

Construindo o Módulo do Domínio de um Assistente Inteligente Utilizando a Ferramenta Java Expert System Shell

Luiz Henrique Zeferino^{1,2}, Clevi Elena Rapkiewicz¹, Gudelia Morales¹

¹Laboratório de Engenharia de Produção – Universidade Estadual do Norte Fluminense (UENF)
Av. Alberto Lamego, 2000 – 28015-620 – Campos – RJ – Brazil

²Fundação de Apoio à Escola Técnica (FAETEC)
Av. Clarimundo de Melo, 847 – 21230-280 – Rio de Janeiro – RJ – Brazil

{lhzefero, clevis, gudelia}@uenf.br

Resumo. Este artigo descreve o desenvolvimento do módulo do domínio de um assistente inteligente para Educação Matemática implementado com a ferramenta JESS (Java Expert System Shell) que utiliza o algoritmo RETE e a linguagem Java. O domínio do assistente proposto é constituído pelas Seções Cônicas: a parábola, a elipse e a hipérbole. É descrita a modelagem do módulo do domínio desse assistente bem como o funcionamento básico desse componente na interação desse assistente com o estudante. Além da implementação dos exercícios que constituem o domínio, utilizando-se a ferramenta JESS, destaca-se a solução adotada para a comunicação entre os ambientes JESS e Java.

Palavras-chave: Inteligência Artificial, Módulo do Domínio, Educação Matemática.

Abstract. This paper describes the development of the module of the domain of an intelligent assistant for Mathematical Education implemented with the tool JESS (Java Expert System Shell) that uses the algorithm RETE and the language Java. The proposed assistant's domain is constituted by the Conical Sections: the parabola, the ellipse and the hyperbole. The modelling of the module of that assistant's domain is described as well as the basic operation of that component in that assistant's interaction with the student. Besides the implementation of the exercises that they constitute the domain, being used the tool JESS, it stands out the solution adopted for the communication among the tools JESS and Java.

Key words: Artificial intelligence, Domain Expert, Mathematical Education.

1. Introdução

O tópico Seções Cônicas (integrante da Geometria Analítica) não é lecionado (quando o é) de maneira eficiente na terceira série do Ensino Médio. Esse problema revelou-se pela prática docente (de uma década) de um dos autores deste trabalho e pela análise de livros textos brasileiros de Matemática para o Ensino Médio, efetuada por [Lima 2001]. Embora existam inúmeros fatores que acarretam tal problema, para o professor cumprir esse tópico podem ser destacas duas barreiras a serem ultrapassadas:

- i) o número de pré-requisitos é grande, pois esse tópico utiliza a Matemática Numérica, a Simbólica e a Gráfica;
- ii) o número elevado de alunos por turma (até 60 alunos), o que não permite ao professor uma atuação adequada junto aos alunos que apresentam maiores dificuldades.

A utilização de um Sistema Tutor Inteligente (STI) implica em melhorar a aprendizagem desse importante tópico da Matemática do Ensino Médio. Um STI permite a emulação de um professor, ou seja, um STI sabe o que deve ensinar, como deve ensinar, e ainda é capaz de obter informações pertinentes sobre o estudante que está aprendendo, através de uma avaliação contínua deste.

Sistemas Tutores Inteligentes (STIs) podem ser divididos em Tutores Inteligentes e Assistentes Inteligentes. Na prática essa divisão é necessária em função do porte do STI: Assistentes Inteligentes são pequenos sistemas tutores inteligentes para realizar tarefas simples e são implementados em microcomputadores, sendo que o modelo de aluno simplificado e o conjunto de estratégias é restrito [Pinto 1995]. A estratégia pedagógica utilizada pelo STI na interação com o usuário também diferencia um Assistente Inteligente de um Tutor Inteligente. No assistente a estratégia é baseada na monitoração do usuário, nos resultados obtidos ao longo da interação e com poucas interrupções por parte do assistente, o qual busca estimular o aluno a utilizar os recursos do ambiente e a refletir sobre seus passos. Já no tutor inteligente a estratégia pedagógica utilizada pelo ambiente é mais invasiva e apresenta um maior número de interrupções por parte do tutor na interação com o aluno.

O presente artigo destaca a representação do conhecimento em um assistente inteligente desenvolvido para o ensino das seções cônicas, ou seja, a elipse, a parábola e a hipérbole. Apresenta-se a *shell* para sistemas especialistas JESS (Java Expert System Shell) utilizada na criação e controle de fatos e regras que constituem os exercícios que formam o domínio do assistente inteligente desenvolvido.

Este artigo é composto de quatro seções, além da presente introdução. A seção 2 apresenta a arquitetura adotada e algumas considerações sobre a modelagem do domínio aplicada por Zeferino (2003) na implementação de seu assistente inteligente desenvolvido para o ensino das Seções Cônicas. Na seção 3 são apresentados a *shell* JESS e o algoritmo RETE, bem como a forma como foram elaborados os exercícios que constituem o sistema desenvolvido. Na seção 4 descreve-se a comunicação entre os componentes utilizados na implementação do sistema proposto. A seção 5 constitui-se das considerações finais e sugestões para trabalhos futuros.

2. Arquitetura do Assistente Inteligente Proposto

A arquitetura adotada é a clássica [Giraffa 1999] ou funcional tripartida formada pelos componentes: Módulo do Conhecimento do Domínio, Módulo Tutorial ou Instrucional, Módulo de Controle, Modelo do Estudante e Interface com o usuário (aluno). A Figura 1 apresenta a arquitetura adotada. Neste artigo é discutido o componente Módulo do Domínio, bem como o seu desenvolvimento utilizando a *shell* JESS e a orientação a objetos (Java).

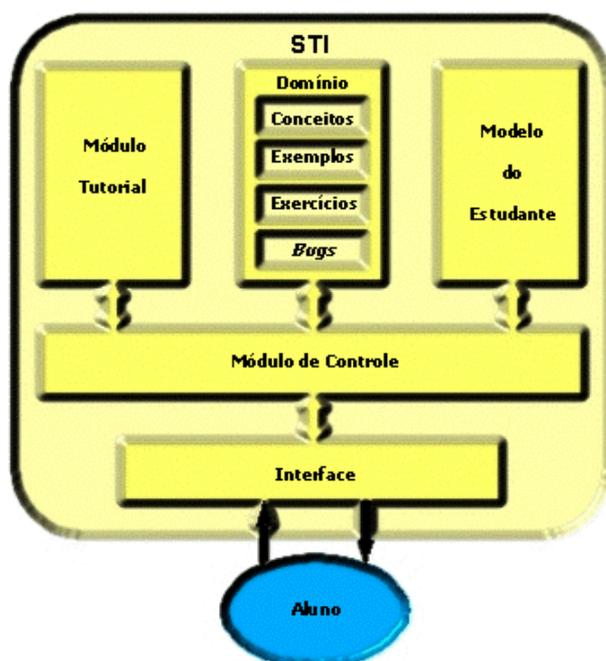


Figura 1: Arquitetura utilizada no assistente

2.1. Domínio do Ambiente Proposto

A base do conhecimento do domínio contém o material a ser ministrado pelo assistente e foi estruturada nas categorias: Conceitos, Exemplos, Exercícios e *Bugs*, conforme ilustrado na figura 1. A categoria **Conceitos** representa definições, demonstrações e princípios matemáticos - destacando que a conceituação é indispensável para o bom resultado das aplicações da teoria matemática [Lima 2001]. A categoria **Exemplos** contém o material ilustrativo que pode ser exibido através de texto, gráficos, animações, etc. A categoria **Exercícios** está dividida em duas partes: uma com exercícios relacionados à manipulação algébrica e a outra a exercícios de aplicação - visto que a maior deficiência do livro didático brasileiro do Ensino Médio, revelada pela análise dos mesmos [Lima 2001], é a falta de exemplos e exercícios de aplicação, o que de certa forma contraria os Parâmetros Curriculares Nacionais brasileiros do Ensino Médio (PCNs). Finalmente, a categoria **Bugs** contém o conhecimento acerca de alguns erros típicos cometidos pelos estudantes na teoria em estudo [Hasegawa e Nunes 1995].

O assistente inteligente funciona a partir da oferta de exercícios ao estudante. Este na busca da solução desses exercícios é levado à descoberta dos recursos disponíveis no sistema, bem como à busca dos conceitos matemáticos necessários para essa solução. A relação entre as categorias que constituem o domínio do assistente e citadas anteriormente é ilustrada na Figura 2.

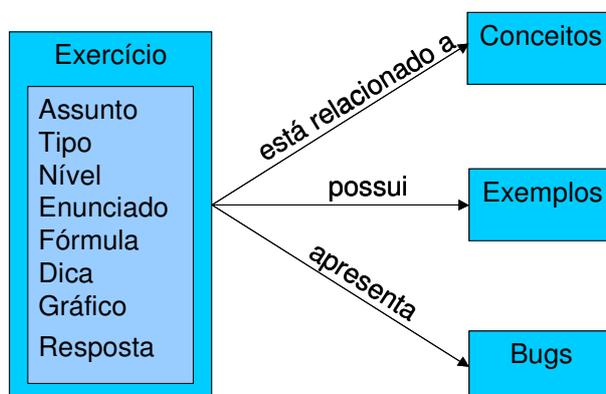


Figura 2: Relação entre os elementos constituintes do domínio do assistente.

O módulo do domínio é, geralmente, considerado o componente central de qualquer STI. Em essência, este modelo incorpora a maior parte da "inteligência" do sistema na forma do conhecimento necessário para solucionar problemas do domínio [Anderson 1988]. Desta forma, a identificação das

categorias de conhecimento do domínio é de suma importância para sua modelagem computacional [Ladeira e Viccari 1996].

No assistente proposto estão presentes três categorias de conhecimento: a declarativa - que está representada pelos conceitos matemáticos estruturados, a procedural (do procedimento) - que representa o raciocínio necessário para realizar uma tarefa específica e a categoria heurística - que define o conjunto de operações executadas para resolver problemas.

2.1.1. Características do Domínio do Assistente Proposto

Um aspecto importante sobre o domínio do assistente proposto é a possibilidade da exploração gráfica das curvas que o constitui: a parábola, a hipérbole e a elipse - tal característica é um ponto que pode ser explorado com riqueza de recursos no ambiente computacional - através de construções gráficas e animações, o que é, ao mesmo tempo, um fator atraente ao aluno.

Uma outra característica importante é a possibilidade de exploração de exercícios de aplicação na própria Matemática e em outras disciplinas como a Física, a Química e a Biologia, pois empregos das Cônicas estão presentes no estudo da Função Quadrática, deslocamento de projéteis, refletores elípticos, parabólicos ou hiperbólicos, acústica física e biológica, expansão de gases, movimentos de corpos celestes, entre outros. Tal característica pode ser explorada com riqueza de recursos no ambiente computacional através de gráficos interativos, animações, hipertexto, etc.

Quanto aos exercícios, são definidos dois tipos básicos. Um que consiste da aplicação direta de uma fórmula específica. Dependendo do caso, a fórmula pode apresentar um, dois ou três parâmetros - como por exemplo o cálculo de uma coordenada do foco ou vértice. Outro tipo de exercício consiste na utilização de uma seqüência padrão de passos para se obter a solução - caso da utilização do método de completar o quadrado.

A Tabela 1 fornece exemplos desses tipos de exercícios. Essa classificação é necessária pois cada tipo de exercício, segundo esse enfoque, necessita de suas próprias formas (ou regras de produção) de análise, além das regras de produção gerais, utilizadas pelo assistente para mapear a ação do aluno. Na realidade são fatos comparados às regras de produção pela máquina de inferência da ferramenta JESS, para que o sistema identifique que tipo de assistência de escolher para auxiliar no aprendizado do aluno, de acordo como as estratégias disponíveis e as informações que o sistema possui sobre o aluno num dado momento de interação.

Tabela 1: Exemplos de exercícios quanto ao método de solução.

Tipo de Problema	Fórmula ou Método para resolução	Exemplo de Problema
1- Problema com um parâmetro	$p = \left \frac{1}{4a} \right $	Determine a distância entre o foco e o vértice da parábola de equação $y=2x^2+3x+8$.
2- Problema com dois parâmetros	$x_v = \frac{-b}{2a}$	Um fabricante produz semanalmente x unidades de um determinado artigo. Sabendo que o lucro da venda das x unidades é determinado pela expressão $L(x)=-4x^2+32x-8$, determine o número x de unidades que fornece lucro semanal máximo.
3- Problema com três parâmetros	$y_v = \frac{-b^2 + 4ac}{4a}$	Um golfinho salta sobre a superfície do mar, que pode ser representada pelo eixo x no plano cartesiano y_0x (x e y dados em metros), conforme a parábola de equação $y=-(x^2-4x)/2$. Determine a altura do salto do golfinho em metros.
4- Problema resolvido com um conjunto de passos	Completar o quadrado	Determine as coordenadas do vértice da parábola de equação $y=x^2+2x$, utilizando o método de completar o quadrado, ou seja, escrevendo essa equação sob a forma $y=a(x-x_v)^2+y_v$.

Para construção do domínio foram utilizadas a linguagem Java (orientação a objetos), hipertexto e a ferramenta JESS. O hipertexto foi utilizado como base para transmissão de conceitos e exemplos (conhecimento declarativo) no módulo de domínio devido a possibilidade de utilização de animações, *links*, figuras, *gifs* animados, etc. O ambiente Java foi utilizado na criação das janelas interativas dos gráficos necessários ao domínio do assistente desenvolvido, bem como na implementação do módulo de controle e na interface do assistente. Para a criação da base de exercícios e métodos para controlar a solução dos mesmos na interação do assistente com o estudante - o que constitui o coração do sistema - foi utilizada a ferramenta JESS, que será descrita a seguir.

3. A Ferramenta JESS e o Algoritmo RETE.

JESS é uma *Shell* para sistemas especialistas totalmente escrita em linguagem Java no *Sandia National Laboratories* [Herzberg 2002]. Essa ferramenta foi originalmente inspirada na *Shell CLIPS (C Language Integrated Production System)* desenvolvida pelo *Software Technology Branch (STB)*, NASA/Lyndon B. Johnson Space Center, sendo esta ferramenta elaborada para facilitar o desenvolvimento de software que modele o conhecimento de um ser humano ou especialista [Giarratano 2002].

JESS pode ser utilizada de duas maneiras: como uma linguagem de programação, visto que pode acessar todas as classes e bibliotecas Java; e como uma máquina de inferência, ou seja, um conjunto de algoritmos que aplica de maneira eficiente as regras aos fatos. A opção efetuada na implementação do assistente foi a segunda, visto que o ambiente proposto não é um sistema especialista e sim um assistente inteligente.

A ferramenta JESS permite com facilidade, além das regras de produção, a utilização de funções predefinidas e a definição de funções dentro do próprio ambiente JESS. Outro ponto importante deste ambiente é a possibilidade da importação de funções pré-definidas ou implementadas pelo usuário no ambiente Java.

A máquina de inferência JESS utiliza o algoritmo RETE, considerado muito eficiente, empregado por linguagens baseadas em regras, tais como CLIPS, ART, OPS5, OPS83 *apud* [Borba e Fernandes 2002], [Forgy 1982]. O objetivo desse algoritmo é a unificação dos padrões nas regras, com a finalidade de determinar quais regras são satisfeitas.

Se o processo de unificação tivesse de ocorrer uma única vez (o que é improvável), a solução seria simples: a máquina de inferência examinaria cada regra e procuraria o conjunto de fatos que satisfariam os padrões de uma específica regra e, então, as regras satisfeitas seriam selecionadas. Entretanto, em linguagens baseadas em regras, o processo de unificação ocorre repetidamente. Quase sempre a lista de fatos é modificada: novos fatos podem ser acrescentados e/ou fatos antigos podem ser removidos da lista de fatos inicial. Desta forma, o problema da unificação torna-se contínuo: embora o conjunto de regras permaneça constante, o conjunto de fatos está permanentemente em mudança. Verifica-se que essa técnica é muito lenta pois a máquina de inferência deve checar cada fato, a cada ciclo de execução, para cada regra (o que caracteriza a chamada redundância temporal). O que o algoritmo RETE faz é tirar vantagem dessa característica: salva o estado de unificação do ciclo anterior e calcula as mudanças no conjunto de fatos. Neste caso, os fatos procuram as regras. Se a cardinalidade do conjunto de fatos que muda é razoável, o ganho na redução do número de iterações é significativo no caso do algoritmo RETE.

A desvantagem do algoritmo RETE é o uso intensivo de memória: o uso de memória para simplesmente comparar todos os fatos com todas as regras é desprezível (embora o número de iterações possa ser elevado). Já a quantidade de memória necessária para armazenar todo o ciclo de unificação (anterior) e as alterações no conjunto de fatos é considerável.

Outra vantagem do algoritmo RETE destacada por Borba e Fernandes (2002) é que esse algoritmo usa a similaridade estrutural presente nas regras. Esta similaridade refere-se ao fato de que regras podem ter padrões ou grupos de padrões similares. O RETE tira vantagem desta característica colocando os componentes comuns agrupados, para que os mesmos não sejam considerados mais de uma vez, aumentando assim a eficiência.

Quanto ao encadeamento, o *default* da máquina de inferência dessa ferramenta é o encadeamento para frente (*forward-chaining*), sendo esse encadeamento dirigido para os dados e usado principalmente nos sistemas em tempo real de verificação de falhas - o que é indicado para a utilização em um conjunto de regras que codificam a experiência de especialistas na resolução de problemas (caso do ensino da Matemática, por exemplo). Entretanto é permitido o encadeamento para trás (*backward-chaining*) através de um comando especial.

A ferramenta JESS permite com facilidade definir *bugs*, bem como as regras do módulo do domínio que determinam tais *bugs*. No sistema os erros típicos do aluno são determinados de duas maneiras: a primeira consiste na criação de fatos que representam esses erros típicos e as regras que encontram esses erros; a segunda consiste na criação de regras e sub-regras para identificarem tais erros típicos. A diferença e a justificativa pela opção de dois tipos de modelagem (simplificadas aqui para melhor entendimento) podem ser compreendidas através da solução do exercício a seguir:

"Um parábola possui equação $y=5x^2-3x+10$, determine a coordenada x do vértice dessa parábola."

É sabido que a determinação da coordenada x do vértice é dada por $xv = -b/(2*a)$. No caso do exercício em questão $a = 5$ e $b = -3$. São exemplos de erros comuns, na aplicação desses valores na fórmula para cálculo:

- (a) esquecimento ou não uso de parênteses;
- (b) trocar os valores de a por b e de b por a ;
- (c) trocar o sinal de a ;
- (d) trocar o sinal de b .

Podem então ser criados os fatos *bugs* desse exercício:

Bug $-b/2*a$

Bug $-a /2*b$

Bug $-b/(2*(-a))$

Bug $b/(2*a)$

A regra para esses *bugs* seria:

Se ocorrer *Bug* X

Então informar ao sistema para a escolha da estratégia referente ao *Bug* X

Quando o aluno envia uma expressão para o sistema, caso essa seja um desses quatro casos de *bugs* (fatos) - o que pode ser considerado rápido - semelhante a rapidez com que um professor experiente consegue detectar (ou quase prever) erros habituais de alunos.

Contudo, um aluno pode cometer erros que necessitem de um conjunto de regras pois, devido aos erros múltiplos como por exemplo esquecer os parênteses e trocar os sinais de a e b , o número de variações pode crescer combinatoriamente. Neste caso, são necessárias regras auxiliares ou sub-regras que permitem determinar o erro ou erros cometidos. Essas são na verdade as regras do elemento inverso, das propriedades associativa, distributiva, comutativa e elemento neutro, regra que verifica o uso de parênteses. A sistemática nesse caso é similar à utilizada por Heffernan (2001): percorre-se a expressão enviada pelo aluno ao sistema procurando primeiro os parâmetros; em seguida seus valores, posição, operadores e uso de parênteses, se for o caso - utilizando para isso as regras auxiliares. Compare-se então as duas expressões: a do aluno e a do módulo do domínio. Neste caso o número de iterações é elevado, quando comparado à idéia do primeiro caso em que um erro é visto como um fato (que procura por apenas uma regra específica). Como um exemplo da sistemática acima comentada: o domínio deve reconhecer que, entre outras possibilidades,

$-(-4)/(2*3)$, $4/(2*3)$,

$-(-4)/(3*2)$, $4/(3*2)$,

$-(-4)/6$ e $4/6$,

representam $-b/(2*a)$, com $a = 3$ e $b = -4$. Neste caso se o aluno enviasse ao sistema $-4/2*3$ o módulo do domínio detectaria os erros: troca do sinal de b e falta de parênteses ao redor de $2*a$ e informaria o sistema de tais fatos para que esse utilizasse a estratégia adequada. Essa estratégia adequada está definida no módulo tutorial [Zeferino 2003].

4. Comunicação entre o Ambiente Constituintes do Assistente

Para a comunicação entre os elementos na implementação do ambiente procurou-se isolar ao máximo os elementos adotados, conforme ilustrado na Figura 3.

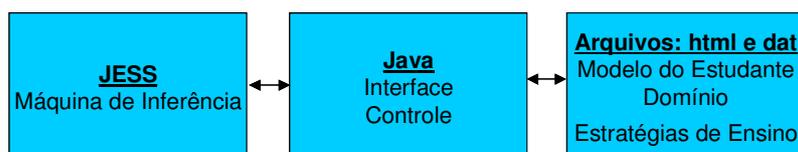


Figura 3: Comunicação e Funções dos Ambientes da implementação do Assistente.

Tal sistemática para a comunicação possui três objetivos: i) facilitar ao máximo a possibilidade de alteração/ampliação dos módulos constituintes do assistente, como o domínio, estratégias de ensino, recursos gráficos, etc.; ii) permitir a substituição de um componente por outro equivalente, como exemplo substituir o conjunto de arquivos .dat por um banco de dados; iii) possibilidade da inclusão de novos elementos no assistente, como exemplos podem ser citados um ambiente visual para edição de exercícios, estratégias de ensino, criação de hipertexto, etc. e um componente que permita a comunicação através de recursos sonoros.

Quanto às dificuldades encontradas referentes ao trabalho com a ferramenta JESS podem ser destacadas duas: a primeira refere-se a dificuldade de encontrar-se material sobre essa ferramenta no Brasil e no Exterior, principalmente quanto à área de Matemática; a segunda refere-se à dificuldade da comunicação entre todos os módulos da arquitetura adotada, conforme ilustra a Figura 1, tomando-se como base o módulo de controle – a qual pode ser considerada a maior dificuldade encontrada na implementação, inicialmente.

Um outro aspecto importante para a implementação é que essa forma de comunicação e a utilização do JESS como uma máquina de inferência permite a utilização de uma ferramenta visual para a implementação da janelas que constituem o assistente implementado – no caso optou-se pelo ambiente *FORTE for Java* – o que permite maior velocidade na implementação do sistema, proporciona-se uma comunicação mais amigável entre o sistema e o aluno. Permite-se também que sejam feitos testes para os respectivos ajustes para analisar-se o desempenho do sistema como um todo, impondo-se assim maior confiabilidade ao sistema.

5. Considerações Finais

O sistema implementado reuniu conhecimentos na área de Inteligência Artificial e Educação Matemática e funciona a partir da oferta de exercícios ao estudante. O aluno na busca da solução desses exercícios é levado à descoberta dos recursos disponíveis no sistema, bem como à busca dos conceitos matemáticos que envolvem essa solução. Para tanto estão disponíveis recursos de assistência diversos. O sistema visa auxiliar o trabalho docente de professores do Ensino Médio de Matemática quanto a um tópico da Geometria Analítica – As Seções Cônicas.

A pesquisa preliminar desenvolvida sobre os recursos disponíveis para a Educação Matemática (livros, software e sites), sendo identificados pontos negativos e positivos desses recursos, aliada ao estudo da forma de estruturação de conteúdos para o ensino de Matemática (conceitos, exemplos,

exercícios e *bugs*) foi de suma importância para a concretização da proposta deste trabalho. Verificou-se a necessidade de dividir os exercícios propostos aos estudantes pelo sistema em duas partes ou categorias: uma com exercícios relacionados à manipulação algébrica e a outra com exercícios de aplicação dentro da própria Matemática ou em outras disciplinas como a Física, Química ou Biologia - visto que a maior deficiência do livro didático brasileiro do Ensino Médio, revelada pela análise dos mesmos [Lima 2001], é a falta de exemplos e exercícios de aplicação, o que não está de acordo com as recomendações dos Parâmetros Curriculares Nacionais brasileiros do Ensino Médio (PCNs). Verificou-se também a necessidade da construção de um catálogo de *bugs* (conhecimento acerca de alguns erros típicos cometidos pelos estudantes no ensino das Seções Cônicas) para um bom funcionamento do sistema.

Na implementação do sistema optou-se pela linguagem Java, por ser uma solução não-proprietária e, também, por apresentar uma grande quantidade de códigos fonte livres em matemática (gráficos, manipulação algébrica, animações, *parser*, etc.), tal conclusão surgiu a partir da pesquisa preliminar realizada do *sites* que envolvem as Seções Cônicas. A opção por essa linguagem indicou, conseqüentemente, a utilização da ferramenta JESS, visto que esta apresenta compatibilidade satisfatória com o ambiente Java diante das necessidades exigidas pelo sistema proposto.

Quanto às dificuldades encontradas do desenvolvimento da proposta, pode ser destacado o atual estado da arte em IA, principalmente quanto à dificuldade de modelagem dos aprendizes. Na realidade a maior dificuldade deste trabalho reside no fato de apesar de existirem e serem conhecidos recursos que possam ser apresentados ao aluno durante o processo de assistência, a determinação do momento em que o aluno necessita desse recurso (diagnóstico das necessidades de assistência ao aluno em uma dada situação) é o ponto mais fraco do sistema planejado.

Essa dificuldade ocorreu neste trabalho, em parte, porque embora o ambiente JESS seja escrito em linguagem Java, houve a necessidade de serem pesquisadas formas de solução para a comunicação adequada entre esses dois ambientes, visto que o JESS foi projetado para a criação de Sistemas Especialistas e não para Assistentes Inteligentes, ou seja, foi feita uma adaptação após verificar-se a escassez de material (trabalhos ou artigos) sobre essa ferramenta no Brasil e pouco material no exterior e também pelo fato de sua interface ser em DOS, sendo isso também constatado por Borba e Fernandes (2002) para área médica e que foi constatado para área do ensino de matemática. Uma sugestão seria trabalhar-se na criação de um componente equivalente ao JESS mas direcionado a um STI, visto que um Assistente Inteligente possui os mesmos componentes e características de um STI [Pinto 1995].

Referências

- Anderson, J. (1988) The Expert Module. In: Polson, M., Richardson, J. (eds.) Foundations of Intelligent Tutoring Systems, New Jersey: Lawrence Erlbaum Associates, p. 21-51.
- Borba, C. A., Fernandes, A. M. R. (2002) “Desenvolvimento de um Sistema Inteligente para Diagnóstico Diferencial por Imagens Utilizando Java Expert System Shell e o Algoritmo Rete”. Anais do II Workshop de Informática Aplicada à Saúde.
- Forgy, C. L. (1982) “Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem”. Artificial Intelligence, 19:17-37.
- Giarratano, J.C, (2002) CLIPS – “User's Guide”; <http://www.ghg.net/clips/download/documentation>, acesso em 12 de dezembro de 2002.
- Giraffa, L. M. M. (1999) Uma Arquitetura de Tutor Utilizando Estados Mentais. (Tese de Doutorado) - Porto Alegre: UFRGS.
- Hasegawa, R. e Nunes, M. (1995) “TOOTEMA: Uma Ferramenta para a Construção de Sistemas Tutores Inteligentes em Matemática”, Anais do VI Simpósio Brasileiro de Informática na Educação. Florianópolis.
- Heffernan, N. T. (2001) Intelligent Tutoring Systems Have Forgotten the Tutor: Adding a Cognitive Model of Human Tutors. (Tese de Doutorado) - Pittsburgh: Carnegie Mellon University.

- Herzberg, C. "JESS, The Java Expert System Shell". Disponível em <http://herzberg.ca.sandia.gov/jess>, acesso em 12 de dezembro de 2002.
- Ladeira, M., Viccari, R. M. (1996) "Representação de Conhecimento Incerto", XV Jornada de Atualização em Informática - XVI Congresso da Sociedade Brasileira de Computação, Recife, 56p.
- Lima, E. (2001) Exame de Textos - Análise de Livros de Matemática para o Ensino Médio. Rio de Janeiro: IMPA/SBM, 467p.
- Pinto, S.C. (1995) M-Assiste: Um Meta-Assistente Adaptativo para suporte à Navegação em Documentos Hiperfídia. (Dissertação de Mestrado) - Rio de Janeiro: UFRJ.
- Zeferino, L. H. (2003) Um Assistente Inteligente para o Ensino das Seções Cônicas: Modelagem e Prototipação. (Dissertação de Mestrado). Campos dos Goytacazes: UENF.