo desenhado no tabuleiro, eliminando as casas da região definida.

- *Seleção*: Permite selecionar uma figura desenhada no tabuleiro (retângulo ou ponto) e move-la para outra posição.

A Figura 4.3.1.3-1 apresenta a interface de criação visual de regiões.

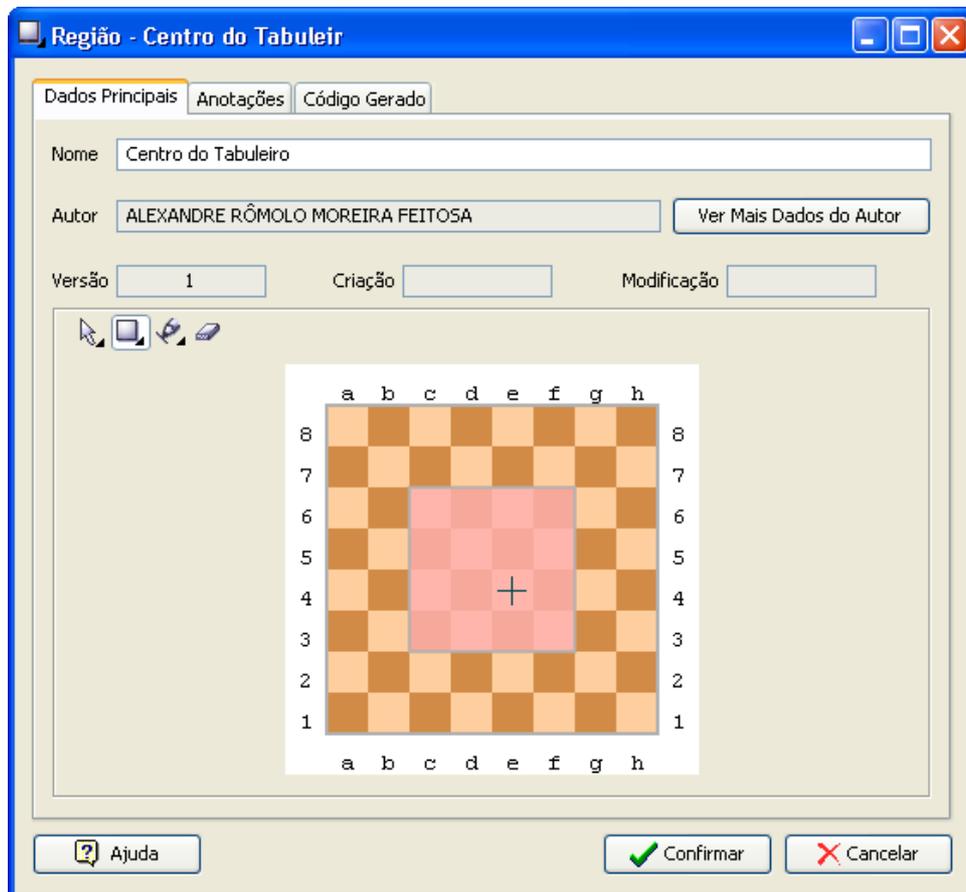


Figura 4.3.1.3-1 – Tela de Definição de Regiões

Como é possível criar várias regiões a serem utilizadas pelas heurísticas formalizadas, foi necessário também projetar um mecanismo que permite ao usuário verificar quais regiões foram definidas até o momento. A técnica escolhida segue o mesmo princípio das ferramentas de editoração gráfica. É apresentado ao autor a figura de um tabuleiro de Xadrez onde as regiões definidas são desenhadas sobre ele, cada região com uma cor específica. Como é possível a existência de regiões que se sobreponham, cada região é desenhada usando o efeito de semi-transparência, o usuário tem a opção de movimentá-las para a frente (cima) ou para trás (fundo), e

ele ainda pode definir se uma região estará visível ou não. A Figura 4.3.1.3-2 apresenta a interface principal de edição de uma Etapa onde é possível visualizar as regiões definidas pelo Autor.

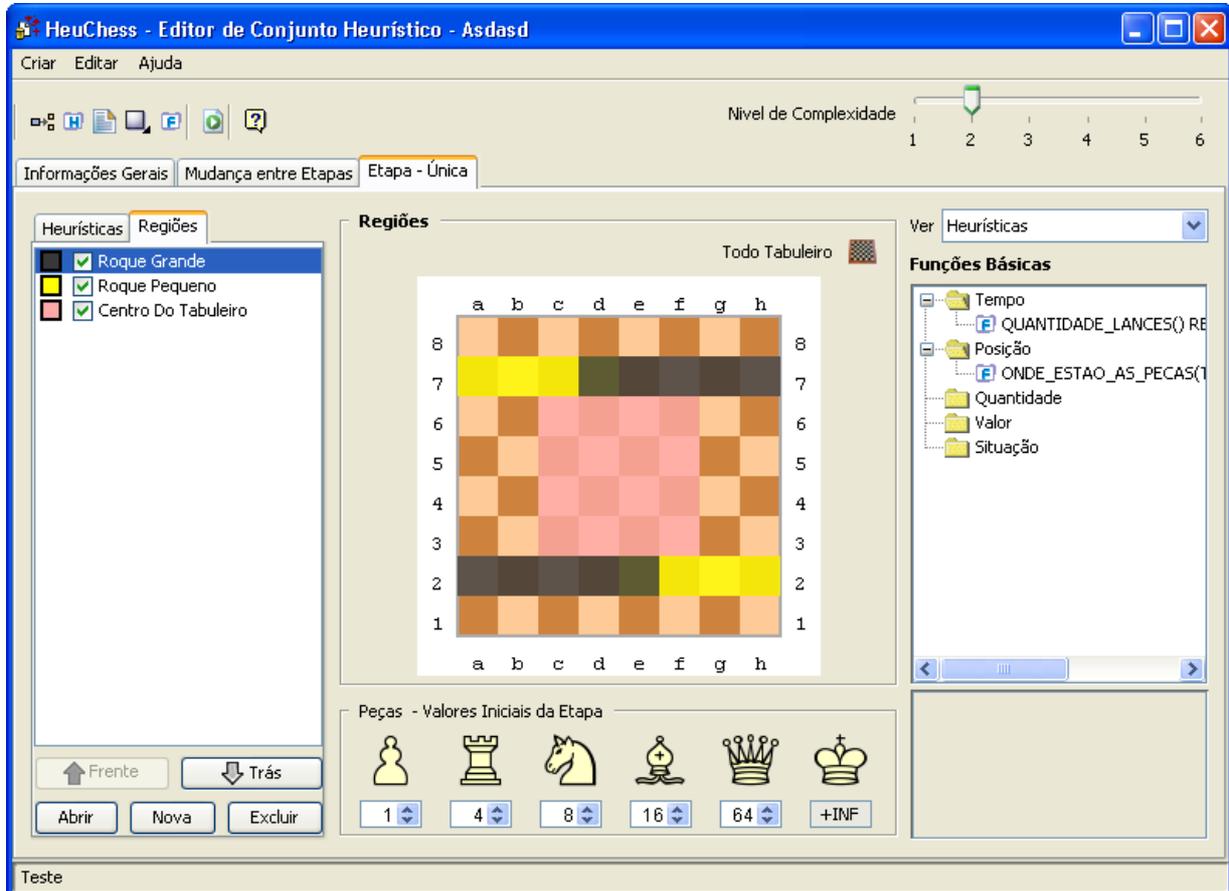


Figura 4.3.1.3-2 – Tela de Edição de Etapas e de Visualização de Regiões

Com esta abordagem para a manipulação de regiões o aprendiz poderá a princípio entender este processo apenas como o desenho (marcação) de posições na tabuleiro. O fato de que uma Região significa um conjunto matemático de casas não precisa ser tratado em um primeiro momento. Conforme sua experiência na ferramenta e seu conhecimento sobre definição de heurísticas aumente, ele poderá mudar o nível de expressividade do conjunto heurístico, e utilizar outras representações externas para entender a formalização de uma Região.

4.3.1.4 Agrupamento de Funções por Categoria

A definição de uma nova Heurística tem como primeiro passo a construção das condições de ativação da mesma. Esta atividade é auxiliada por interfaces gráficas orientadas a facilitar a tarefa (Veja a Seção 4.3.1.2). Neste processo o autor precisará escolher funções de identificação de contextos, e a fim de auxiliá-lo nesta ação as funções foram agrupadas em categorias. Cada categoria é um grupo formado por funções que identificam o mesmo tipo de contexto de situação de jogo (Veja a Seção 3.1.4). Vejamos as categorias definidas:

- *Tempo*: Funções que identificam contextos Temporais.
- *Posição*: Funções que identificam contextos Posicionais
- *Quantidade*: Funções que identificam contextos Quantitativos.
- *Valor*: Funções que identificam contextos Qualitativos.
- *Situação*: Funções que identificam contextos Comportamentais.

A Figura 4.3.1.3-2 apresenta no seu lado direito uma árvore de funções onde cada categoria de função corresponde a uma pasta específica.

4.3.1.5 Diagrama de Transição entre Etapas

No momento da criação do Conjunto Heurístico uma Etapa padrão será associada automaticamente a ele, pois um conjunto heurístico obrigatoriamente precisa ter pelo menos uma Etapa. Conforme o aprendiz vá assimilando o conhecimento sobre a definição de heurísticas, ele poderá aumentar o nível de expressividade a ser utilizado. Quando o nível for igual ou superior a quatro (Pleno), ele poderá definir novas etapas para o Conjunto Heurístico e Heurísticas de

Transições entre elas (Veja a Seção 3.2.5).

Com o intuito de facilitar a compreensão destes conceitos, foi definida uma aba específica no editor de Heurísticas para apresentar uma representação externa das transições definidas entre as Etapas. Esta representação é na forma de um diagrama de máquina de estados, onde temos um símbolo para indicar o estado inicial (elipse), setas para indicar as transições (a ponta da seta indica o estado destino), e vértices que indicam os estados possíveis (retângulos). A Figura 4.3.1.5 apresenta a aba de edição de Etapas.

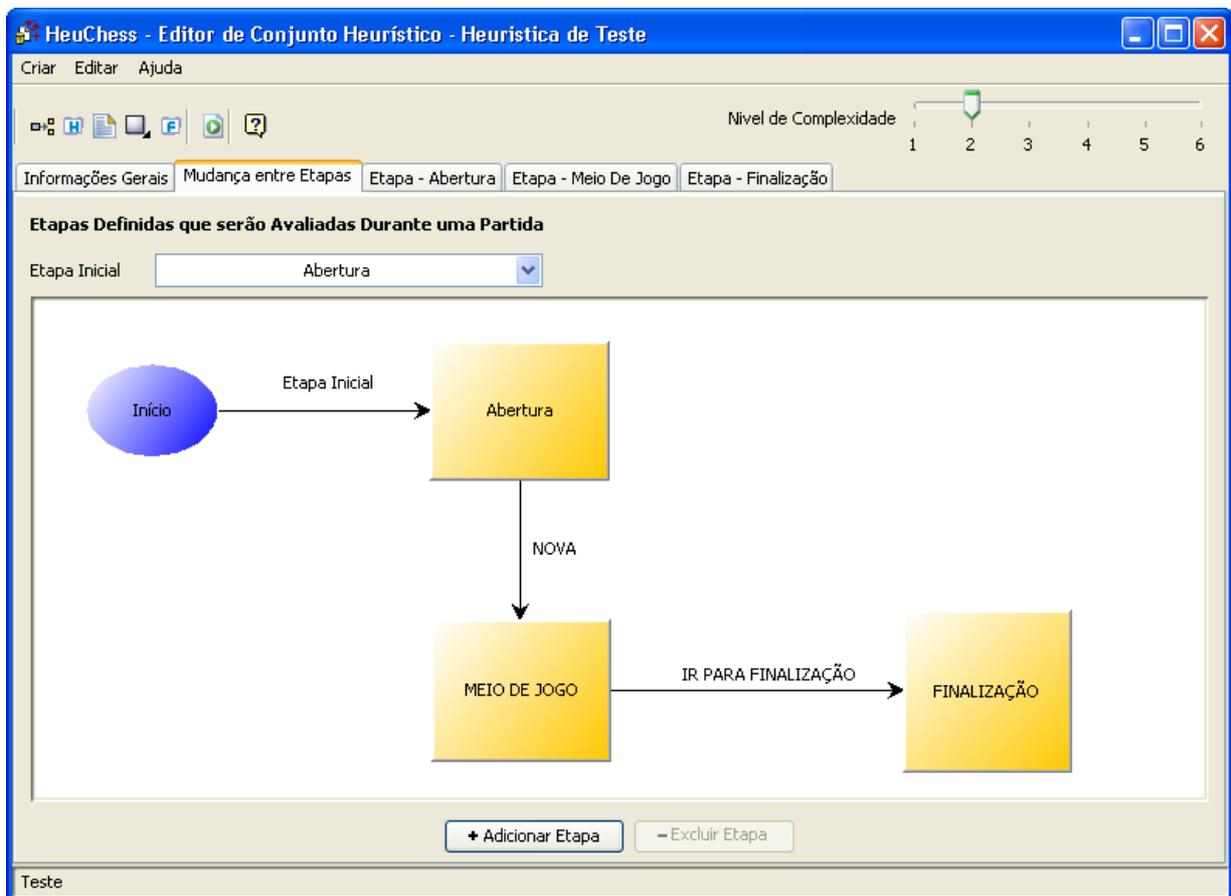


Figura 4.3.1.5 – Diagrama de Transições entre Etapas

O usuário ainda pode interagir direto com o diagrama, criando novos elementos, apagando existentes, ou acessando os dados de cada item clicando neles.

4.3.1.6 Geração Automática de Codificação Heurística

O processo de formalização de conhecimento Heurístico de jogo utilizando a linguagem *DHJOG* resulta em vários elementos codificados, são eles: heurísticas, regiões, valores iniciais de peça, e funções de identificação de contexto. A ferramenta de autoria de Heurísticas deixa transparente para o usuário o processo de geração destas codificações. Um aprendiz iniciante não precisa ter contato com a representação intermediária da linguagem *DHJOG*, sendo que apenas as representações externas de mais alto nível são utilizadas num primeiro momento. A Figura 4.3.1.6-1 apresenta a especificação de uma Heurística de Valor de Peça, na aba principal da tela de edição de Heurística.

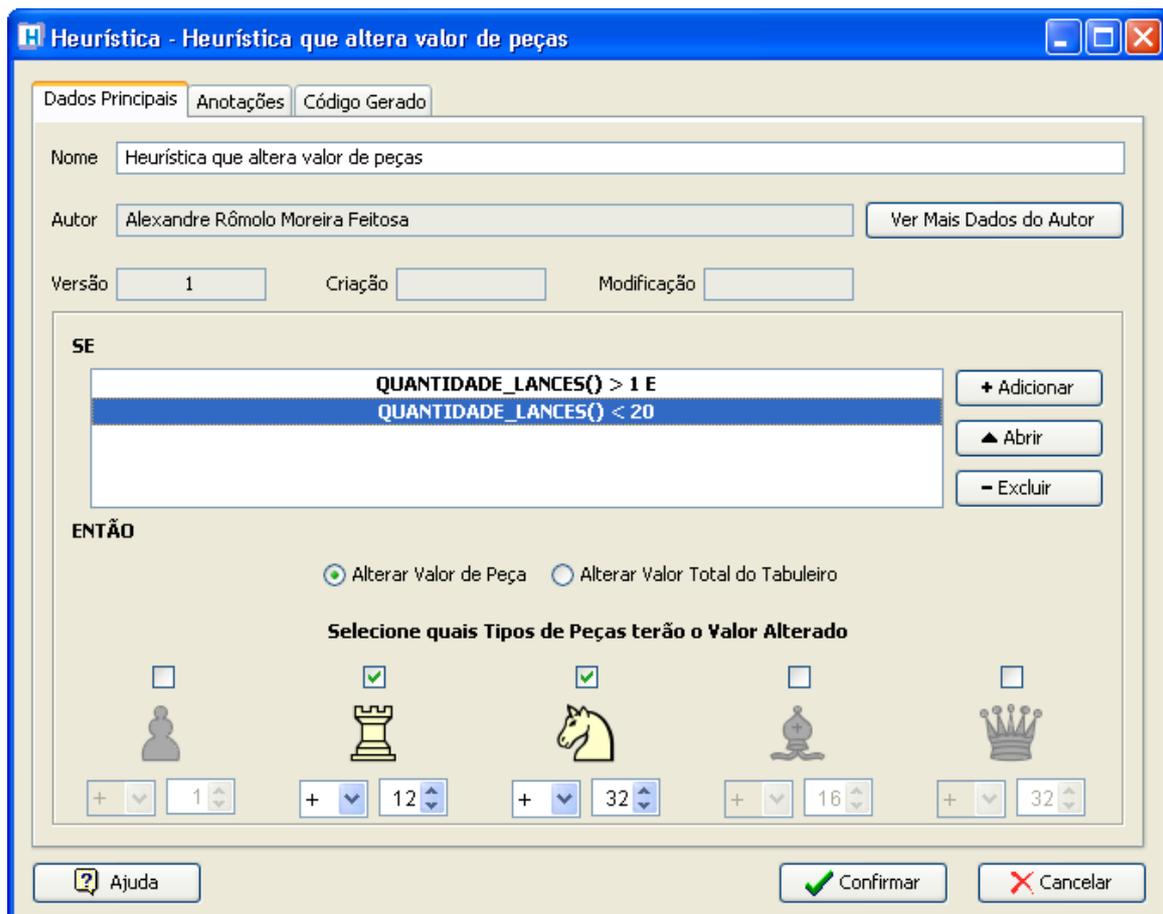


Figura 4.3.1.6-1 – Tela de Edição de Heurística - Aba principal

Conforme o aprendiz vá subindo o nível utilizado da expressividade da linguagem, mais familiar estará com o processo de formalização de heurísticas (Veja a Seção 3.2.5). Ao alcançar o nível quatro (Pleno), será disponibilizado a ele abas para a visualização da codificação gerada pela ferramenta. Isto permitirá ao usuário associar as alterações feitas em um elemento heurístico com as alterações geradas na sua codificação. Entender o processo (sintaxe e semântica) da implementação destes elementos é um pré-requisito fundamental para o nível seis (Especialista). Neste nível o autor poderá alterar e criar funções de identificação de contextos. A Figura 4.3.1.6-2 apresenta na aba de código gerada a implementação da Heurística de Valor de Peça apresentada na Figura 4.3.1.6-1.

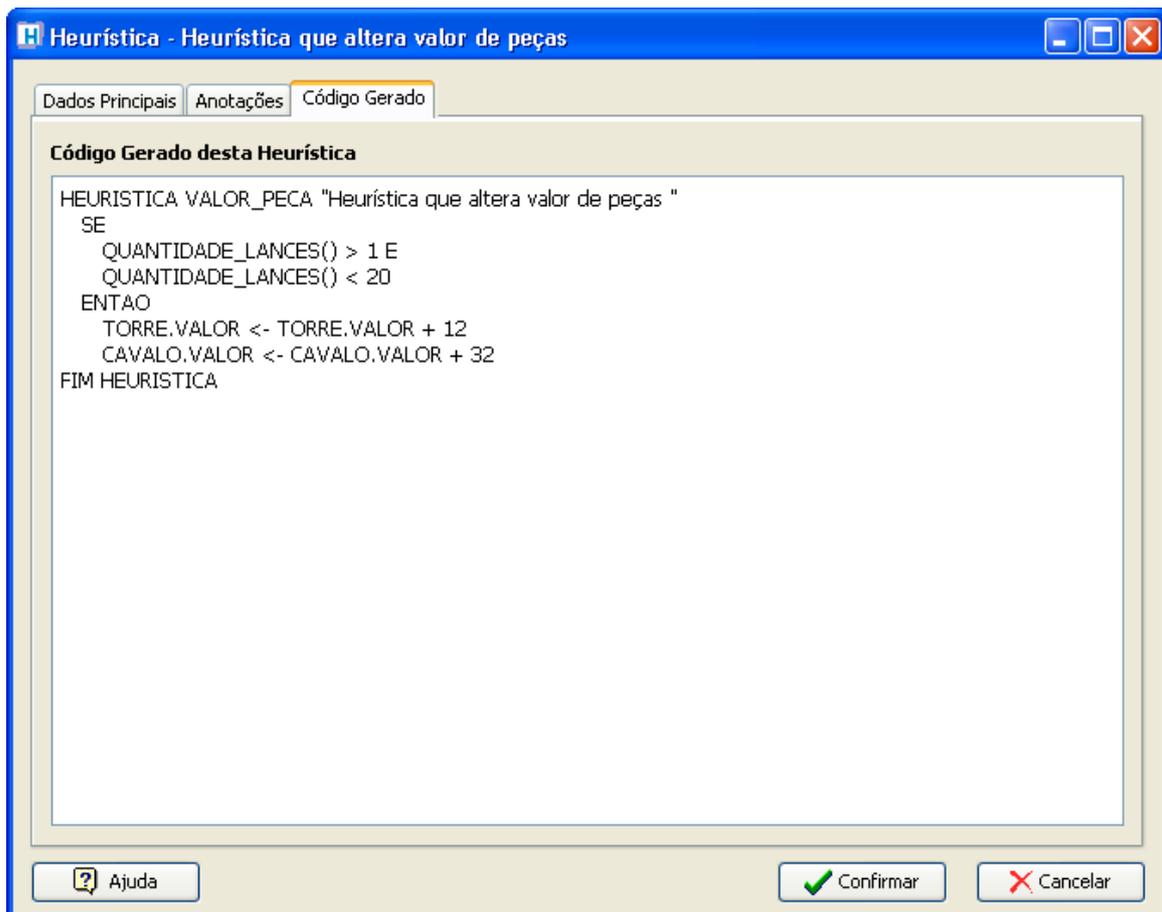


Figura 4.3.1.6-2 – Tela de Edição de Heurística – Aba de visualização de código gerado

4.3.1.7 Uso de Múltiplas Representações Externas

Podemos constatar que a ferramenta de autoria utiliza várias representações externas para apresentar o mesmo conhecimento. Uma região, por exemplo, pode ser vista como uma figura geométrica, ou como um conjunto de elementos formalizado matematicamente. Uma heurística pode ser vista em um nível mais simples apenas pelos seus elementos de atuação (Figura 4.3.1.6-1), ou na forma textual, através de uma linguagem de programação em português (Figura 4.3.1.6-2). Uma Heurística de Transição de Etapa pode ser vista como seus elementos de ação, de forma textual, ou de forma gráfica, através de uma seta no diagrama de transição de etapas (Figura 4.3.1.5). Por tanto, esta implementação é um exemplo de sistema de ensino MER (*Multiple External Representations*), e possui as vantagens inerentes a este tipo de aplicação, apresentadas nos estudos de Labeke (2001) e Aisnworth (1999). Veja Seção 2.1.1.

4.3.1.8 Conceito de Caixa Preta e Caixa de Vidro integradas.

Na Seção 2.1.2 é apresentado um estudo de caso sobre o uso da técnica Caixa Preta (*black box*) e Caixa de Vidro (*glass box*) em ferramentas de ensino, o sistema SimForest (Murray *et al.* 2001). O sistema apresentado implementa as duas técnicas em aplicações distintas. Já no projeto da Ferramenta de Autoria de Heurísticas optou-se por contemplar as duas técnicas na mesma implementação, visto que isto facilitará ainda mais o uso da aplicação pelo usuário.

O aprendiz inicialmente não precisa ter acesso a como certas ações e conceitos são executados e codificados. Ele apenas utiliza estes elementos e os quantifica como vantagem ou desvantagem ao jogador. Isto caracteriza o uso do conceito Caixa Preta. Conforme o usuário vá assimilando o processo de formalização de conhecimento heurístico, ele poderá aumentar o nível de

expressividade a ser utilizado em suas heurísticas. Já no nível quatro (Pleno) ele poderá analisar a codificação dos elementos heurísticos, e entender como os mesmos são implementados e processados (Veja a Seção 4.3.1.6). Por exemplo, ele poderá verificar como um função de identificação de harmonia de peões foi implementada. No nível cinco (Avançado) ele poderá alterar a Expressão de Cálculo Heurístico de cada Etapa. Esta expressão é o principal modelo matemático no processo de análise heurística de uma situação de jogo (Veja a Seção 3.1.7). E por último, no nível seis (Especialista) ele poderá alterar e criar novas funções. As funcionalidades existentes a partir do nível quatro implementam o conceito Caixa de Vidro.

O uso destes conceitos permite que o usuário utilize a ferramenta desde o início do aprendizado, pois a princípio será exigida uma carga menor de conhecimento cognitivo para a execução das tarefas. A opção de utilizar as duas técnicas na mesma implementação vem do fato de que alterar os modelos matemáticos (função e expressão de cálculo heurístico) faz parte do processo de formalização de conhecimento heurístico. E não faz sentido o usuário ter que editar o conhecimento em uma ferramenta no primeiro momento, e depois em outra para refinar sua execução. A proposta desde o início do projeto é de que a ferramenta de edição de conhecimento heurístico seja única.

4.3.1.9 Combinação Automática de Componentes

Com o objetivo de tornar a criação de heurísticas mais prática, e de aumentar a produtividade do usuário nesta tarefa, a tela de edição de Etapas (Figura 4.3.1.3-2) é dotada de mecanismos para tratamento de eventos gerados pelo arrastar e soltar de componentes (*Drag-and-Drop*). Do lado direito desta tela estão as funções de identificação de contextos, agrupadas em categorias, no centro dela na base estão os tipos de peças existentes no jogo, e os seus valores

iniciais na etapa corrente, e no centro ao meio estão desenhadas sobre um tabuleiro de Xadrez as regiões definidas até o momento.

O usuário pode arrastar uma função de identificação de contextos e soltá-la sobre um tipo de peça ou região definida. Automaticamente, será invocado um processo para geração de uma nova heurística já previamente preenchida utilizando o elemento alvo (onde a função foi solta). Preferencialmente tenta-se usar o alvo como parâmetro da função, ou na comparação do seu retorno. Caso isso não seja possível e o alvo seja um tipo de peça, o mesmo será utilizado como elemento a ser incrementado pela ação da heurística. Caso contrário, será criada uma nova heurística sem o uso do elemento atingido.

4.3.1.10 Limitações da Ferramenta de Autoria de Heurísticas

O protótipo da Ferramenta de Autoria de Heurísticas apresentada nas seções anteriores foi projetada e desenvolvida para servir de base para uma implementação final desta aplicação, sendo que o foco principal deste trabalho realizado era a validação da linguagem *DHJOG* no processo de formalização de heurísticas, e a definição das representações externas a serem adotadas. Visto isto, identificamos algumas limitações técnicas e conceituais na ferramenta desenvolvida.

Como limitação técnica podemos citar o fato de que ferramenta não esta integrada ao servidor de Xadrez do CEX (Centro de Excelência de Xadrez), e como ainda não foi implementado no servidor o algoritmo completo do jogador automático, pois ainda falta desenvolver o interpretador de heurísticas, não foi possível realizar um teste funcional das heurísticas formalizadas pela ferramenta. Outra questão técnica é o fato de que o acesso ao banco de dados criado para armazenar as heurísticas está sendo realizado diretamente via TCP/IP, o que deixa o banco suscetível a ataques externos. Para a versão final da aplicação deve ser considerada a adoção de um

modelo de acesso via *Web Service*¹⁰ ao servidor.

Como limitação conceitual podemos citar o fato de que o projeto do protótipo não abordou os outros módulos de interface necessários a aplicação cliente, e a integração do editor com eles. Também não foi discutido os mecanismos necessários para permitir e controlar a cópia de partes de uma heurística de jogo formalizada por outros aprendizes. Estes tópicos estavam fora do escopo proposto para este trabalho, e devem ser abordados em estudos futuros.

4.4 Próximas etapas do desenvolvimento do sistema

O sistema proposto *CACAREJE* é composto por diversos módulos que atuam de forma integrada (Veja a Seção 4.1). O seu desenvolvimento se dá através de diversos estudos e trabalhos científicos, realizados de forma coordenada. No trabalho de mestrado de Daniel Martineschen (2006) foi desenvolvido um protótipo do comparador de heurísticas, que hoje está servindo de base para o desenvolvimento final deste módulo por Antonio Hobmeir e Tiago Sak, na forma de um trabalho de conclusão de curso de graduação. O protótipo da Ferramenta de Autoria de Heurísticas vem se juntar aos estudos realizados, e servirá de base para os próximos trabalhos. No capítulo 5 são apresentadas sugestões de trabalhos futuros.

¹⁰ Web Service é um método padronizado e transparente de acesso remoto a funcionalidades de uma aplicação.

5 CAPÍTULO

CONCLUSÃO E TRABALHOS FUTUROS

Apresentamos neste trabalho de mestrado mecanismos e ferramentas que propiciem a definição formal e incrementável de conceitos heurísticos de Xadrez. Este estudo está integrado ao projeto PROTEX, de apoio computacional ao ensino de Xadrez nas escolas brasileiras (Direne *et al.* 2004), e foi realizado em conjunto com o trabalho de mestrado de Daniel Martineschen (2006). Dentro destes trabalhos foi delineada a arquitetura de um sistema de ensino de conceitos heurísticos de jogos, que estimule a alternância da colaboração e da competição entre os aprendizes, o *CACAREJE* (Veja Capítulo 4). Após os estudos realizados durante o levantamento bibliográfico sobre este assunto e os temas correlatos, ficou constatado que não existem trabalhos sobre o uso desta abordagem de ensino, e nem sobre o uso da formalização de conhecimento heurísticos de jogos.

A conclusão desta dissertação gerou três grandes resultados que vão de encontro ao desenvolvimento do ambiente de ensino proposto. Primeiramente foi feita uma análise dos conceitos relacionados a heurísticas de jogos, e após a compreensão destes fundamentos foi definida a linguagem para Definição de Heurísticas de Jogos – *DHJOG*. Através dela é possível formalizar o conhecimento heurístico, e posteriormente utilizá-lo em jogadores automáticos para a realização de competições entre heurísticas.

Segundo, foi especificado junto com o trabalho do Daniel Martineschen a arquitetura completa do ambiente de ensino *CACAREJE*. Sendo que o trabalho do Daniel focou na questão entre a alternância da colaboração e da competição, e o presente trabalho na formalização do conhecimento heurístico, e nos mecanismos de interface da aplicação cliente.

E por último, mas não menos importante, foi implementado um protótipo da ferramenta de

autoria de heurísticas. Para o projeto desta ferramenta foi feito um estudo sobre técnicas de representação externa do conhecimento. E após esta análise, alguns destes conceitos foram adaptados ao contexto de uso da aplicação, e utilizados no seu desenvolvimento. A implementação deste protótipo possibilitou uma validação inicial do uso da linguagem *DHJOG*, e servirá de base para a implementação final da ferramenta.

É importante citar também os resultados que serão alcançados com a implantação do ambiente de ensino *CACAREJE*, tais como: o aumento do ferramental disponível para o ensino de Xadrez nas escolas; o estímulo à competição, ao estudo das partidas, e a troca de conhecimento entre aprendizes; e a possibilidade de que o ensino heurístico de Xadrez possa ser utilizado como atividade prática por outras disciplinas escolares, como por exemplo, no ensino de lógica e de matemática.

5.1 Trabalhos Futuros

Podemos sugerir a partir deste trabalho três linhas de estudos futuros:

- 1) **Projetar e desenvolver os módulos restantes do ambiente *CACAJERE***: Estas itens já foram devidamente conceituados neste trabalho e na dissertação do Daniel Martineschen (2006), são eles: O Comparador Genérico de Definições Heurísticas, Escalonador Genérico de Partidas Automáticas, Ferramenta para Definição de Situações de Jogo, Repositório de Objetos Heurísticos, Ferramentas de Comunicação entre Usuários, Módulo para Criação/Participação em Competições Heurísticas, Módulo para Controle da Execução de Partidas, Módulo para Definição de Permissões de Acesso, e a versão final da Ferramenta para Autoria de Heurísticas de Jogos. Todas estas ferramentas deverão estar integradas ao mesmo ambiente de ensino, e ao servidor de Xadrez do CEX

(Centro de Excelência de Xadrez).

- 2) **Aumentar o poder de formalização da linguagem *DHJOG***: Isto pode ser realizado criando-se novas funções para a identificação de estratégias de Xadrez que abordem temas mais avançados, como técnicas de abertura, harmonia de peões, e etc. Esta atividade deverá ser efetuada por Enxadristas Mestres, sob a supervisão de integrantes do projeto PROTEX. Outra linha de ação é incluir novos elementos de representação na linguagem que permitam formalizar estratégias avançadas, por exemplo, que compreendam várias jogadas. Para isto, possivelmente será necessário lidar com um histórico de situações de jogo, e forçar jogadas seguidas.

- 3) **Diminuir o conhecimento cognitivo necessário para a realização das Tarefas**: Sempre é possível agregar técnicas e estratégias para aumentar a facilidade do uso de um sistema pelo usuário, e em um ambiente de ensino isto é uma das prioridades. Podemos sugerir de imediato o estudo para a inserção na aplicação cliente de um agente inteligente, que possa monitorar o aprendiz durante o uso dos recursos do sistema, e que o auxilie no processo de compreensão e utilização dos conceitos heurísticos de Xadrez. Outro ponto a também ser considerado é a necessidade de inclusão de novas representações externas na ferramenta de edição. Por exemplo, um gráfico dinâmico da ativação de cada heurística na avaliação de uma situação de jogo, funcionaria de forma semelhante a uma ferramenta de depuração¹¹ de um ambiente de desenvolvimento, e poderá auxiliar ainda mais o aprendiz a entender o processo de formalização e interpretação de heurísticas.

¹¹ Uma ferramentas de depuração (*debug*) normalmente permite ao programador acompanhar valores de variáveis em memória, executar linha a linha trechos de código, e acompanhar a pilha de chamada de funções.

APÊNDICE A

ESPECIFICAÇÃO DA LINGUAGEM *DHJOG*

Neste seção iremos apresentar a especificação da linguagem de definição de Heurísticas de jogos *DHJOG*. Esta linguagem possibilita a formalização completa de uma heurística de jogo, que posteriormente poderá ser utilizada por um jogador automático para simular partidas.

A representação do conhecimento heurístico de jogos pode ser dividida em duas etapas: (i) Especificar como uma situação atual do jogo pode ser representada, quais elementos existem e em quais situações eles poderão estar dispostos em um determinado momento; (ii) Como representar o conhecimento heurístico de como avaliar uma situação de jogo, quais heurísticas serão utilizadas em determinado momento de jogo, quando se desloca-se de um momento de jogo para outro (por exemplo do início para o meio de uma partida), e como as características da atual situação de jogo interferem no cálculo final do seu valor heurístico. A Figura A apresenta o diagrama com os pacotes que formam a linguagem *DHJOG*.

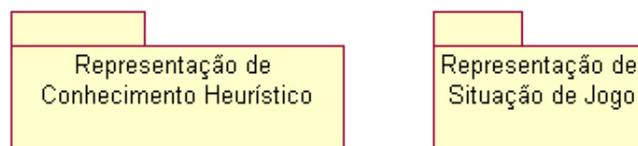


Figura A – Diagrama de Pacotes da Linguagem *DHJOG*

A1. Elementos para Representação do Conhecimento Heurístico

A Figura A.1 apresenta o diagrama de classes dos elementos definidas na *DHJOG* para a representação do conhecimento heurístico de um jogo.

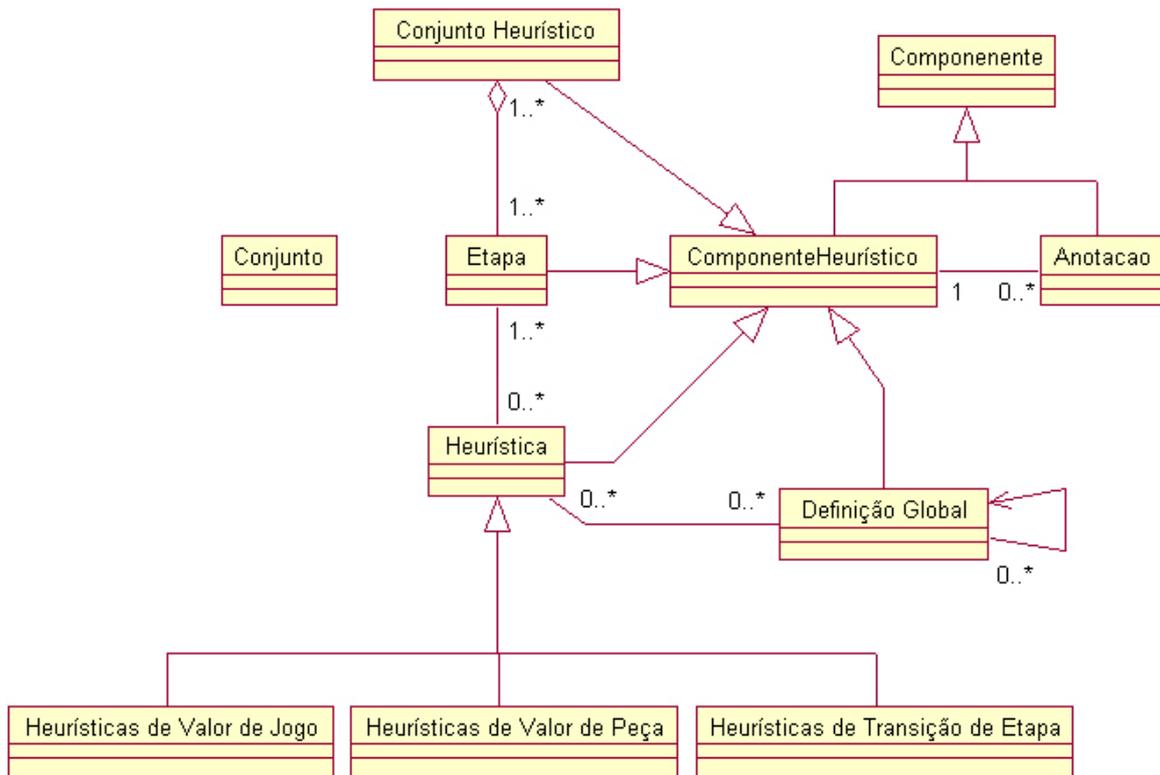


Figura A.1 – Classes para Representação do Conhecimento Heurístico

Veremos nas próximas seções a especificação completa de cada um destes elementos definidos. A Tabela A.1 apresenta a legenda das cores utilizadas na identificação dos elementos da linguagem. O uso deste padrão visa facilitar o entendimento da especificação.

| Legenda de Cores | |
|------------------|---|
| Cor | Descrição |
| | Tipo de Atributo ou de variável local |
| | Nome de Atributo |
| | Nome de método |
| | Constante |
| | Palavra Reservada, Operador ou Comando |
| | Nome de identificador, expressão ou bloco criado pelo usuário |

Tabela A.1 – Legenda de Cores Utilizadas

A1.1 Componente

Um componente é um elemento que possui os seguintes atributos:

- **TEXTO NOME**: Nome que identifica este componente:
- **TEXTO AUTOR**: Informação que identifica qual usuário foi Autor deste componente.
- **INTEIRO VERSAO**: Número que indica a versão deste componente. A primeira versão tem valor 1.

A1.2 Componente Heurístico

Um componente heurístico é um elemento que além de possuir as características de um Componente possui ainda a possibilidade de possuir relacionado a ele zero ou mais Anotações. Ele define os seguintes atributos:

- **ANOTACAO[] ANOTACOES**: Anotações criadas pelo autor e por outros usuários relacionadas a este componente.

A.1.3 Conjunto

Este elemento é utilizado para criar um conjunto de elementos de um tipo específico, como por exemplo um conjunto de etapas, heurísticas, ou anotações. Os elementos são organizados internamente através de uma lista dinâmica. Não existe limite para inclusão de elementos. As operações de conjuntos matemáticos são suportadas com objetos deste tipo, e ele não aceita elementos repetidos. Um elemento do tipo Conjunto possui a seguinte estrutura de métodos:

- **total()**: Indica quantos elementos existem dentro deste conjunto. Este atributo é somente de leitura.
- **adicionar(COMPONENTE)**: Adiciona um novo elemento ao conjunto.
- **remover(COMPONENTE)**: Remove o elemento passado do conjunto.
- **removerTodos()**: Remove todos os elementos do conjunto.
- **elemento(INTEIRO)**: Retorna o elemento que está na posição especificada. Caso a posição seja inválida apenas retorna vazio. O elemento retornado não é retirado do Conjunto.

A.1.4 Conjunto Heurístico

Um Conjunto Heurístico representa a codificação feita por um Autor sobre o conhecimento heurístico de um determinado jogo. Este elemento é utilizado por um jogador automático para avaliar uma determinada situação de jogo. Ele define os seguintes atributos:

- **ETAPA ETAPA_INICIAL**: Etapa que será executada na primeira vez em que este Conjunto Heurístico for invocado para avaliar uma situação de jogo.
- **ETAPA ETAPA_ATUAL**: Etapa que está atualmente ativada como etapa atual.
- **ETAPA[] ETAPAS**: Conjunto com todas as etapas pertencentes a este Conjunto Heurístico.
- **SITUACAO_JOGO SITUACAO_ATUAL**: Situação atual de jogo que será avaliada e qualificada pelo conhecimento heurístico aqui representado.

A.1.5 Etapa

Uma etapa representa um determinado momento do jogo onde o autor quer utilizar apenas um conjunto determinado de heurísticas. Através do uso de etapas pode-se adotar estratégias diferentes de acordo com o momento atual do jogo. Uma etapa é composta por:

- **Valores Iniciais das Peças:** Responsável pela inicialização dos valores de cada tipo de peça no início da execução da etapa. Isto garante que cada Peça tenha um valor válido a ser utilizada durante o cálculo do valor heurístico da situação de jogo atual. Os valores das peças poderão ser alterados através das Heurísticas de Valor de Peça.
- **Expressão de Cálculo Heurístico:** Expressão que é responsável por calcular o valor heurístico inicial da situação atual de jogo. Esta expressão deve retornar um valor real e pode utilizar os valores de cada peça neste cálculo. O valor da situação atual do jogo pode posteriormente ser alterada pelas Heurísticas de Valor de Jogo.

Um elemento Etapa possui os seguintes atributos específicos:

- **TEXTO VALOR_INICIAL_PECAS:** Código que define quais são os valores iniciais dos tipos de Peças do Jogo.
- **TEXTO EXPRESSAO_CALCULO_HEURISTICO:** Código que indica como será realizado o cálculo do valor heurístico da situação atual de jogo.
- **HEURISTICA_TRANSICAO_ETAPA[] HTES:** Conjunto com as Heurísticas de Transição de Etapas relacionadas a esta Etapa.
- **HEURISTICA_VALOR_PECA[] HVPS:** Conjunto com as Heurísticas de Valor de Peça relacionadas a esta Etapa.
- **HEURISTICA_VALOR_JOGO[] HVJS:** Conjunto com as Heurísticas de Valor de

Jogo relacionadas a esta Etapa.

A.1.6 Heurísticas

Uma heurística é um componente na forma de uma regra de produção onde caso uma determinada expressão condicional seja satisfeita um determinado conjunto de ações será executado. Vejamos a seguir os tipos existentes de Heurísticas:

- **Heurística de Transição de Etapa – HTE:** É uma heurística que determina se o conjunto heurístico deve adotar uma outra etapa como etapa atual, de acordo com uma condição de transição avaliada.
- **Heurística de Valor de Peça – HVP:** É uma heurística que determina se o valor de determinados tipos de peças serão alterados. Cada tipo de peça pode receber um novo valor que poderá ser utilizado pela expressão de cálculo heurística para qualificar a situação de jogo atual. Desta forma determinados tipos de peças podem ganhar maior ou menor importância.
- **Heurística de Valor de Jogo – HVJ:** É uma heurística que caso uma condição seja satisfeita o valor de situação de jogo já calculado pela expressão de cálculo heurístico será alterado para um novo valor.

Um elemento Heurística possui os seguintes atributos específicos:

- **TEXTO CONDICAO:** Código que indica a expressão condicional a ser avaliada para indicar se esta heurística será ou não ativada.
- **TEXTO ACAO:** Código que indica a ação que será realizada caso a expressão condicional tenha o valor lógico VERDADEIRO.

- **DEFINICAO_GLOBAL[] DEFINICOES_GLOBAIS**: Definições globais que foram utilizadas na construção do código da Condição e da Ação desta heurística.

A.1.7 Definição Global

São definições de elementos que poderão ser utilizados internamente pelas etapas e pelas heurísticas na construção (sintaxe) das expressões condicionais ou das ações a serem executadas. Estes elementos podem ser funções ou variáveis. Um elemento Definição Global possui os seguintes atributos específicos:

- **TEXTO DEFINICAO**: Código que define como será construída esta definição global.
- **DEFINICAO_GLOBAL[] DEFINICOES_GLOBAIS**: Definições globais que foram utilizadas na construção do código de Definição.

A.1.7.1 Função

Uma função é uma funcionalidade que quando chamada pode receber alguns parâmetros de entrada, ela realiza um processamento interno, e devolve uma resposta específica. O uso de funções permite isolar determinados conceitos em módulos, facilitando o reaproveitamento deles na construção de outros elementos, e pode tornar transparente para usuários iniciantes como determinadas ações são realizadas, o seu uso diminui a carga de esforço cognitivo necessário para iniciar a formalização de heurísticas.

A.1.7.2 Variável Global

É uma variável que pode ser utilizada pelas etapas e heurísticas de um conjunto heurístico na especificação das condições de testes, e das ações a serem realizadas. Por exemplo, a definição de uma região de um tabuleiro de Xadrez representa um conjunto de casas, esta definição pode ser utilizada por várias funções na hora de avaliar determinadas situações do tabuleiro.

A.1.8 Anotação

Representa um texto incluído por um usuário (autor ou não) em determinado elemento do conjunto heurístico a fim de explicar, questionar, elogiar ou criticar o conhecimento utilizado. É utilizado para auxiliar a compreensão do conhecimento heurístico representado, e para o intercâmbio de informações entre diversos autores. Ela possui os seguintes atributos específicos:

- **TEXTO_TITULO**: Título definido para esta Anotação
- **TEXTO_DESCRICAÇÃO**: Texto redigido para esta anotação.
- **COMPONENTE_HEURISTICO COMPONENTE**: Identifica para qual Componente Heurístico esta anotação está relacionada.
- **TIPO_ANOTACAO TIPO**: Identifica qual o tipo desta anotação, pode ser um dos seguintes valores {EXPLICACAO, QUESTAO, POSITIVA, NEGATIVA, NORMAL}.

A.2 Elementos para Representação de Situação de Jogo

Na Figura A.2 é apresentado o diagrama de classes com os elementos para a representação da situação de uma situação de jogo.

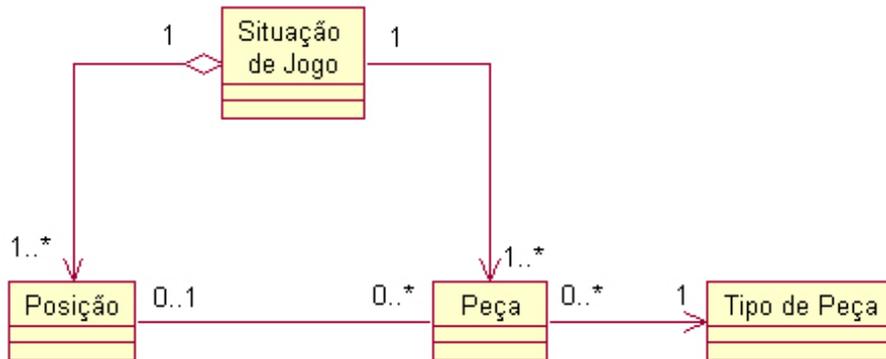


Figura A.2 – Representação de uma Situação de Jogo

A.2.1 Posição

Elemento que representa um lugar de um jogo onde uma ou mais peças podem ser colocadas. No Xadrez seriam as casas do tabuleiro, já em um jogo de baralho como o Truco poderiam ser: o monte, a vira, a primeira carta, segunda carta e terceira carta, e as cartas jogadas na mesa. Este elemento possui os seguintes atributos:

- **TEXTO NOME**: Nome que identifica a posição no contexto do jogo.
- **DONO_ELEMENTO DONO**: Especifica quem é o proprietário desta posição.
- **POSICAO POSICAO_DIREITA**: Posição a direita desta posição.
- **POSICAO POSICAO_ESQUERDA**: Posição a esquerda desta posição.
- **POSICAO POSICAO_FRENTE**: Posição a frente desta posição.

- **POSICAO POSICAO_TRAS**: Posição localizada a trás desta posição.
- **PECA PECA_ATUAL**: Peça atualmente nesta posição.

A.2.2 Tipo de Peça

Podemos classificar as peças de um determinado jogo em um ou mais tipos. Esta classificação é utilizada para valorar as peças de forma distinta de acordo com suas funcionalidades dentro do jogo. Um elemento Tipo Peça possui os seguintes atributos:

- **TEXTO NOME**: Identifica o tipo de Peça em questão.
- **INTEIRO VALOR**: Valor heurístico que cada uma das peças que forem deste tipo possuirão dentro de uma situação de jogo, na qual estas peças apareçam.

A.2.3 Peça

Representa um elemento específico de um jogo que é de um determinado Tipo de Peça, e caso esteja presente no jogo estará associada com uma determinada posição. Como um Peão branco do xadrez, ou mesmo a carta três de paus de um baralho. Uma Peça possui os seguintes atributos:

- **TIPO_PECA TIPO**: Tipo da peça.
- **DONO_ELEMENTO DONO**: Especifica quem é o proprietário desta peça.
- **POSICAO POSICAO_ATUAL**: Especifica a posição na qual esta peça está posicionada.

A.2.4 Situação de Jogo

É o elemento que representa todas as informações necessárias para identificar um determinado momento de uma partida de um jogo. Como a disposição de peças nas posições existentes, quantas rodadas já foram executadas, pontuação atual, e outras características específicas do jogo tratado. A principal característica da situação de jogo é o valor heurístico atribuído a ela pelo Conjunto Heurístico. É a partir deste valor que o jogador automático pode escolher a próxima jogada, comparando o valor heurístico de cada nova situação de jogo resultante das jogadas possíveis no momento. A situação que tiver o maior valor heurístico representa uma situação de jogo mais favorável ao conjunto heurístico definido pelo Autor. O elemento Situação de Jogo possui os seguintes atributos:

- **POSICAO[] POSICOES**: Todas as posições existentes na situação de jogo, armazenadas em um único conjunto.
- **PECA[] PECAS**: Todas as peças que atualmente estão dispostas nesta situação de jogo, armazenadas em um único conjunto.
- **INTEIRO QUANTIDADE_LANCES**: Representa quantos lances foram realizados até o momento no jogo.
- **REAL VALOR**: Valor heurístico atribuído a esta situação de jogo e que auxiliará o Jogador automático a escolher a próxima jogada.

A.3 Linguagem de Codificação de Heurísticas - LCH

A linguagem de codificação de heurísticas - LCH é uma linguagem definida internamente na *DHJOG*. Ela faz parte do ferramental disponível para a formalização de conhecimentos heurísticos de jogos, e o seus objetivos são:

- Permitir identificar contextos, características, e comportamentos dentro de uma situação de jogo.
- Permitir alterar o valor dos elementos envolvidos no cálculo heurístico, desde os tipos de peças até a próprio valor heurístico previamente calculado para a situação de jogo.
- Definir valores globais que possam ser reutilizados em outras especificações feitas neste mesma linguagem.

A.3.1 Características da Linguagem

A linguagem definida possui as seguintes características:

- Orientada a Objetos.
- Interpretada.
- Não é sensível ao contexto.
- Próxima da Linguagem natural. Utiliza estruturas definidas em PORTUGOL e operadores matemáticos semelhantes aos utilizados no ensino médio.

A.3.2 Elementos que podem ser Codificados

Existem as seguintes estruturas que podem ser codificadas com esta linguagem:

- Ação e condição de heurísticas,
- Inicialização dos valores dos tipos de peças,
- Expressão de cálculo heurístico,
- Escrita de definições globais (Funções e Regiões),

A.3.3 Tipos de dados

A linguagem possui uma estrutura de tipos de dados híbrida, com tipos de dados primitivos e tipos de dados construídos (Classes).

A.3.3.1 Tipos Primitivos

Os tipos de dados primitivos definidos são:

- **INTEIRO**: Número sem casas decimais.
- **REAL**: Número com possibilidade de utilizar casas decimais.
- **LOGICO**: Representa o valor verdade ou falso.
- **TEXTO**: Seqüência de caracteres que representam um texto.

A.3.3.2 Tipos Construídos

As classes da arquitetura de formalização de heurísticas que podem ser referenciadas dentro da codificação são: CONJUNTO_HEURISTICO, ETAPA, DEFINICAO_GLOBAL, POSICAO, TIPO_PECA, PECA e SITUACAO_JOGO.

A.3.4 Constantes

As seguintes constantes foram definidas na linguagem: **VAZIO**, **VERDADEIRO**, **FALSO**, **EU**, **OPONENTE**, **NINGUEM**, **EXPLICACAO**, **QUESTAO**, **POSITIVA**, **NEGATIVA** e **NORMAL**.

A.3.5 Domínios

Um domínio é um grupo formado por constantes válidas para um determinado tipo de atributo. Na *LCH* foram definidos os seguintes domínios:

- **VALOR_LOGICO** = {**VERDADEIRO**, **FALSO**}.
- **DONO_ELEMENTO** = {**EU**, **OPONENTE**, **NINGUEM**}.
- **TIPO_ANOTACAO** = {**EXPLICACAO**, **QUESTAO**, **POSITIVA**, **NEGATIVA**, **NORMAL**}.

A.3.6 Operadores

Os seguintes operadores são suportados pela linguagem:

| Operadores Matemáticos | |
|-------------------------------|------------------|
| + | Adição |
| - | Subtração |
| - | Subtração |
| × ou * | Multiplicação |
| ÷ ou / ou DIVIDE | Divisão |
| RESTO_DIVISAO | Resto da divisão |

Tabela A.3.6-1 – Operadores Matemáticos

| Operadores Lógicos | |
|---------------------------|----------------|
| E | Operação E |
| OU | Operação OU |
| NAO | Negação Lógica |

Tabela A.3.6-2 – Operadores Lógicos

| Operadores Relacionais | |
|-------------------------------|------------------|
| > | Maior que |
| < | Menor que |
| >= ou MAIOR_IGUAL | Maior ou igual a |
| <= ou MENOR_IGUAL | Menor ou igual a |
| = ou IGUAL | Igual |
| ≠ ou <> ou DIFERENTE | Diferente |

Tabela A.3.6-3 – Operadores Relacionais

| Operadores de Atribuição | |
|----------------------------------|---------------------------------|
| <- ou <=< ou := ou RECEBE | Atribui um valor a uma variável |

Tabela A.3.6-4 – Operadores de Atribuição

| Operadores de Precedência | |
|----------------------------------|------------|
| () | Parênteses |

Tabela A.3.6-5 – Operadores de Precedência

| Operadores de Manipulação de Conjuntos | |
|---|---|
| $[]$ | Declaração de conjunto e acesso a um elemento em uma determinada posição |
| $\{ \}$ | Declaração dos elementos de um conjunto, cada elemento separado por vírgula |
| + | Soma dois conjuntos |
| - | Subtração entre conjuntos |
| \cup ou UNIAO | União de dois conjuntos |
| \cap ou INTERSECCAO | Intersecção entre dois conjuntos |
| \in ou PERTENCE | Elemento pertence ao conjunto |
| \notin ou NAO_PERTENCE | Elemento não pertence ao conjunto |
| \supset ou CONTEM | Conjunto contém o outro conjunto |
| $\not\supset$ ou NAO_CONTEM | Conjunto não contém o outro conjunto |
| \subset ou CONTIDO | Conjunto está contido no outro conjunto |
| $\not\subset$ ou NAO_CONTIDO | Conjunto não está contido no outro conjunto |

Tabela A.3.6-6 – Operadores de Manipulação de Conjuntos

| Operadores de Manipulação de Texto | |
|---|---|
| “ ” (Aspas duplas) | Identificador de Constate Literal TEXTO |

Tabela A.3.6-7 – Operadores de Manipulação de Textos

| Operadores para Definição de Comentários | |
|---|---------------------------------|
| # | Comentário para uma única linha |

Tabela A.3.6-8 – Operadores para Definição de Comentários

A.3.7 Estruturas de Controle

Na linguagem foram definidas as seguintes estruturas de controle:

- Estrutura de Seqüência (Processamento seqüencial de comandos).
- Seleção Simples. **SE..ENTAO** (*IF*).
- Seleção Composta. **SE..ENTAO..SENAO** (*IF – ELSE*).
- Repetição com Variável de Controle. **PARA..DE..ATE..FACA** (*FOR*)

A.3.8 Sintaxe

Nesta seção são apresentadas as regras de sintaxe para a codificação dos elementos através da *LCH*.

A.3.8.1 Delimitador de final de Comando

Preferencialmente será escrito apenas um comando por linha, neste caso, o próprio caracter de final de linha será o separador de comando. Caso seja necessário colocar mais de um comando na mesma linha eles devem ser separados pelo caracter “;”.

A.3.8.2 Palavras Reservadas

As palavras reservadas na *LCH* são: **INTEIRO, REAL, LOGICO, TEXTO, POSICAO, VERDADEIRO, FALSO, VAZIO, TIPO_PECA, PECA, SITUACAO_JOGO, CONJUNTO, COMPONENTE, COMPONENTE_HEURISTICO, ETAPA, CONJUNTO_HEURISTICO, DEFINICAO_GLOBAL, ANOTACAO, HEURISTICA, TRANSICAO_ETAPA, VALOR_PECA, VALOR_JOGO, EU, Oponente, NINGUEM, EXPLICACAO, QUESTAO, POSITIVA, NEGATIVA, NORMAL, SE, ENTAO, FIM, PARA, DE, ATE, FACA, RETORNA, FUNCAO, AUTOR e VERSAO.**

A.3.8.3 Definição de Identificador

Um identificador é um nome definido pelo autor para referenciar um elemento do Conjunto Heurístico, como por exemplo uma função, heurística, região, etapa ou anotação. O nome de um identificador não pode ser uma palavra reservada, não pode iniciar com número, e caso possua espaços em branco deve ser definida entre aspas duplas.

A.3.8.4 Declaração de Variáveis

| |
|---|
| TIPO NOME ou TIPO [] NOME |
|---|

A segunda forma de declaração com [] define uma variável do tipo conjunto.

A.3.8.5 Seleção Simples

```
SE (EXPRESSAO_CONDICIONAL) ENTAO
  <COMANDOS>
FIM SE
```

A.3.8.6 Seleção Composta

```
SE (EXPRESSAO_CONDICIONAL) ENTAO
  <COMANDOS>
SENAO
  <COMANDOS>
FIM SE
```

A.3.8.7 Repetição com Variável de Controle

```
PARA VARIAVEL_INTEIRA DE VAL_INICIAL ATE VAL_FINAL FACA
  <COMANDOS>
FIM PARA
```

A.3.8.8 Heurística Transição de Etapa

```
HEURISTICA TRANSICAO_ETAPA NOME
SE
  (EXPRESSAO_CONDICIONAL)
ENTAO
  ETAPA_ATUAL <- NOME_ETAPA
FIM HEURISTICA
```

A.3.8.9 Heurística Valor de Peça

```
HEURISTICA VALOR_PECA NOME
SE
  (EXPRESSAO_CONDICIONAL)
ENTAO
  TIPO_PECA.VALOR <- TIPO_PECA.VALOR <OPERADOR> <VALOR>
  .....
FIM HEURISTICA
```

A.3.8.10 Heurística Valor de Jogo

```

HEURISTICA VALOR_JOGO NOME
SE
    (EXPRESSAO_CONDICIONAL)
ENTAO
    JOGO.VALOR <- JOGO.VALOR <OPERADOR> <VALOR>.
FIM HEURISTICA

```

A.3.8.11 Região

```

POSICAO[] NOME <- {POSICAO1, POSICAO2, POSICAO3, ...}

```

A.3.8.12 Função

```

FUNCAO NOME
DESCRICAO <- "TEXTO"
RETORNA
    TIPO NOME DESCRICAO <- "TEXTO"
PARAMETROS
    TIPO NOME DESCRICAO <- "TEXTO"
    .....
COMANDOS
    <COMANDOS>
FIM FUNCAO

```

Uma função retorna um valor através do comando **RETORNA**. Este comando deve ser chamado junto com o valor a ser retornado.

A.3.9 Definição da Linguagem *DHJOG* em BNF

Na Figura A.3.9-1 e A.3.9-2 é apresentada a definição completa da gramática da Linguagem *DHJOG* utilizando a Metalinguagem BNF (*Backus-Naur Form*).

```

<conjunto_heurístico> ::= “CONJUNTO_HEURISTICO” <id>
                        “ETAPA_INICIAL” <id>
                        { <regiões> }
                        <etapa>+
                        { <anotação> }
                        “FIM CONJUNTO_HEURISTICO”

<id> ::= <letra> { <letra> | <digito> | <símbolo> }
<letra> ::= “A” | “B” | “C” | “D” | “E” | “F” | “G” | “H” | “I” | “J” | “K” | “L” |
           “M” | “N” | “O” | “P” | “Q” | “R” | “S” | “T” | “U” | “V” | “X” | “Z”
<símbolo> ::= “_”
<digito> ::= “0” | “1” | “2” | “3” | “4” | “5” | “6” | “7” | “8” | “9”
<texto> ::= “” { <qualquer_caractere> } “”

<anotação> ::= “ANOTACAO” <id>
             (“EXPLICACAO” | “QUESTAO” | “POSITIVA” | “NEGATIVA” | “NORMAL”)
             <texto>
             “FIM ANOTACAO”

<região> ::= “POSICAO” <id> “<- { “ <id> { “, “ <id> } “}”

<etapa> ::= “ETAPA” <id>
           <inicializa_peça>+
           <expressão_cálculo_heurístico>
           { <heurística_transição_etapa> }
           { <heurística_valor_peça> }
           { <heurística_valor_jogo> }
           { <anotação> }
           “FIM ETAPA”

<inicializa_peça> ::= <id> “<-“ <valor_numérico>
<valor_numérico> ::= [“-“] <digito>+
<expressão_cálculo_heurístico> ::= “SITUACAO_JOGO.VALOR <-“ <expressão_numérica>

<heurística_transição_etapa> ::= “HEURISTICA_TRANSICAO_ETAPA” <id>
                                “SE (“<expressão_condicional>”) ENTÃO”
                                “ETAPA_ATUAL <-“ <id>
                                “FIM HEURISTICA”

<heurística_valor_peça> ::= “HEURISTICA_VALOR_PECA” <id>
                           “SE (“<expressão_condicional>”) ENTÃO”
                           <altera_valor_peça>+
                           “FIM HEURISTICA”

```

Figura A.3.9-1 – Formalização da *DHJOG* em BNF (Parte 1)

```

<altera_valor_peça> ::= <id> “<-“ <id> <op_mat> <valor_numérico>

<heurística_valor_jogo> ::= “HEURISTICA VALOR JOGO” <id>
    “SE (“<expressão_condicional>”) ENTAO”
    “SITUACAO JOGO.VALOR <- SITUACAO JOGO.VALOR”
    <op_mat><valor_numérico>
    “FIM HEURISTICA”

<op_mat> ::= “+” | “-” | “*” | “/” | “RESTO_DIVISAO” | “UNIAO” | “INTERSECCAO”
<expressão_numérica> ::= <operando_numérico> { <op_mat> <operando_numérico> } |
    (“ <operando_numérico> { <op_mat> <operando_numérico> } “)
<operando_numérico> ::= <id> | <valor_numérico> | <expressão_numérica>
<op_lógico> ::= “E” | “OU” | “NAO”
<op_relacional> ::= “>” | “<” | “=>” | “<=” | “<” | “=” |
    “PERTENCE” | “NAO PERTENCE” |
    “CONTEM” | “NÃO CONTEM” |
    “CONTIDO” | “NAO CONTIDO”
<valor_lógico> ::= “VERDADEIRO” | “FALSO”
<expressão_condicional> ::= <condição> { <op_lógico> <expressão_condicional> }
<condição> ::= <operando_condicional> [ <op_relacional> <operando_condicional> ]
<operando_condicional> ::= <id> | <valor_lógico>

<função> ::= “FUNCAO” <id>
    “DESCRICAO” <texto>
    “RETORNA”
    <tipo> <id> “DESCRICAO” <texto>
    [ <parametros> ]
    “COMANDOS”
    { <definição_variável> }
    { <comando> }+
    “FIM FUNCAO”

<tipo> ::= ( <tipo_primitivo> | <tipo_construído> ) [ “[ ]” ]
<tipo_primitivo> ::= “INTEIRO” | “REAL” | “LOGICO” | “TEXTO”
<tipo_construído> ::= “ETAPA” | “POSICAO” | “TIPO_PECA” | “PECA”
<parâmetros> ::= “PARAMETROS” <definição_parâmetro>+
<definição_parâmetro> ::= <tipo> <id> “DESCRICAO” <texto>

<definição_variável> ::= <tipo> <id>

<comando> ::= <atribuição> | <retorno_função> | <estrutura_de_controle>
<atribuição> ::= <id> “<-“ <expressão_numérica> | <expressão_condicional>
<retorno_função> ::= RETORNA <valor_numérico> | <valor_lógico> | <id>

<estrutura_controle> ::= <estrutura_decisão> | <estrutura_repetição>
<estrutura_decisão> ::= “SE (“<expressão_condicional>”) ENTAO”
    { <comando> }+
    [ “ENTAO”
    { <comando> }+ ]
    “FIM SE”
<estrutura_repetição> ::= “PARA” <id> “DE” <valor_numérico> “ATE” <valor_numérico> “FACA”
    { <comando> }
    “FIM PARA”

```

Figura A.3.9-2 – Formalização da *DHJOG* em BNF (Parte 2)

APÊNDICE B

EXTENSÃO DA LINGUAGEM *DHJOG* PARA O JOGO DE XADREZ

Neste seção iremos apresentar a extensão realizada na linguagem *DHJOG* para o jogo de Xadrez. Esta extensão visou acrescentar elementos e atributos para uma representação completa do conhecimento heurístico do jogo de Xadrez.

B.1 Elementos para Representação de Situação de Jogo de Xadrez

A Figura B.1 apresenta o diagrama de classes dos elementos para representação de Situação de Jogo do Xadrez.

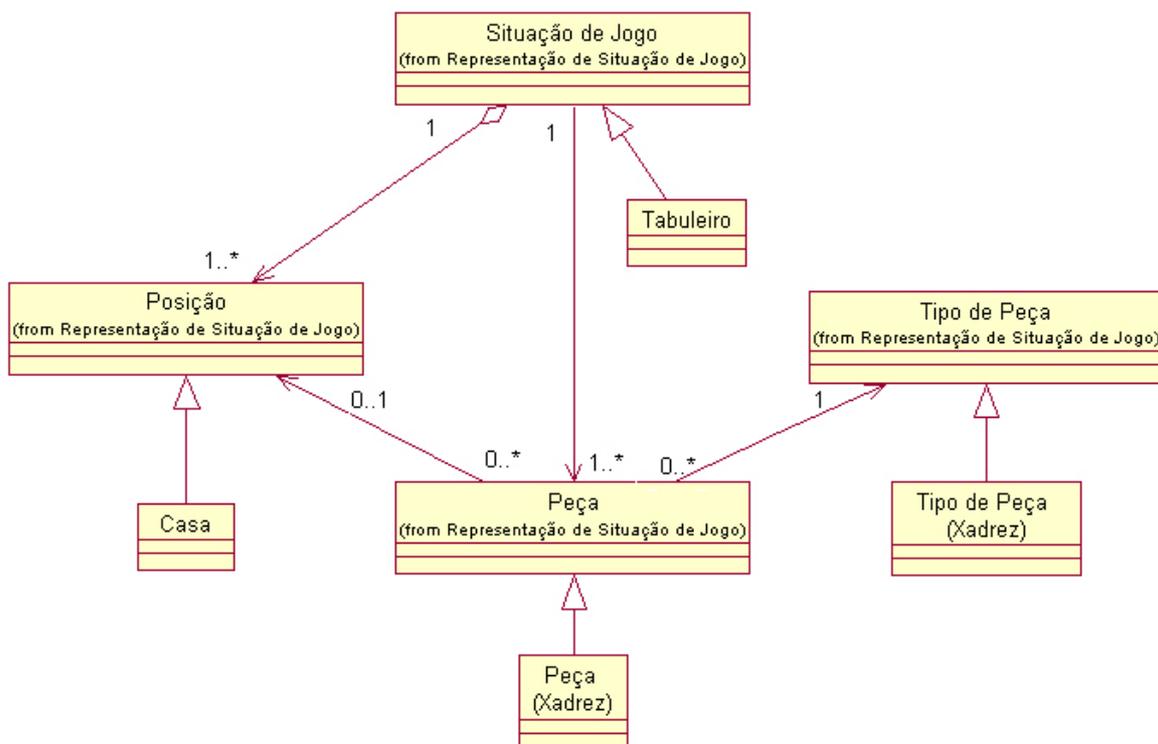


Figura B.1 – Elementos para a Representação de uma Situação de Jogo do Xadrez

B.1.1 Casa

Representa uma posição do tabuleiro de Xadrez, elemento estendido de Posição. Foi utilizado o sistema algébrico como forma de definir o nome das casas do tabuleiro, este sistema divide o tabuleiro em 8 linhas horizontais numeradas de 1 a 8, a começar do lado das peças brancas, e em 8 colunas verticais identificadas com as letras de “a” a “h”, que começam da esquerda para a direita. Este sistema é o utilizado atualmente pela FIDE (Federação Internacional de Xadrez). Na Figura B.1.1 o primeiro Tabuleiro utiliza o sistema algébrico.

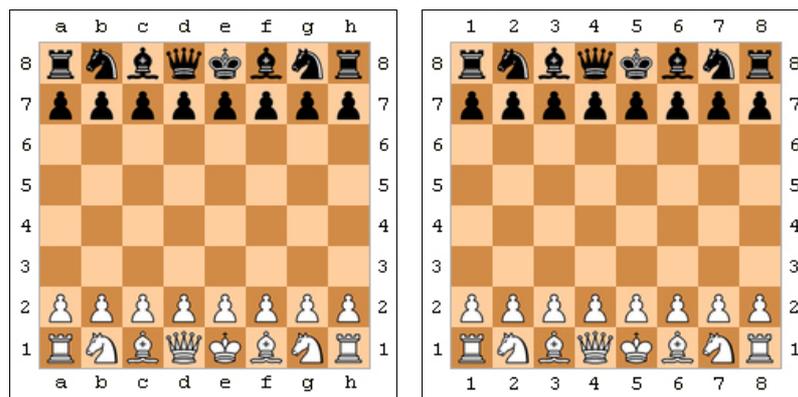


Figura B.1.1 – Sistema Algébrico e Sistema Cartesiano

A fim de permitir referenciar uma casa do tabuleiro também como um ponto em um plano cartesiano, foi introduzido o conceito de coordenado X e Y para uma determinada casa. O tabuleiro possui então pontos no eixo X de 1 a 8, e no eixo Y de 1 a 8, aceitando apenas valores inteiros. Na Figura B.1.1 o segundo tabuleiro apresenta o sistema cartesiano. Um elemento Casa possui os seguintes atributos:

- **TEXTO NOME:** Nome da casa, pode ser um dentre as seguintes valores: A1, A2, A3, A4, A5, A6, A7, A8, B1, B2, B3, B4, B5, B6, B7, B8, C1, C2, C3, C4, C5, C6, C7, C8,

D1, D2, D3, D4, D5, D6, D7, D8, E1, E2, E3, E4, E5, E6, E7, E8, F1, F2, F3, F4, F5, F6, F7, F8, G1, G2, G3, G4, G5, G6, G7, G8, H1, H2, H3, H4, H5, H6, H7 ou H8.

- **INTEIRO X**: Coordenada X, sendo um valor de inteiro de 1 a 8.
- **INTEIRO Y**: Coordenada Y, sendo um valor inteiro de 1 a 8.
- **COR_CASA COR**: Cor da casa, pode ser “BRANCA” ou “PRETA”.
- **CASA CASA_DIREITA**: Casa a Direita desta casa.
- **CASA CASA_ESQUERDA** : Casa a Esquerda desta casa.
- **CASA CASA_FRENTE**: Casa a Frente desta casa.
- **CASA CASA_TRAZ**: Casa a Traz desta casa.
- **PECA PECA_ATUAL**: Peça atualmente nesta casa.

B.1.2 Tipo de Peça

Representa os tipos existentes de peças em um jogo de Xadrez, são eles: PEAO, TORRE, CAVALO, BISPO, DAMA e REI.

B.1.3 Peça

Representa uma peça existente em um jogo de Xadrez. O jogo no seu início possui 32 peças dispostas no tabuleiro. Cada jogador inicia com um total de 16, sendo 8 peões, 2 cavalos, 2 bispos, 2 torres, 1 dama e 1 rei. Sendo que um jogador utiliza peças da cor Branca e o outro peças da cor Preta. Um elemento Peça possui os seguintes atributos:

- **TIPO_PECA TIPO**: Tipo da peça, pode ser PEAO, TORRE, CAVALO, BISPO, DAMA ou REI.
- **DONO_PECA DONO**: Especifica qual o jogador é o proprietário desta peça, pode ser a constante “EU” ou “OPONENTE”.
- **CASA CASA_ATUAL**: Especifica a casa na qual esta peça está posicionada.

B.1.4 Tabuleiro

Este elemento estende “Situação de Jogo” e representa um determinado momento em um jogo de Xadrez. Este elemento é gerado pelo Jogador automático e repassado para o Conjunto Heurístico avaliá-lo. O elemento Tabuleiro possui os seguintes atributos:

- **CASA[] CASAS**: Todas as casas existentes no tabuleiro, armazenadas em um único conjunto.
- **CASA A1, A2, A3, A4, A5, A6, A7, A8, B1, B2, B3, B4, B5, B6, B7, B8, C1, C2, C3, C4, C5, C6, C7, C8, D1, D2, D3, D4, D5, D6, D7, D8, E1, E2, E3, E4, E5, E6, E7, E8, F1, F2, F3, F4, F5, F6, F7, F8, G1, G2, G3, G4, G5, G6, G7, G8, H1, H2, H3, H4, H5, H6, H7 e H8**. Cada casa do tabuleiro do Xadrez tem um acesso direto através de um atributo com o seu nome na descrição algébrica.
- **PECA[] PECAS**: Todas as peças que atualmente estão dispostas no tabuleiro, armazenadas em um único conjunto.
- **TIPO_PECA PEAO, TORRE, CAVALO, BISPO, DAMA e REI**. Cada tipo de peça do Xadrez possui um atributo no Tabuleiro acessando-o direto.
- **INTEIRO QUANTIDADE_LANCES**: Representa quantos lances foram realizados até

o momento no jogo.

- **PECA[] PECAS_EM_PASSANT**: Um conjunto que representa os peões que estão suscetíveis de serem capturados através da jogada “*Captura en-passant*”.
- **LOGICO ROQUE_FEITO_EU**: Valor lógico que indica se o jogador representado por este Conjunto Heurístico já realizou a jogada “*Roque*”.
- **LOGICO ROQUE_FEITO_OPONENTE**: Valor lógico que indica se o jogador que é adversário deste Conjunto Heurístico já realizou a jogada “*Roque*”.
- **REAL VALOR**: Valor heurístico atribuído a esta situação de jogo e que auxiliará o Jogador automático a escolher a próxima jogada.

B.1.5 Mudanças realizadas na linguagem de Codificação de Heurísticas

A fim de adaptar a linguagem de codificação de heurísticas ao contexto do jogo de Xadrez as seguintes alterações foram realizadas:

- As referencias as classes gerais de “Elementos para Representação de Situação do Jogo” (Item A.2) foram substituídas pelos “Elementos para Representação de Situação de Jogo do Xadrez” (Item B.1).
- As seguintes constantes foram incluídas: **BRANCA** e **PRETA**.
- O seguinte domínio foi incluído: **COR_CASA** = {**BRANCA, PRETA**}

REFERÊNCIAS BIBLIOGRÁFICAS

- AINSWORTH, Shaaron (1999). *Designing effective Multi-representational Learning Environments*. University of Nottingham.
- ARRIADA, M. C. (2001). *Critérios para a análise de ferramentas computacionais de apoio à aprendizagem cooperativa*. Dissertação de Mestrado, UFSC.
- BOFF, E (2000). *Construindo um ambiente de ensino-aprendizagem cooperativo: uma experiência interdisciplinar*. Em Anais do XI SBIE – Simpósio Brasileiro de Informática na Educação, Maceió,AL, pp. 112-19.
- BORGES, M. A. F. (2001). *Analysing interaction in a collaborative game: a case study*. Em Anais do XII SBIE – Simpósio Brasileiro de Informática na Educação, Vitória, ES, pp. 185-93.
- BRATKO, I., Mozeitie, I., Lavrae, N. (1990). *Kardio: a study in deep and qualitative knowledge for expert systems*, MIT Press, Cambridge, MA, USA.
- CEX (2005). Centro de Excelência de Xadrez, página web <http://www.cex.org.br> acessada em agosto de 2005.
- CHURCH, A. (1956). *Introduction to Mathematical Logic*, Princeton, New Jersey:Princeton University Press.
- DIRENE, A., Bona, L., Silva, F., Dos Santos, G., Guedes, A., Castilho, M., Sunyé, M., Hartmann, C., De Andrade Neto, P., de Mello, S., Sunyé Neto, J. E Silva, W. (2004). *Conceitos e ferramentas de apoio ao ensino de xadrez nas escolas brasileiras*. Em Anais do XXIV Congresso da Sociedade Brasileira de Computação: WIE – Workshop sobre Informática na Escola, Salvador, Brasil, R. Macedo, Ed., SBC, pp-816-825.

- DOUBEK, J. (1982). *Xadrez para Principiantes*, Tecnoprint.
- FERREIRA, J. S. e Labidi, S. (1998). *Modelagem do aprendiz baseado no paradigma de ensino cooperativo*. Em Anais do IX SBIE – Simpósio Brasileiro de Informática na Educação Fortaleza, CE, pp. 494-505.
- FIDE (2006). *Fédération Internationale des Échecs* (Federação Internacional de Xadrez), <http://www.fide.com/ocial/handbook.asp?level=EE101>, site acessado em agosto de 2006.
- GAVA, T. B. S. e De Menezes, C. S. (2003). *Uma ontologia de domínio para a aprendizagem colaborativa*. In Anais do XIV SBIE { Simpósio Brasileiro de Informática na Educação, p.355-364, Rio de Janeiro/RJ.
- GIRAFFA, L. M. M. (1999). *Uma Arquitetura de Tutor Utilizando Estados Mentais*. Tese de Doutorado, Porto Alegre: CPGCC/UFRGS.
- GREENO, J. G., Collins, A., Beranek, B., & Resnick, L. B. (1994) *Cognition and Learning*. In D. Berliner & R. Calfee (Eds.), *Handbook of educational psychology*, pp. 1-51.
- HOLZNER, S. (2001). *Desvendando XML*, Campus.
- HÜBSCHER-YOUNGER, T., Narayana, N. H. (2001). *Features of Shared Student-Created Representations*, em AI-ED 2001, Workshop External representations in AIED, San Antonio, Texas.
- JOHNSON, D. W. e Jhonson, T. J. (1989). *Cooperation and competition: Theory and research*. Interaction Book Company, Edina, MA.
- JOHNSON, D. W., Johnson T. J., e Smith, K. A. (1991) *Active Learning: Cooperation in the College Classroom*. Interaction Book Company, Edina, MA.
- JOHNSON, D. W. e Jhonson, T. J. (1994). *Learning together and alone: Cooperative, competitive, and individualistic learning*. Allyn & Bacon, Needham Heights, MA.

- JOOLINGEN, W. V., Löhner, S. (2001). *Representations in collaborative modeling tasks*, em AI-ED 2001, Workshop External representations in AIED, San Antonio, Texas.
- LABEKE, N. (2001). *Multiple External Representations in Dynamic Geometry: a Domain-Inspired Design*, em AI-ED 2001, Workshop External representations in AIED, San Antonio, Texas.
- MARTINESCHEN, D. (2006). *Alternância entre competição e colaboração para promover o aprendizado por meio de heurísticas de jogos*, Em dissertação de conclusão de Mestrado da Universidade Federal do Paraná.
- MARTINESCHEN, D., Direne, A., Bona, C., Silva, F., Castilho M., Guedes, A., Sunyé, M. (2006). *Alternância entre competições e colaboração para promover o aprendizado por meio de heurísticas de jogos*. Anais do XXVI Congresso da Sociedade Brasileira de Computação. WIE – Workshop sobre Informática na Escola, SBC, Campo Grande, Brasil, Julho.
- MURRAY, T., Winship, L., Bellin, R., Cornell, M. (2001). *Toward Glass Box Educational Simulations: Reifying Models for Inspection and Design*, em AI-ED 2001, Workshop External representations in AIED, San Antonio, Texas.
- OLGUÍN, C. J. M., Delgado, A. L. N., Botero, S. W. e Ricarte, I. L. M. (2000). *O uso de agentes em ambientes de aprendizagem colaborativos*, em Anais do XI SBIE – Simpósio Brasileiro de Informática na Educação, Maceió, AL, pp. 236-43.
- PRESSMAN, R. S. (1995). *Engenharia de Software*, Makron Books, São Paulo.
- QIN, Z., Jhonson, D. W e Jhonson, T. J. (1995). *Cooperative versus competitive efforts and problem solving*. Reviews of Educational Research, 129-43.
- RAMOS, E. M. F. (1999). *O papel da avaliação educacional nos processos de aprendizados autônomos e cooperativos*. Capítulo do Livro LINSINGEN *et alli* (editores) A Formação do Engenheiro. Editora da UFSC, Florianópolis.

- RICH, E. Knight K. (1993). *Artificial Intelligence*. McGraw-Hill.
- ROSATELLI, M. C. (2000). *Novas tendências da pesquisa em inteligência artificial na educação*. In R. C. Nunes (Ed.), VIII Escola de Informática da SBC Sul, pp. 179-210. Porto Alegre: Editora da UFRGS.
- ROSATELLI, M. C.; Thiry, M.; Barcia, R.M. (2000). *Um Ambiente Colaborativo para o Ensino a Distância*, Revista Brasileira de Informática na Educação, n.6, p.9-20.
- ROUSE III, R. (2001). *Game Design – Theory & Practice*, Wordware Publishings, Inc.
- SCHÄFER, H. E Direne A. I. (2000). *Conceitos e ferramentas para apoiar o ensino de xadrez através de computadores*. Em Simpósio Brasileiro de Informática na Educação – SBIE2000, Maceió/AL, pp. 97-104.
- SKINNER, B.F. (1958) *Teaching Machines*. Science, 128.
- SOUZA, F. R. D., Ferrari, R. H. e Lenzi, R. L. A. (2004). *Meta-linguagem e ferramentas para a definição de heurísticas para jogos*. Em Monografia de conclusão de curso Universidade Federal do Paraná.
- SWIDERSK, Z., Parkes, A., (2001). *A Volume Mechanism for Content-Based Control of Multimedia*, em AI-ED 2001, Workshop External representations in AIED, San Antonio, Texas.
- TEDESCO, P. (1999). *Mediating Meta-Cognitive Conflicts in a Collaborative Problem-Solving Environment*. Le Mans: In the Proceedings of Young Researchers' Track, 9th International Conference on Artificial Intelligence in Education, p. 43-44, July.
- W3C (2005). World Wide Web Consortium, página web <http://www.w3c.org> acessada em agosto de 2005.

WENGER, Etienne, (1987). *Artificial Intelligence and Tutoring Systems: Computational and Cognitive Approaches to the Communication of Knowledge*. Morgan Kaufmann Publishers, Los Altos, California.

WILLIAMS, B. (2001). *The Role of External Representations in Intelligent Tutoring System authoring: Supporting localised decision making in a complex and evolving global context*, em AI-ED 2001, Workshop External representations in AIED, San Antonio, Texas.

ZHANG, J. & Norman, D. A. (1994). *Representations in distributed cognitive tasks*. Cognitive Science, 18, 87-122.

ZHANG, J. (1997). *The nature of external representations in problem solving*. Cognitive Science, 21(2), 179-217.

ZHANG, J. (2001). *External Representations in Complex Information Processing Tasks*, em Encyclopedia of Library and Information Science, Marcel Dekker, New York.