Implementação de uma Ferramenta de Apoio à Criação de Sistemas Especialistas

Fabrício Kelmer Pinto, Elio Lovisi Filho (Orientador)

Departamento de Ciência da Computação - Faculdade de Ciência da Computação e Comunicação Social (FACICS) - Universidade Presidente Antônio Carlos - Campos Magnus - Campolide - MG

fabriciotobe@hotmail.com, professor_elio@nextwave.com.br

Resumo. *Os Sistemas Especialistas são aplicativos utilizados na construção de máquinas que auxiliam os profissionais de diversas áreas. A construção de um sistema genérico pode auxiliar no aprendizado da matéria e na solução de casos simples. Este artigo exibe a elaboração de um sistema genérico de criação de Sistemas Especialistas.*

1. Introdução aos Sistemas Especialistas

Nesta seção será apresentada uma introdução aos Sistemas Especialistas (SE), sua definição, funcionalidade e características.

1.1 Definição

Os Sistemas Especialistas automatizam a solução de problemas que são resolvidos por pessoas que possuem uma especialidade em uma determinada área. Sua definição provém de seu nome, da seguinte forma:

- Sistema: "Conjunto de elementos, materiais ou idéias, entre os quais se possa encontrar ou definir alguma relação" [1].

- Especialista: "Pessoa que se consagra com particular interesse e cuidado a certo estudo. Conhecedor, perito" [1].

Os SE’s tem como objetivo "substituir" um especialista na solução de algum problema. Ao contrário dos Sistemas Convencionais, onde se possui um algoritmo que realiza operações determinadas e retorna uma resposta correta, os SE’s trabalham com problemas para a qual nem sempre se possui uma resposta convencional, sendo assim ele se baseia em buscas heurísticas na tentativa da obtenção de uma solução de um problema.

Exemplo:

Um sistema de diagnóstico da praga e doença do cajueiro, o SECAJU. Este sistema reúne o conhecimento de especialistas em praga e doenças da cultura do caju. O principal objetivo do sistema é diagnosticar e recomendar as melhores soluções para os produtores de caju [2].

1.2 Fundamentação

Os SE’s resolvem problemas específicos, baseando-se em regras, que são elaboradas e fornecidas, regras as quais, possuem uma fundamentação provinda de alguma hipótese, "lei", fórmulas e/ou fatos. Um especialista baseia sua decisão sobre um problema, a partir do conhecimento que ele adquiriu durante sua vida profissional ou

durante seus estudos. Logo, sua maior fonte de conhecimento é a sua memória, onde ele buscará maior parte das respostas.

Durante a análise de um problema, um SE seleciona as premissas que lhe foram impostas e compara com o seu conhecimento, tirando suas conclusões sobre o caso, obtendo assim a melhor solução ou a solução mais correta para o caso. Este processo pode ser um processo complicado e demorado, dependendo muito da aplicação em questão. Durante este processo ele pode deparar com fatos ainda não apresentados e não representados em sua Base de Dados.

Partindo do ponto que um SE deve possuir um mecanismo de aprendizado, ele deve adquirir fatos não apresentados e colocá-los em sua Base de Dados como um novo conhecimento. Este auto-aprendizado, que é uma das idéias mais pregadas na Inteligência Artificial, é também um grande desafio na implementação de SE's.

O SE também pode chegar a uma solução inesperada, porém tal solução é justificada pela base de conhecimento possuída pelo especialista [1].

Os SE's devem se comportar assim como os especialistas, buscando sempre a melhor solução através de seu conhecimento, adquirindo novos conhecimentos quando estes forem impostos, assim melhorando a qualidade de suas soluções posteriores.

1.3 Base de Conhecimento

Para se obter uma solução, um SE precisa possuir uma grande base de conhecimento sobre o assunto abordado pela mesma. Logo seu sucesso depende diretamente do conhecimento que lhe é fornecido e da forma com a qual esse conhecimento será acessado durante a sua execução.

Um SE possui quatro importantes componentes, que são Base de Dados (Base de Conhecimento), Conjunto de Operadores (mecanismos de operação), Estratégia de Controle e Interface com o Usuário, como apresentado na Figura 1.

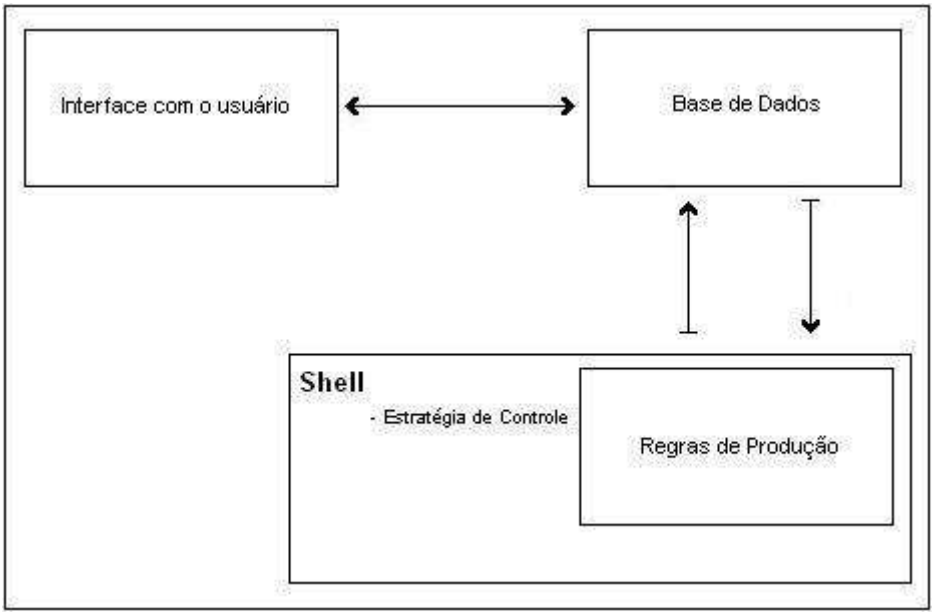

Figura 1. Idéia base de um SE.

Base de Dados é onde ficam alocadas informações essenciais para a execução de um SE. Ela carrega as informações correntes, ou seja, as situações atuais e as metas (conclusões), que são o objetivo dos SE's. Por exemplo, num SE para diagnosticar

crianças com gripe, a situação atual seriam as informações adquiridas através do usuário e a meta seria a conclusão, se a criança está ou não gripada [1].

Conjunto de Operadores são as unidades que vão determinar a execução da Base de Dados. Durante a criação do SE, são criadas regras de produção, essas são as responsáveis pela determinação das teses, conclusões, “leis”, hipótese, entre outros. Mas para que elas sejam criadas, é necessária uma boa representação do conhecimento (Base de Conhecimento) o que irá fazê-las possuírem sentido. Esses operadores podem ser OR, AND e NOT (em maior parte das aplicações de SE, onde temos regras de produção do tipo SE - ENTÃO), o que varia de aplicação para aplicação [1]. O Conjunto de Operadores faz parte das Regras de Produção, como pode ser visto na Figura 1.

Estratégia de Controle é o que determinará como as soluções serão alcançadas, através do método *bottom up* (raciocínio para frente) ou *top-down* (raciocínio para trás). Na *Bottom up* as soluções são obtidas a partir da aplicação de operadores sobre a base de conhecimento produzindo uma situação modificada. Na *top-down* as soluções são obtidas com a aplicação de operadores nas metas, produzindo assim submetas. Tanto *bottom up* quanto *top-down* podem ser usadas combinadas na mesma aplicação, gerando uma variante chamada de Análise Significado-Final ou *means-ends* [1]. Esta estratégia é formulada na *Shell*, pois ela é a responsável pela forma com a qual o SE será executado, como pode ser visto na Figura 1.

1.4 Interface com o Usuário

Todo SE deve ter a sua forma de comunicação com o usuário e por meio da interface. Cada SE possui um tipo de interface, logicamente devido à sua especialidade.

Interface com o usuário é: “em um sistema computacional, conjunto de elementos de hardware e software destinados a possibilitar a interação com o usuário” [3].

A interface representa muito para o usuário, uma vez que sem ela, ele não conseguiria trabalhar. Esta deve ser muito organizada e possuir comentários para auxiliar o usuário durante a utilização do SE.

A interface é a responsável por estabelecer um contato entre o usuário e o SE. Ela recebe as informações fornecidas pelo usuário e as repassa para a Base de Dados, assim como visto na Figura 1.

1.5 *Shells*

A *Shell* é um programa que ajuda a implementar uma aplicação de Inteligência Artificial. A sua construção, no caso de Sistemas Especialistas, visa a implementação de uma "máquina" que seja capaz de interpretar os conhecimentos a ela passada e aplicá-los da forma correta e no tempo correto. A *Shell* é principal parte de um SE, já que o restante é apenas interface e Base de Dados.

A *Shell* é quem vai executar as informações da Base de Dados e do Conjunto de Operadores, por tanto ela deve ser muito confiável, pois a má interpretação de uma regra fornecida como conhecimento, pode culminar em resultados ruins ou indesejados.

Na construção de uma *Shell* é que se define qual será a estratégia de controle, podendo haver qualquer umas da sugeridas (*botton up*, *top-down* ou *means-ends*, anteriormente mencionadas).

2. Expert SINTA

Nesta seção será apresentada uma abordagem geral sobre a ferramenta Expert SINTA.

2.1 Visão Geral

O Expert SINTA é um conjunto de ferramentas computacionais fundamentadas em técnicas de Inteligência Artificial para geração automática de SE's. Utilizando um modelo de representação do conhecimento baseado em regras de produção e fatores de confiança, apresenta como objetivo principal simplificar o trabalho de implementação de SE's através do uso de uma máquina de inferência compartilhada, da construção automática de telas e menus, do tratamento de incerteza nas regras de produção e da utilização de explicações sensíveis ao contexto da base de conhecimento modelada [4].

O Expert SINTA trabalha com o tratamento de incertezas o que é atualmente um grande problema na implementação de SE's. O tratamento de incertezas consiste na aplicação de probabilidade nas regras de inferência, ou seja, aplicar o grau de confiabilidade daquela resposta. Daí vem a dificuldade, pois não podemos nem sempre aplicar probabilidade a problemas do mundo real.

2.2 Interface

O Expert SINTA possui uma interface simplificada, mas mesmo assim exige um pouco de conhecimento do usuário, no quesito elaboração de uma aplicação e principalmente referente ao conhecimento de SE's. A Figura 2 demonstra um exemplo da interface. Ele possui telas bem distribuídas e simplificas, mas durante a elaboração de algum sistema, pode ser confuso, pois exige um "vai e volta" nas telas.

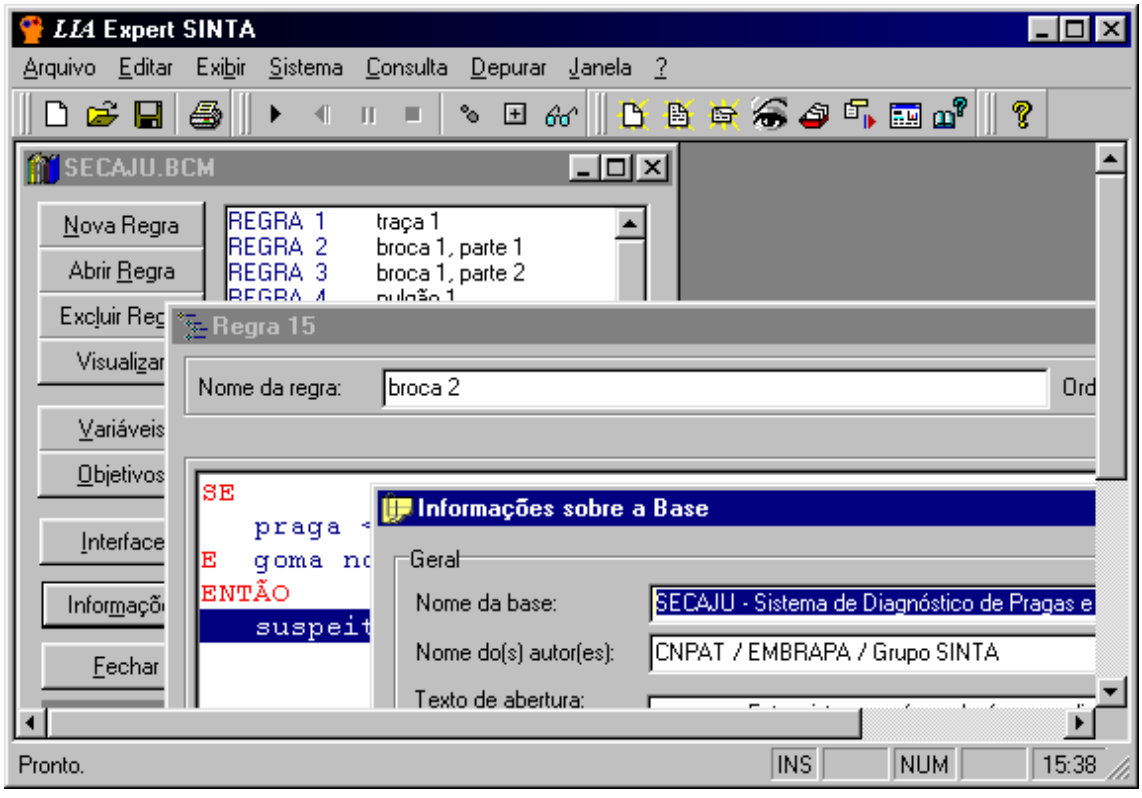

Figura 2. Interface do programa Expert SINTA

O usuário precisa entrar em uma opção para inserir as variáveis e seus valores, em outra para determinar as perguntas e uma para determinar os objetivos do SE.

Durante a estruturação das perguntas e repostas, o usuário tem uma certa dificuldade, pois ele implementa um conjunto de diferentes respostas, mas não possui a explicação que durante a compilação ele terá somente perguntas objetivas, onde suas respostas serão SIM ou NÃO.

2.3 Base de Conhecimento

Para montar a Base de Conhecimento do Expert SINTA, devem ser fornecidos as variáveis e seus valores, definido as variáveis meta e por fim a criação das regras de produção.

A Base é acessada durante a execução do SE e à medida que o usuário insere as informações, elas são alocadas para análise.

Esta Base de Conhecimento é elaborada durante a criação das regras de produção. Uma vez que as variáveis e seus valores são inclusos no sistema, cria-se então a partir deles regras de produção. Estas regras serão a Base de Conhecimento, elas é que irão determinar a partir das entradas do usuário, qual será a melhor solução ou solução correta da aplicação. A Base de Conhecimento nos SE's, em geral, deve ser muito bem elaborada, no Expert SINTA para toda regra criada, deve se criar outras que atendam a todo o conjunto solução possível, pois o sistema pode chegar a soluções indesejadas ou não conhecidas pelo sistema. O Expert SINTA não implementa um mecanismo de auto-aprendizado, para evitar este tipo de solução indesejada.

3. Implementação do SEG – Gerador de Sistemas Especialistas

Visto que o Expert SINTA é um programa que exige um certo grau de conhecimento por parte do usuário, foi construído um programa que permite a criação de SE's de forma mais simplificada. As modificações mais visíveis são a interface e a forma de entrada dos dados.

A forma com a qual o sistema irá interagir com o usuário, será a mesma, através de perguntas objetivas, onde serão feitas as perguntas até que algum objetivo seja alcançado.

O SEG foi implementado na linguagem de programação Delphi, versão 7, que é uma linguagem orientados a objetos. Por tanto, o mesmo roda em ambiente Windows.

Seu desenvolvimento utilizou os conceitos de classes encontrados na orientação a objeto. Utilizou-se também, o ponto forte do Delphi, que é a parte visual, aproveitando-se dos vários recursos já existentes neste. Os detalhes da implementação serão vistos a seguir, cada um em seu tópico.

3.1 Funcionalidade

O SEG irá funcionar de forma diferente ao Expert SINTA. As informações serão fornecidas a ele em formato de texto, como será explicado mais adiante.

Uma vez que este texto é fornecido, o sistema irá efetuar a verificação do mesmo, para tentar localizar inconsistências, como erro de sintaxe e/ou semântica. Após esta

verificação o sistema liberará a base para ser executada.

A execução se dará da seguinte forma. Serão feitas perguntas objetivas ao usuário, suas respostas serão armazenadas na Base de Conhecimento e posteriormente combinadas com as regras obtendo assim os resultados. A qualquer momento, quando não estiver sendo executado, o usuário poderá inserir novas regras para complementar o seu SE, estas serão testadas para verificação de inconsistências.

3.2 Base de Conhecimento

Como todo SE, o SEG possuíra a sua própria Base de Conhecimento.

No SEG a Base de Conhecimento é fornecida de forma diferente, visto que, por exemplo, no Expert SINTA ela é fornecida de forma distribuída, esta é passada durante a formulação das regras. No SEG a forma de entrada de variáveis, valores de variáveis e regras de produção, será feita de uma vez só. Estas entradas serão feitas da forma apresentada a seguir.

Formulam-se as regras, com base em alguma hipótese, "lei", fórmulas e/ou fatos como visto anteriormente. Essas regras são passadas ao sistema em formato de texto. Este texto terá o formato SE - ENTÃO, como é apresentado a seguir na Regra Geral e no exemplo.

Regra Geral:

```
<Nome da regra>:
SE <variável> É <valor lógico>
[<operando> <variável> É <valor lógico> ...]
ENTÃO <variável meta> É <valor lógico>.
```

Exemplo:

```
Regra 1:
SE amanhã pode chover é falso
E tenho dinheiro suficiente é verdadeiro
E tenho tempo suficiente é verdadeiro
ENTÃO devo fazer churrasco amanha é verdadeiro.

Regra 2:
SE amanhã pode chover é verdadeiro
OU tenho dinheiro suficiente é falso
OU tenho tempo suficiente é falso
ENTÃO devo fazer churrasco amanhã é falso.

Regra 3:
SE metereologia diz que pode chover é verdadeiro
ENTÃO amanha pode chover é verdadeiro.

Regra 4:
SE não vou sair hoje é verdadeiro
E ocorreu emergência é falso
ENTÃO tenho dinheiro suficiente é verdadeiro.

Regra 5:
SE minha namorada ligou é verdadeiro
ENTÃO não vou sair hoje é falso.

Regra 6:
SE meu professor marcou prova é verdadeiro
```

```
ENTÃO tenho tempo suficiente é falso.

Regra 7:
SE metereologia diz que pode chover é falso
ENTÃO amanhã pode chover é falso.

Regra 8:
SE não vou sair hoje é falso
E ocorreu emergência é verdadeiro
ENTÃO tenho dinheiro suficiente é falso.

Regra 9:
SE minha namorada ligou é falso
ENTÃO não vou sair hoje é verdadeiro.

Regra 10:
SE meu professor marcou prova é falso
ENTÃO tenho tempo suficiente é verdadeiro.
```

Certos caracteres e palavras reservadas são importantes durante a elaboração das regras de produção, é muito importante que os mesmos sejam utilizados nos locais corretos. São estes os caracteres:

. (ponto) – determina o fim de uma regra.

: (dois pontos) – determinar o fim do nome da regra.

SE – determina o início da regra.

ENTÃO – determina a conclusão da regra.

E – operando lógico.

OU – operando lógico.

É – determinar o valor da variável.

Logo, é importante que as palavras reservadas não sejam utilizadas na elaboração de variáveis e os caracteres devem ser utilizados apenas para as suas finalidades, pois o filtro de reconhecimento de texto os enxerga como “itens” reservados para a *Shell*.

A Base de Conhecimento será elaborada da seguinte forma, em termos de execução do sistema:

Cada regra possui um conjunto de premissas (variáveis), como foi visto no exemplo anterior, estas serão colocadas separadamente em uma lista (vetor). Cada variável possuí um conjunto de informações, como nome, o nome da regra a qual ela defini um estado meta, valores e uma pergunta (depende da aplicação). Estas informações são vitais para a Base de Conhecimento. A Figura 3 demonstra a elaboração desta estrutura.

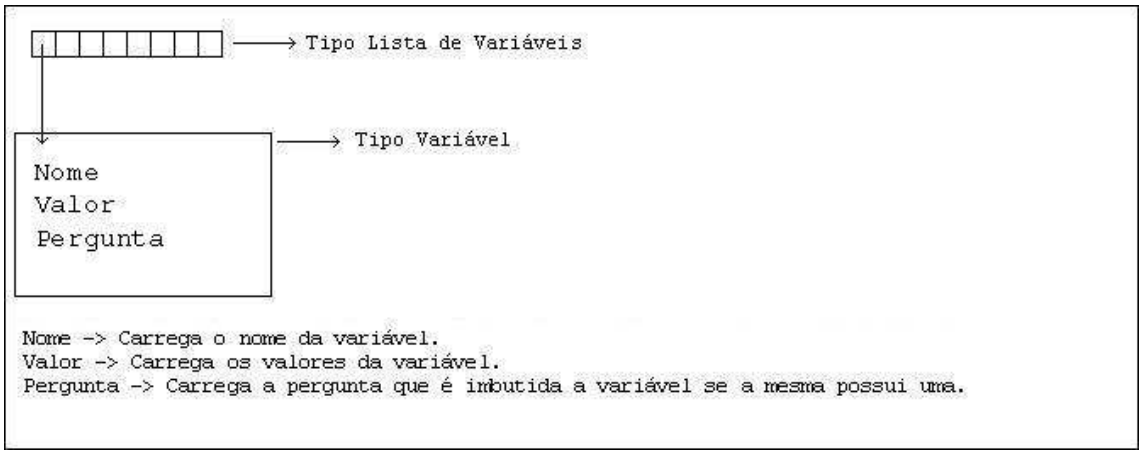

FIGURA 3. Estrutura das variáveis.

As regras por sua vez também possuem uma estrutura. Esta estrutura é uma lista (vetor), assim como as variáveis. O conjunto de informações das regras são um cabeçalho, que contém o nome da regra e o nome da variável meta da mesma, o corpo, que é uma lista contendo as suas premissas distribuídas entre as posições da mesma, como pode ser visto na Figura 4.

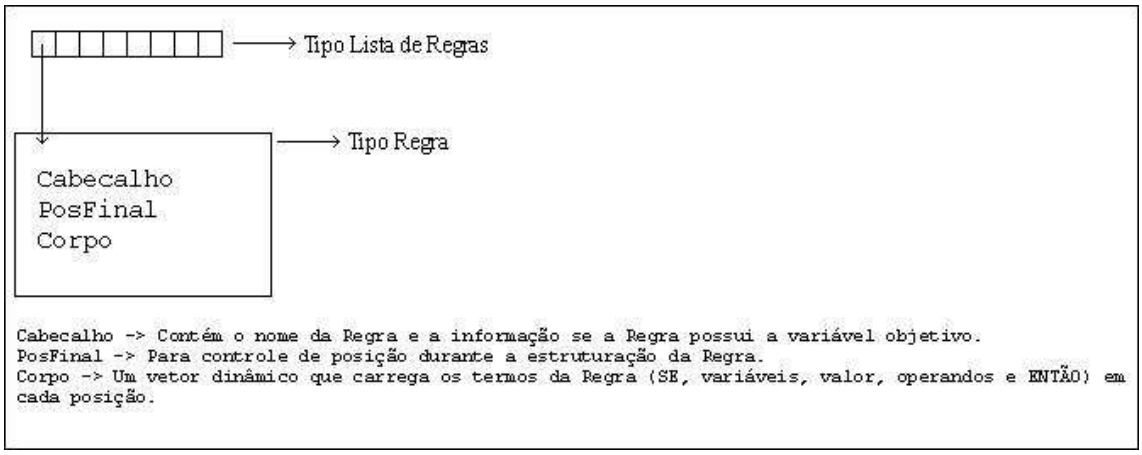

FIGURA 4. Estrutura da lista de regras.

3.3. Interface com o Usuário

A interface do SEG será dispostas em telas simplificadas, divididas em “módulos”. Ele possuirá uma tela principal, que dará acesso a estes módulos, estes serão as telas de inserção de texto e propriedades das variáveis. Durante a execução, serão exibidos “módulos” com as perguntas, chegando a conclusão, o mesmo será exibido da mesma forma, como visto na Figura 5 e na Figura 6.

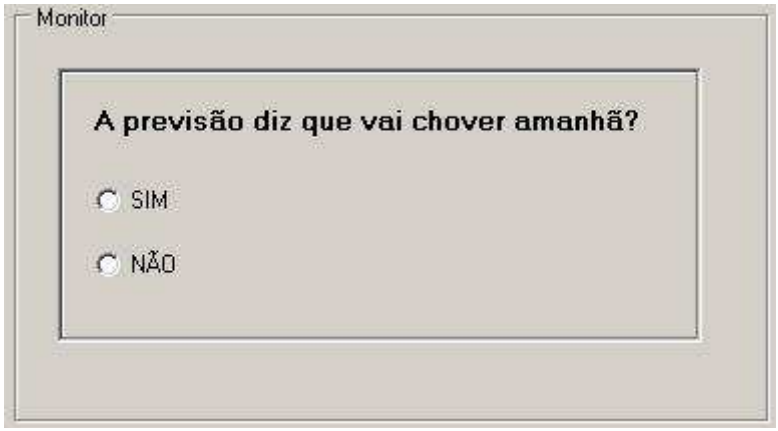

Figura 5. Módulo de pergunta.

O usuário irá primeiramente inserir um texto contendo regras de produção em um formato pré-estabelecido. Esta entrada de texto será feita em uma tela contendo uma caixa de texto. Logo após a inserção do texto, o usuário irá executar as regras para correção e verificação de erros.

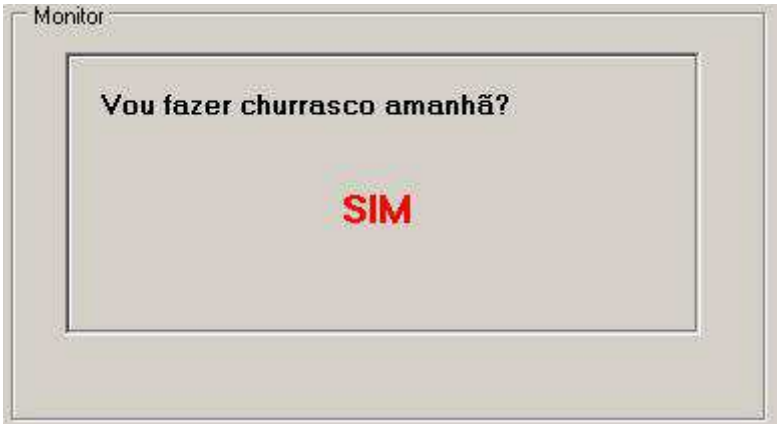

Figura 6. Exemplo do módulo de conclusão.

Com o texto previamente testado e aprovado, o usuário terá de abrir a tela onde ele fará a atribuição de perguntas as variáveis. Esta tela possui a lista de todas as variáveis. O usuário terá a opção de escolher a variável e posteriormente inserir o seu valor. As variáveis objetivo serão selecionadas na mesma tela, onde o usuário irá selecionar o tipo de variável (normal ou objetivo).

Durante a execução serão apresentadas pequenas telas, contendo a pergunta e as opções de resposta. Quando o objetivo for atingido, será exibida a resposta na mesma tela.

3.4. *Shell*

O principal componente do SEG, a *Shell*, é implementada de forma bem simplificada. A *Shell* do SEG não apresenta o mecanismo de probabilidade, como no Expert SINTA. Sua estratégia de controle é a *bottom up*, ou seja, o raciocínio é feito a partir da meta, produzindo submetas.

A execução se dará da seguinte forma:

Primeiramente a *Shell* localizará entre as regras, qual é a regra meta ou quais as regras metas. Caso sejam encontradas mais de uma regra meta, ela parte da primeira

localizada, não encontrando uma solução satisfatória, ele pega outra regra meta e reinicia a execução.

Partindo de uma regra meta, ele inicia a execução a partir da palavra reservada SE, passando a primeira variável, ele verifica se existe alguma pergunta imposta àquela variável. Caso tenha, a pergunta é feita ao usuário se não possui pergunta, ele localiza uma regra aonde esta variável é uma meta e chama recursivamente um método interno, da *Shell*, que realiza a execução, como pode ser visto na Figura 7. Para cada valor retornado, ele é armazenado para análise da solução posteriormente. Quando todas soluções forem adquiridas e não houver mais premissas a serem analisadas, a *Shell* vai checar se existe alguma regra que satisfaça o conjunto solução obtido, gerando assim a resposta para o usuário, caso não haja nenhum conjunto solução, será dada uma resposta negativa ao usuário.

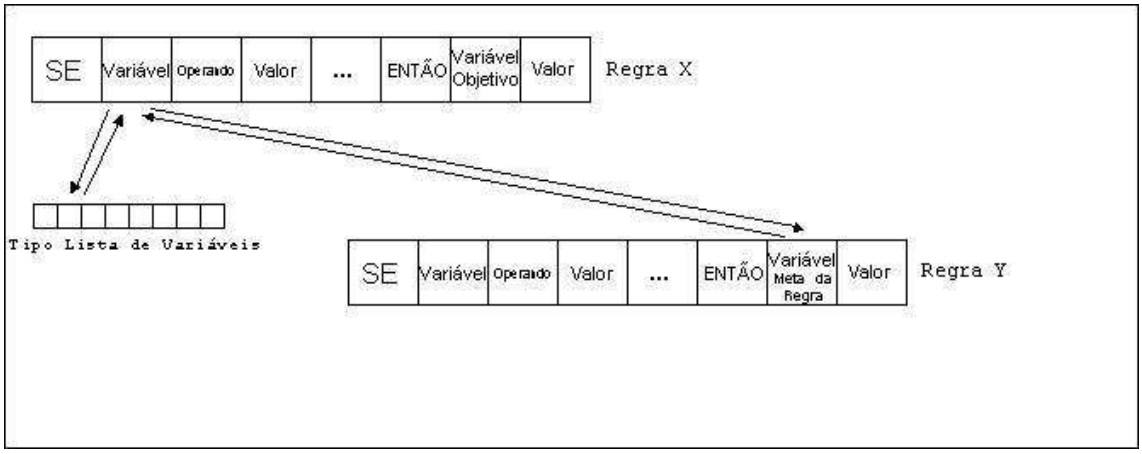

Figura 7. Durante a execução, a *Shell* verifica na lista de variáveis se a variável em questão possui uma pergunta atribuída a ela. Caso não haja, ele localiza a regra que possui a variável como meta e recursivamente reinicia a execução a partir da nova regra.

4. Considerações Finais

O estudo da disciplina de Inteligência Artificial contém vários ramos, um deles é o que aborda os Sistemas Especialistas. O desenvolvimento de trabalhos nesta área contribui com uma enorme parcela para o desenvolvimento da Inteligência Artificial e a combinação destes trabalhos com outros trabalhos em desenvolvimento e já desenvolvidos enriquecem mais ainda o campo da Inteligência Artificial.

O que foi demonstrado neste artigo, é que existe uma nova forma de construção de Sistemas Especialistas e que quanto mais novas formas forem exploradas, haverá um enorme crescimento desta área da Inteligência Artificial.

Os Sistemas Especialistas propõem uma grande evolução na Inteligência Artificial, uma vez que, eles podem ser aplicados as mais diversas áreas, principalmente na Medicina. Além de poderem ser combinados com as diversas outras subdivisões da Inteligência Artificial, vindo a enriquecer mais a área da Inteligência Artificial.

Como trabalhos futuros, proponho a inserção de mecanismos de auto-aprendizado, o que representa atualmente um grande problema na Inteligência Artificial, a elaboração da probabilidade, mesmo que não seja totalmente indicado e a correção de possíveis falhas que o novo sistema possa demonstrar e que não tenham sido vistos,

5. Referências Bibliográficas

1. FÁVERO, Alexandre José; SANTOS, Nilson Moutinho dos; - Disponível em <http://www.din.uem.br/ia/especialistas/> - acesso 16 de Maio de 2006.

2. SECAJU - Sistema de Diagnóstico de Pragas e Doenças do Cajueiro – Aplicativo para o Expert SINTA.

3. FERREIRA, Aurélio Buarque de Holanda – Novo Aurélio Século XXI: o dicionário da língua portuguesa, 3ed. , Nova Fronteira, Rio de Janeiro, 1999.

4. BEZZERA, Ricardo de Andrade e Silva; SILVESTRE, Ricardo Sousa; NOGUEIRA, José Helano Matos; ALCÂNTARA, João Fernando Lima; PEQUENO, Tarcísio Cavalcante; CASTRO. Rafael de Andrade; CORREIA, Shirlene de Holanda - Expert SINTA - Disponível em <http://www.lia.ufc.br/~bezerra/exsinta/> - acesso em 26 de Nov de 2005.