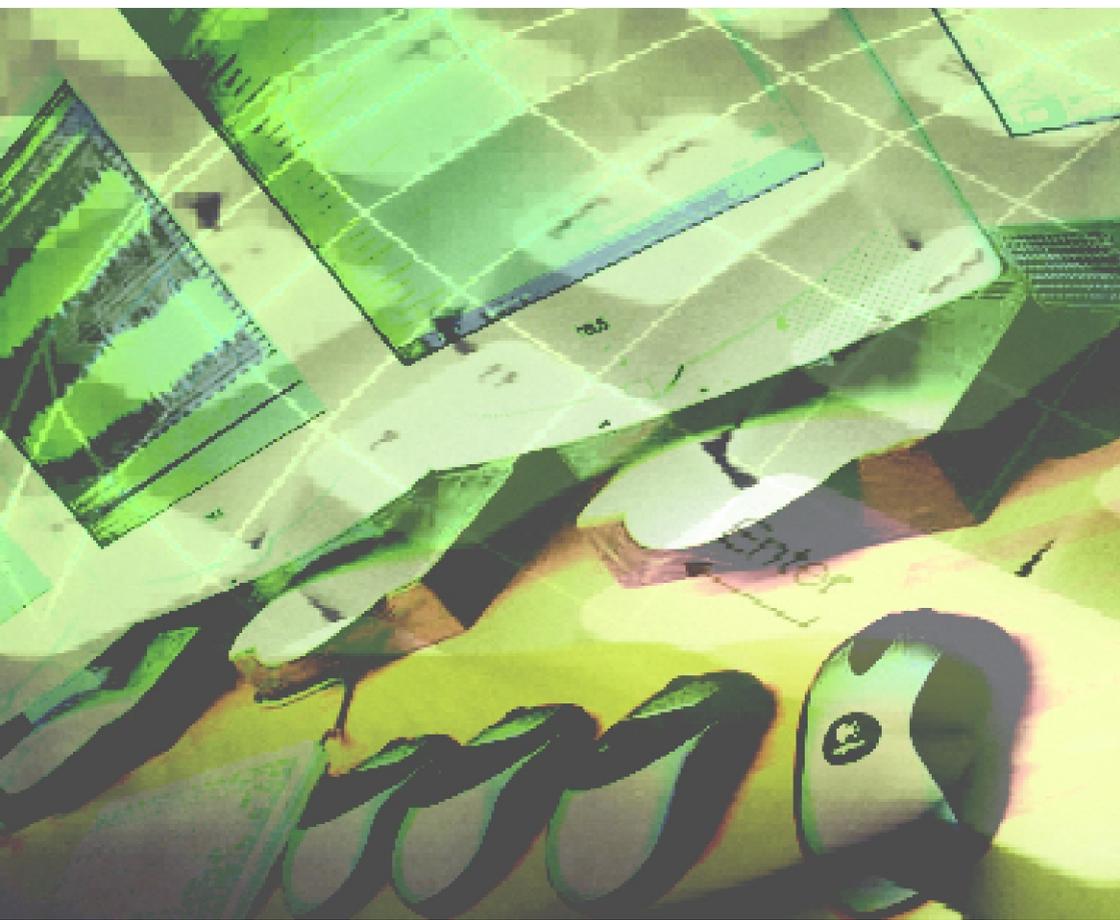


Boletim de Pesquisa 15 e Desenvolvimento *Abril, 2007*

ISSN 1677-9266

Uma Abordagem Orientada a Objetos para Ambiente de Diagnóstico Remoto





*Empresa Brasileira de Pesquisa Agropecuária
Embrapa Informática Agropecuária
Ministério da Agricultura, Pecuária e Abastecimento*

ISSN 1677-9266
Abril, 2007

Boletim de Pesquisa e Desenvolvimento 15

Uma Abordagem Orientada a Objetos para Ambiente de Diagnóstico Remoto

Silvia Maria Fonseca Silveira Massruhá
Juliano Pastorelli Dutra
Sérgio Aparecido Braga Cruz

Campinas, SP
2007

Embrapa Informática Agropecuária
Área de Comunicação e Negócios (ACN)

Av. André Tosello, 209
Cidade Universitária "Zeferino Vaz" Barão Geraldo
Caixa Postal 6041
13083-970 - Campinas, SP
Telefone (19) 3789-5743 Fax (19) 3289-9594
URL: <http://www.cnptia.embrapa.br>
e-mail: sac@cnptia.embrapa.br

Comitê de Publicações

Adriana Farah Gonzalez (secretária)
Ivanilde Dispatto
José Iguelmar Miranda
Kleber Xavier Sampaio de Souza (presidente)
Marcia Izabel Fugisawa Souza
Silvio Roberto Medeiros Evangelista
Stanley Robson de Medeiros Oliveira

Suplentes

Laurimar Gonçalves Vendrusculo
Maria Goretti Gurgel Praxedes

Supervisor editorial: *Ivanilde Dispatto*
Normalização bibliográfica: *Marcia Izabel Fugisawa Souza*
Editoração eletrônica: *Área de Comunicação e Negócios (ACN)*

1ª. edição on-line - 2007
Todos os direitos reservados.

Massruhá, Silvia Maria Fonseca Silveira.

Uma abordagem orientada a objetos para ambiente de diagnóstico remoto / Silvia Maria Fonseca Silveira Massruhá, Juliano Pastorelli Dutra, Sérgio Aparecido Braga Cruz. —
Campinas : Embrapa Informática Agropecuária, 2007.

33 p. : il. — (Boletim de Pesquisa e Desenvolvimento / Embrapa Informática Agropecuária; 15).

ISSN 1677-9266

1. Inteligência artificial. 2. Inferência abduativa. 3. Lógica nebulosa. 4. Orientação a objeto. 5. Sistema especialista. 6. Representação do conhecimento. 7. Doença de planta. I. Dutra, Juliano Pastorelli. II. Cruz, Sérgio Aparecido Braga. III. Título. IV. Série.

CDD - 006.3 (21st. ed.)

Sumário

Resumo.....	5
Abstract.....	6
Introdução.....	7
Material e Métodos.....	9
Resultados e Discussão.....	26
Conclusões.....	28
Referências Bibliográficas.....	30

Uma Abordagem Orientada a Objetos para Ambiente de Diagnóstico Remoto

Silvia Maria Fonseca Silveira Massruhá¹

Juliano Pastorelli Dutra²

Sérgio Aparecido Braga Cruz³

Resumo

No escopo deste trabalho é apresentada uma infra-estrutura orientada a objetos, desenvolvida no âmbito do projeto proposto por Massruhá (2004), que suporta via web a abordagem integrada para diagnóstico, investigação e tratamento de desordens proposta em Massruhá (2003). Nesta nova abordagem, o conhecimento é basicamente modelado através de associações causais entre objetos e a inferência é abduativa. Conceitos de Parsimonious Covering Theory (PCT), lógica nebulosa e teoria de decisão são também integrados, de maneira a tratar os vários aspectos inerentes aos processos envolvidos em raciocínio clínico. A validação do modelo teórico foi realizada em fitopatologia, abordando especificamente a diagnose, investigação e tratamento de doenças de milho. Entretanto, devido a sua generalidade, espera-se que os desenvolvimentos teóricos obtidos possam ser aplicados para resolver problemas de diagnóstico tanto na área vegetal quanto na área animal.

Termos para indexação: inferência abduativa, lógica nebulosa, orientação a objetos, sistemas especialistas, representação do conhecimento, doenças de plantas.

¹ Doutora em Computação Aplicada, Pesquisadora da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP. (e-mail: silvia@cnptia.embrapa.br)

² Bacharel em Ciência da Computação, Consultor da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP. (e-mail: juliano@cnptia.embrapa.br)

³ Mestre em Engenharia Elétrica, Pesquisador da Embrapa Informática Agropecuária, Caixa Postal 6041, Barão Geraldo - 13083-970 - Campinas, SP. (e-mail: sergio@cnptia.embrapa.br)

An Object-oriented Approach for Remote Diagnosis

Abstract

This work presents an object-oriented approach developed in Massruhá (2004) to support by web an integrated framework for diagnosis, investigation and treatment of disorders (diseases, computer faults, etc.) proposed in Massruhá (2003). In this new framework, the knowledge is basically modeled through causal associations between objects and inference is abductive. Concepts of Parsimonious Covering Theory (PCT), fuzzy sets logic and decision theory are also integrated, in order to address the various inherent aspects involved in clinical reasoning. The validation of the theoretical model has been performed in phytopathology, specifically, at the diagnosis, investigation and treatment of the corn plants diseases. However, due to its generality, it is expected that the theoretical developments obtained in this work can successfully employed to solve diagnosis problems in animal and vegetal area.

Index terms: abductive inference, fuzzy sets, object-oriented, expert systems, knowledge representation, plants diseases.

Introdução

No acervo bibliográfico da Embrapa pode-se encontrar vários documentos sobre "Diagnóstico de Doenças e Métodos de Controle", tanto na área animal como vegetal. Todo este conhecimento, entretanto, se apresenta disperso em documentos, sendo que com o tempo o problema se agrava, com os registros se acumulando e tornando ainda mais difícil a manipulação, bem como a recuperação do conhecimento embutido nos mesmos.

Em Tecnologia da Informação (TI), mais especificamente na área de Inteligência Artificial (IA), os sistemas baseados em conhecimento (SBC) são apresentados como uma abordagem eficiente para assegurar que o conhecimento permaneça na empresa após a saída de especialistas e seja utilizada amplamente, sem as barreiras encontradas atualmente.

A comunidade de IA tem atribuído algumas características específicas aos sistemas baseados em conhecimento que o diferencia de um sistema convencional. Neste contexto, os SBCs tem as seguintes propriedades:

- uma base de conhecimento que tem representado tudo que se sabe sobre o problema;
- um mecanismo de inferência capaz de interpretar e raciocinar sobre a base de conhecimento;
- utiliza-se de busca heurística a partir do conhecimento do domínio para resolver os problemas os quais não se conhece um procedimento determinístico para resolvê-lo.

É importante também diferenciar SBCs de sistemas especialistas (SEs). Enquanto SBCs são sistemas capazes de resolver problemas usando conhecimento específico de um domínio de aplicação, os SEs são SBCs que resolvem problemas de um especialista humano. Os SEs podem ser classificados como SBCs que por sua vez são classificados como Sistemas Inteligentes (SIs). No escopo deste trabalho, os termos sistemas inteligentes, sistemas baseados em conhecimento e sistemas especialistas serão utilizados indistintamente.

Sistemas baseados em conhecimento, aqui englobando os sistemas especialistas, têm sido utilizados nos mais variados ramos de atividade como negócios, medicina, ciência e engenharia. A utilização de sistemas especialistas em geral na agricultura também já vem sendo explorada em vários trabalhos (Durkin, 1994; Huber & Doluschitz, 1990; Yialouris & Sideridis, 1996; Mahaman et. al., 2002, 2003). Na década de 1980 foram desenvolvidos vários sistemas especialistas em IA para automatizar a etapa de diagnóstico (Durkin, 1994).

Na literatura podem ser encontrados vários documentos sobre diagnóstico de doenças e métodos de controle tanto na área animal quanto na área vegetal passíveis de serem incorporados em sistemas especialistas (Fernandes & Oliveira, 1997; Lopes e Ávila, 2003, 2005). Em fitopatologia foram desenvolvidos vários sistemas para a diagnose e manejo de doenças, como por exemplo, o Plant/ds, para diagnose de doenças da soja (Michalski et al., 1982); MORECROP - Managerial Options for Reasonable Economical Control of Rust and Other Pathogens (http://pnw_ag.wsu.edu/MoreCrop), para previsão e manejo da ferrugem e outras doenças na cultura do trigo nos EUA; além de outros para doenças do abacaxizeiro, pêras e nectarinas e batata. No Brasil, destacam-se o TomEx, para diagnóstico de doenças do tomateiro e um sistema especialista para diagnóstico de doenças do cafeeiro (Pozza et al., 1997).

Uma das características em comum destes sistemas é a de se comparar um modelo estabelecido para cada desordem com os dados do caso particular. Usualmente, estes sistemas são baseados em regras, onde as manifestações aparecem nas premissas e as causas nas conclusões (i.e. *efeitos* → *causas*). Em conseqüência, o raciocínio adotado para a inferência nestes sistemas é do tipo dedutivo; a partir das regras (*efeitos* → *causas*) e fatos (*efeitos*), deduzem-se novas informações (*causas*). O problema principal com este tipo de abordagem é que o conhecimento do tipo *efeitos* → *causas* é contrário à maneira com que os fenômenos ocorrem na natureza (*causas* → *efeitos*).

Na década de 1990, o conhecimento na forma *causas* → *efeitos* passou a ser modelado, com o uso de raciocínio do tipo abduativo; a partir das associações (*causas* → *efeitos*) e fatos (*efeitos*), criam-se novas hipóteses (*causas*). A Teoria das Coberturas Parcimoniosas - TCP (Peng & Reggia, 1990), representou uma primeira tentativa para formalizar o raciocínio causal abduativo de diagnóstico. A modelagem de incerteza usando informação probabilística na relação causal possibilitou reduzir o tamanho das soluções do problema de diagnóstico, além de ordená-las.

Desde o advento da TCP, vários autores têm investigado teorias da diagnose alternativas, ou extensões/modificações da TCP (Lucas, 1996; Dubois & Prade, 1995; Wainer & Rezende, 1997; Wainer & Sandri, 1998). Entretanto, todos estes trabalhos apresentam soluções parciais na área de diagnóstico e não uma solução completa para o problema de raciocínio clínico. Adicionalmente, as soluções em IA para as tarefas de investigação e tratamentos ainda são muito tímidas (Tunez et al., 1998; Taboada et al., 1999). As tecnologias oferecidas em TD (Henrion et al., 1992), por sua vez, propõem soluções para as tarefas de tomada de decisão nas áreas de investigação e tratamentos. Cada vez mais se nota a necessidade de integração das tecnologias desenvolvidas nas áreas de IA e TD, embora

sejam baseadas em princípios diferentes.

Em Massruhá (2003) foi apresentada uma abordagem integrada para diagnóstico, investigação e tratamento de desordens (Massruha et. al, 2003, 2004, 2005a, 2005b). Nesta nova abordagem, denominada Teoria das Coberturas Nebulosas (TCN), o conhecimento é basicamente modelado através de associações causais e a inferência é abduativa (desordens ->manifestações). Conceitos de TCP, lógica nebulosa e teoria de decisão são também integrados, de maneira a tratar os vários aspectos inerentes aos processos envolvidos em raciocínio clínico. A TCN foi validada para diagnóstico, investigação e tratamentos de doenças de milho.

No âmbito do projeto proposto por Massruhá (2004) está sendo desenvolvida uma infra-estrutura orientada à objetos que suporta, via web, a abordagem integrada para diagnóstico, investigação e tratamento de desordens proposta em Massruhá (2003). No contexto deste trabalho, está descrita a metodologia bem como os resultados obtidos para portar o modelo acadêmico de diagnóstico, investigação e tratamento desenvolvido em Prolog, para uma nova arquitetura baseada em tecnologia J2EE - Java 2 Platform, Enterprise Edition (Sun Microsystems, 2005). A base de conhecimento de doenças de milho foi utilizada como estudo de caso do presente trabalho. Entretanto, o objetivo final deste trabalho de pesquisa é compará-lo com as outras arquiteturas desenvolvidas na Embrapa Informática Agropecuária visando validar a flexibilidade e a facilidade de expandir o *framework* proposto para outras bases de conhecimento.

Material e Métodos

Nesta seção está descrita a arquitetura orientada à objetos proposta para o desenvolvimento de sistemas especialistas via web, bem como o material e os métodos utilizados neste trabalho. Para fins didático, antes de detalhar a arquitetura proposta neste trabalho, alguns conceitos básicos de SE são apresentados e, na seqüência é apresentada uma breve revisão das abordagens anteriores ao presente trabalho adotadas no âmbito da Embrapa Informática Agropecuária.

Simplificadamente, conforme apresentado na Fig. 1, um SE é composto por três módulos principais: **mecanismo de inferência**, **a base de conhecimento** e **a memória de trabalho**. O mecanismo de inferência é o método de aplicação do conhecimento utilizado pelo diagnosticador na busca da solução do problema. A base de conhecimento é onde está representado todo o conhecimento de um domínio. A memória de trabalho é a parte que contém os fatos sobre o problema que são descobertos, concluídos, em uma sessão de execução. A interface é um módulo opcional

que permite a interação entre o SE e o usuário proporcionando a comunicação entre eles. Na Fig. 1 também estão representados os personagens de um SE. O especialista é o que fornece e testa o conhecimento. O valor absoluto do especialista é o conhecimento como ilustrado na Fig. 1. O Engenheiro do conhecimento é que constrói o SE e o usuário é quem efetivamente utiliza o SE entrando com os fatos e obtendo as conclusões.

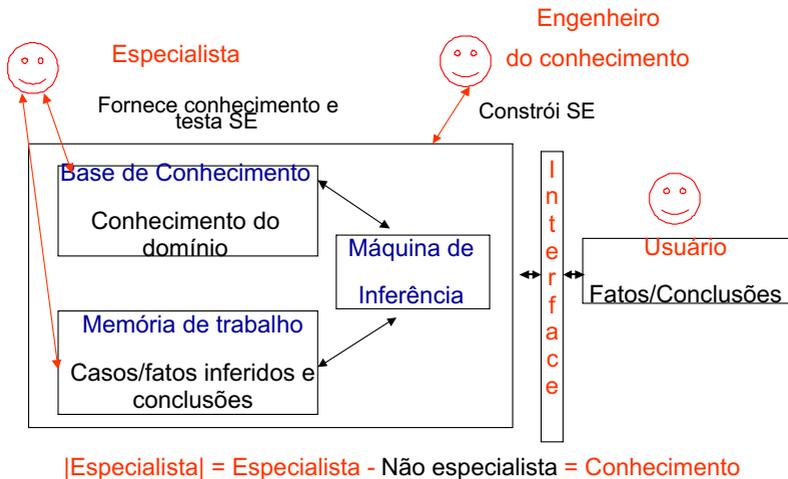


Fig. 1. Estrutura geral de um SE e seus personagens.

Um passo importante na construção de um SE é a identificação das características do domínio de conhecimento. A descrição do conhecimento para a resolução de um determinado problema deve estar contemplada na base de conhecimento. Segundo Rezende (2003), isto inclui asserções sobre o domínio de conhecimento, regras que descrevem relações neste domínio e, em alguns casos, heurísticas e métodos de resolução de problemas. Informalmente, uma BC é um conjunto de representações de ações e acontecimentos do mundo. Cada representação individual é chamada de sentença. As sentenças são expressas em uma linguagem específica, linguagem de representação do conhecimento. Estas linguagens se baseiam em diferentes técnicas de representação do conhecimento (Giarratano & Riley, 1994).

As principais técnicas de representação, que serão discutidas neste trabalho, são: **lógica, regras de produção, redes semânticas, frames e orientação a objetos.**

Desde o final dos anos 70 houve um grande interesse no uso de lógica matemática na pesquisa em IA em contraste a métodos mais intuitivos e heurísticos. A lógica matemática é uma linguagem formal onde uma seqüência de símbolos está de acordo com as regras para a construção de expressões (fórmulas) da linguagem. A lógica matemática possui regras para realizar inferências dedutivas a partir do formato sintático das expressões de linguagem denominado dedução automática. Existem vários tipos de lógicas para dedução automática. O cálculo proposicional é a mais simples delas porque se baseia apenas na existência de constantes e no uso de operadores lógicos. Observe o exemplo a seguir.

$$\begin{aligned} \text{Humano}(x) &\rightarrow \text{Mortal}(x) \\ \text{Mulher}(y) &\rightarrow \text{Humano}(y) \end{aligned}$$

Conforme descrito em Giarratano & Riley (1994), embora o cálculo proposicional seja muito útil para a representação do conhecimento, apresenta algumas limitações. Dentre elas, não se consegue expressar algumas declarações mais genéricas. Por exemplo:

$$\begin{aligned} &\text{Todos humanos são mortais} \\ &\underline{\text{Todas mulheres são humanos}} \\ \therefore &\text{Todas mulheres são mortais.} \end{aligned}$$

Para tal, é necessário usar o cálculo de predicados (ou lógica de primeira ordem). Nesse cálculo, as proposições são formadas por predicados, argumentos, variáveis e quantificadores (ex. \forall - quantificador para todo). A lógica de primeira ordem tem um potencial expressivo e tem sido o instrumento preferido para a formalização do conhecimento durante o processo de desenvolvimento de um SE. O exemplo acima pode ser representado na lógica de predicados como descrito a seguir.

$$\begin{aligned} \forall(x)(H(x) \rightarrow M(x)) \\ \forall(y)(Mu(y) \rightarrow H(y)) \end{aligned}$$

Uma sentença em lógica de predicados ($p \rightarrow q$) é facilmente traduzida para o formato de regras de produção. As regras de produção são estruturas do tipo:

SE <condições> ENTÃO <conclusões> FAÇA <ações>

nas quais a parte SE é uma lista de condições a serem satisfeitas. Cada uma das condições da lista é verificada, e se todas forem satisfeitas, as conclusões são consideradas verdadeiras e as ações são executadas. Em programação baseada em regras, utiliza-se de um motor de inferência para manipular as regras e os fatos até chegar a um resultado final. A seguir é apresentado um exemplo desta técnica.

SE *x* é humano ENTÃO *x* é mortal
 SE *y* é mulher ENTÃO *y* é humano
 ∴ SE *y* é humano ENTÃO *y* é mortal

A rede semântica é uma outra técnica de representação do conhecimento clássica em IA. A rede semântica é um grafo rotulado e direcionado formado por um conjunto de nós representando os objetos e por um conjunto de arcos representando as relações entre os nós (Giarratano & Riley, 1994). O exemplo apresentado acima pode ser mapeado para redes semânticas conforme apresentado na Fig. 2

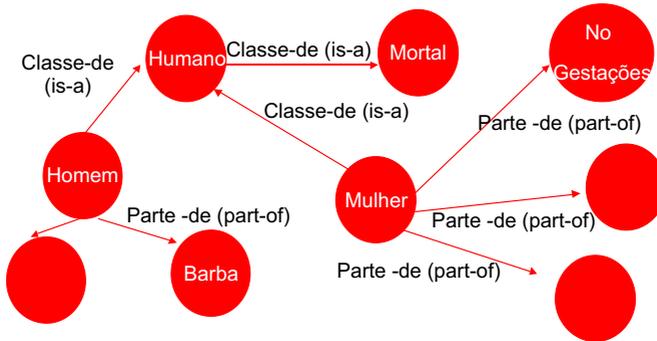


Fig. 2. Exemplo de rede semântica.

Um outro tipo de esquema utilizado para representação do conhecimento em IA é o frame. O frame possui um nome que identifica o conceito por ele definido e consiste de um conjunto de atributos, chamados slots. Os frames são análogos a registros de base de dados, entretanto, uma das características de frames é a herança de propriedades entre os frames (Giarratano & Riley, 1994). Um frame-filho pode herdar os atributos dos frames-pai e assim sucessivamente (Fig. 3).

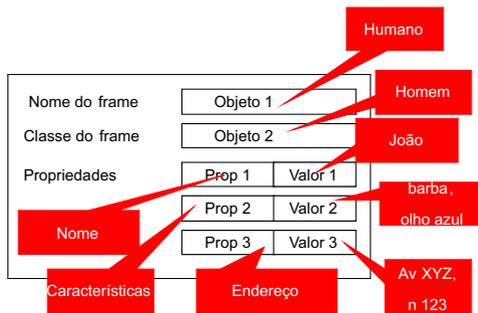


Fig. 3. Exemplo de frame.

Atualmente, adicionalmente as regras, redes semânticas e frames tem-se o paradigma de representação Orientada a Objetos (Rumbaugh, 1999). A orientação a objetos reúne características tanto das redes semânticas quanto dos frames, conforme apresentado na Tabela 1.

Tabela 1. Analogia entre OO, redes semânticas e frames.

<i>Objetos</i>	<i>Frames</i>	<i>Redes semânticas</i>
Atributos	Slots	
Métodos	Procedimentos	
Comunicação através de mensagens	Comunicação através de chamada de procedimentos	
Estrutura hierárquica (classes e subclasses)	Classes e subclasses	Rede semântica (classe-de /is-a)

A orientação a objetos oferece uma representação estrutural de relações estáticas, mas não oferece facilidades diretas para descrever declarativamente como o conhecimento armazenado deve ser utilizado. Esta deficiência pode ser suprida pela integração da representação orientada a objetos com regras de produção que oferecem uma maneira simples de expressar o raciocínio do sistema.

Dentre as linguagens para representação do conhecimento podem-se destacar, CLIPS e Prolog, que serão discutidas neste trabalho. O CLIPS que é um shell para desenvolvimento de sistemas especialistas onde a representação de conhecimento é orientação a objetos associada com regras de produção (Giarratano & Riley, 1994). O Prolog é a linguagem mais usada para desenvolvimento de sistemas especialistas que utiliza os princípios da lógica matemática.

Agora que os principais mecanismos de representação de conhecimentos já foram descritos, os principais mecanismos de inferência que podem ser utilizados para derivar nova informação a partir da base de conhecimento podem ser apresentados. Conforme apresentado em Giarratano & Riley (1994), os principais mecanismos de inferência são: **dedução, indução e abdução**.

A dedução é o mecanismo de inferência comumente utilizado pelos seres humanos para deduzir nova informação a partir de informação relacionada logicamente. Na inferência dedutiva, um resultado específico é produzido a partir de uma regra geral e um caso específico. Entretanto, na dedução, o resultado é um conseqüência lógica da regra geral e do caso que são verdadeiros. A regra de inferência *modus ponens* é a forma básica de raciocínio dedutivo. O esquema geral para representar o raciocínio dedutivo está apresentado a Tabela 2.

Tabela 2. Exemplo do mecanismo de inferência dedução.

<i>Exemplo 1</i>	
<i>Dedução</i>	<i>a = Todas as bolas são da caixa. b = Todas bolas são pretas.</i>
<i>a -> b</i> <u><i>a</i></u>	Regra geral: <i>Todas bolas da caixa são pretas. (a -> b)</i> Caso : <i>Todas bolas são da caixa. (a)</i>
<i>∴ b</i>	Conclusão : <i>Todas as bolas são pretas. (b)</i>

A indução é o mecanismo no qual são realizadas inferências do específico para o geral a partir da mesma situação. O esquema geral para representar este mecanismo está apresentado na Tabela 3.

Tabela 3. Exemplo do mecanismo de inferência indução.

<i>Exemplo 2</i>	
<i>Indução</i>	<i>a = Estas bolas são da caixa. b = Estas bolas são pretas.</i>
<i>a</i> <u><i>b</i></u>	Caso: : <i>Estas bolas são da caixa. (a)</i> Resultado : <i>Estas bolas são pretas. (b)</i>
<i>∴ a -> b</i>	Regra hipotetizada : <i>Todas as bolas da caixa são pretas. (a -> b)</i>

Conforme descrito em Peng & Reggia (1990), recentemente, o conhecimento na forma *causa -> efeito* passou a ser modelado, com o uso de raciocínio do tipo abdutivo. Informalmente, abdução ou inferência abdutiva significa "inferência das melhores explicações, ou as mais plausíveis, para um conjunto de dados ou fatos". A abdução juntamente com a dedução e a indução correspondem às três formas fundamentais de raciocínio. Segundo estudiosos da área, tanto a inferência indutiva como a abdutiva, envolve fazer e testar hipóteses. Tanto que muitas vezes o termo "indução" é utilizado para descrever um processo de inferência que inclui inferência abdutiva. A diferença está na forma de raciocínio; enquanto a indução faz inferências do específico para o geral a partir da mesma situação, a abdução tenta fazer inferências a partir de uma regra geral para situação que não são as mesmas. Por outro lado, a inferência dedutiva é similar à abdução pois é produzido um resultado específico a partir de uma regra geral e um caso específico. Entretanto, na dedução, o resultado é um consequência lógica da regra geral e do caso que são verdadeiros. Já na abdução, se a regra geral (conhecimento geral) e resultado específico (fatos observados) são verdadeiros, o caso específico inferido é somente uma possibilidade, e não uma verdade definitiva. Embora seja possível a criação

de regras causa-efeito que concluem uma causa plausível para um conjunto de efeitos, os especialistas nem sempre conseguem ser precisos e raramente tem certeza absoluta sobre a informação que fornecem (Peng & Reggia, 1990). O esquema geral para representar abdução está descrito na Tabela 4.

Tabela 4. Exemplo do mecanismo de inferência abdução.

<i>Abdução</i>	Exemplo 3
$a \rightarrow b$	$a =$ Todas as <i>bolas</i> são da caixa. $b =$ Estas <i>bolas</i> são pretas.
b	Regra geral: <i>Todas bolas da caixa são pretas. (a \rightarrow b)</i> Caso _____ : <i>Estas bolas são pretas. (b)</i>
$\therefore b$ é plausível	Hipótese Plausível : <i>Estas bolas são da caixa. (a)</i>

Finalmente, após descrever os principais mecanismos de representação do conhecimento e de inferência de um SE, pode-se descrever as diferentes arquiteturas propostas para o ambiente de diagnose remota. Cabe lembrar ainda que a base de conhecimento um sistema especialista para diagnóstico de doenças do milho, que foi desenvolvido na Embrapa Informática Agropecuária em parceria com a Embrapa Milho e Sorgo, foi utilizada como estudo de caso no presente trabalho (Massruhá et al., 2000).

A base de conhecimento original contém 41 doenças de milho com suas respectivas manifestações ou efeitos. No escopo deste trabalho foi necessário coletar informação adicional das 41 doenças de milho para representar a base de conhecimento. O conhecimento utilizado para complementar a base foi obtido através de entrevistas com fitopatologistas da Embrapa e médicos, além de literatura correspondente na área (Reis & Casa, 1996; Fernandes & Oliveira, 1997; Malcolm, 1980; Agrofite, 2003). Adicionalmente a base de desordens, foi especificada uma base de conhecimento contemplando 9 testes laboratoriais e uma base de conhecimento de tratamentos (corretivos) contemplando somente os fungicidas registrados pelo Ministério da Agricultura. A especificação completa da base de conhecimento estão apresentados em Massruhá (2003).

A seguir, são apresentadas as quatro arquiteturas adotadas pela Embrapa Informática Agropecuária: WebLS, ServCLIPS, PROLOG e JAVA (J2EE).

Na Fig. 4, está apresentada a Arquitetura I - WebLS, utilizada no desenvolvimento de um sistema especialista para diagnóstico de doenças do milho que foi desenvolvido na Embrapa Informática em parceria com a Embrapa Milho e Sorgo (Massruhá et al., 2000).

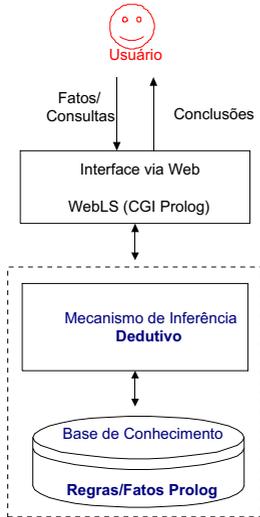


Fig. 4. Arquitetura - WebLS.

A primeira versão do sistema especialista para diagnóstico de doenças do milho (v1.0) foi baseado em regras onde o mecanismo de inferência é dedutivo e sua base teórica é a lógica de primeira ordem. A base de conhecimento deste sistema é constituída de regras de produção geradas a partir de uma árvore de decisão. A árvore de decisão foi gerada a partir de entrevistas com especialistas e de literatura na área (Fig. 5).

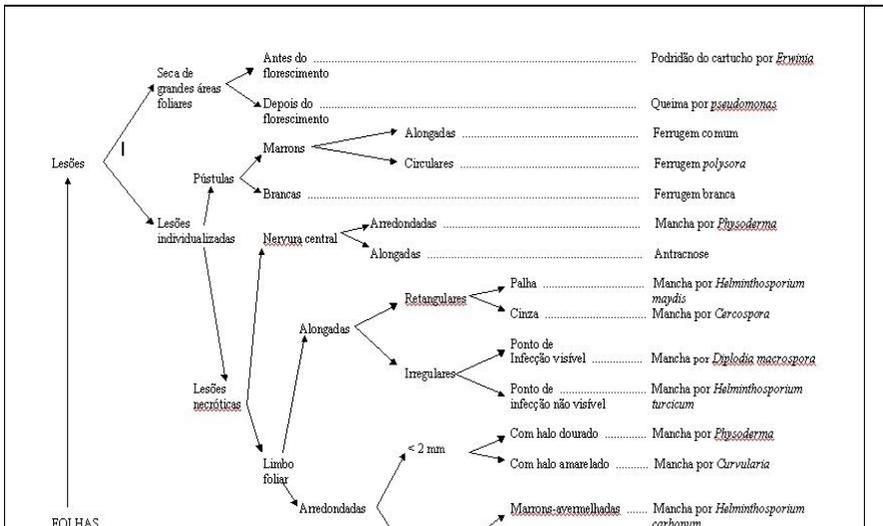


Fig. 5. Exemplo da árvore de decisão para diagnóstico de doenças de milho.

É válido lembrar que, os especialistas descrevem o conhecimento do domínio em direção oposta das regras de produção, isto é, "se uma doença está presente então algumas manifestações ocorrem". Para fins ilustrativo, abaixo está um parágrafo correspondente a descrição de uma doença de milho denominada Ferrugem Comum (d3) retirado de Fernandes & Oliveira (1997).

"A Ferrugem comum, sob condições favoráveis, pode causar seca prematura das plantas, comprometendo a produção. Essa ferrugem pode ser encontrada em qualquer fase do desenvolvimento das plantas de milho, inicialmente nas folhas baixas. Ferrugem Comum forma pústulas predominantemente alongadas, de cor marrom-clara em plantas jovens que tornam-se marrom-escuro à medida em que a planta se aproxima da fase de maturação. Essas pústulas podem ser encontradas em ambas as faces das folhas, e devido ao rompimento da epiderme, apresentam caracteristicamente uma fenda. É uma doença favorecida por temperaturas entre 16 e 23°C e umidade relativa alta."

O sistema de diagnóstico de doenças do milho foi baseado em regras de produção que geralmente tem a forma básica:

SE <manifestações> ENTÃO <causas>

Um exemplo das regras de produção utilizadas na implementação do sistema é:

```
R1: IF local="folhas" AND aparência = "Lesão" AND forma="individualizadas" AND
tipo="pústula" AND cor="marrom" AND formato = "alongada" THEN d1;
```

É interessante observar que na implementação destas regras de produção, não foram consideradas características temporais bem como condições favoráveis para o desenvolvimento da doença tais como, temperatura e umidade. Incertezas das manifestações causadas por determinadas doenças e dos fatos observados também não são contemplados neste sistema o que o torna bastante limitante.

A primeira versão do sistema foi desenvolvida em WEBLS que é um shell para implementar sistemas especialistas do tipo Q&R (questões e respostas) para Windows. A ferramenta WebLS é um CGI (Common Gateway Interface) sobre uma máquina de inferência Prolog que permite a construção dinâmica de páginas e formulários HTML (Hyper Text Markup Language) o que facilita a construção de sistemas especialistas para web (AMZI, 2004). Para esta primeira versão do ambiente foram desenvolvidos (Massruhá, et al., 1999, 2000):

- alguns padrões de interface para os aplicativos web;
- um modelo de dados para incorporar figuras utilizadas, resultados das consultas, base de clientes, etc.;
- um aplicativo de teste do ambiente - diagnose de doenças do milho.

Conforme descrito em Moura & Cruz (2001), embora esta solução fosse satisfatória era necessário migrar o ambiente para a plataforma UNIX e obter uma ferramenta que pudesse ser encapsulada em um aplicativo para CD, que pudesse ser desenvolvido em Java para o caso de decidir-se utilizá-lo em applets (aplicações clientes em Java que rodam em um navegador web) (Sun Microsystems, 2005).

Na Fig. 6 está apresentada a arquitetura II - ServCLIPS para construção de sistemas especialistas via web proposta em Cruz et al. (2001) que é muito similar a arquitetura I. Nesta arquitetura foi desenvolvido o ServCLIPS que utiliza o CLIPS como máquina de inferência e, conseqüentemente, a linguagem de programação do CLIPS para desenvolvimento dos sistemas especialistas. Assim, para que um especialista desenvolva um SE com o ServCLIPS, basta que ele esteja familiarizado com a linguagem CLIPS para representar as regras de produção.

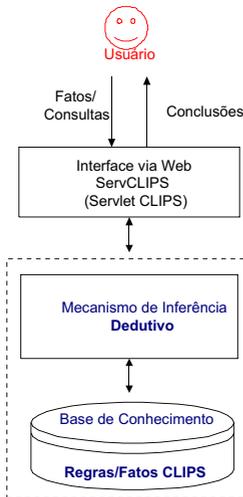


Fig. 6. Arquitetura II - ServCLIPS.

Observe que a versão atual do sistema para diagnóstico de doenças de milho (disponível em <http://diagnose.cnptia.embrapa.br/milho>) foi desenvolvida sobre a ferramenta ServCLIPS que possibilita a construção e execução de sistemas especialistas via internet. Mais detalhes podem ser encontrados em Cruz et al. (2001).

Os grandes desafios da construção destes sistemas estão relacionados com a aquisição, modelagem e representação do conhecimento. Nas arquiteturas I e II, foram utilizadas técnicas manuais para aquisição do conhecimento baseadas em entrevistas, em acompanhamento ou modelos. Quanto à representação do conhecimento foram utilizadas regras de produção que são baseados nos mecanismos de inferência dedutivo (efeitos -> causa). Conforme descrito anteriormente, o problema principal com este

tipo de abordagem é que o conhecimento do tipo *efeitos* → *causas* é contrário à maneira com que os fenômenos ocorrem na natureza (*causas* → *efeitos*). Na década de 1990, o conhecimento na forma *causas* → *efeitos* passou a ser modelado, com o uso de raciocínio do tipo abduativo; a partir das associações (*causas* → *efeitos*) e fatos (*efeitos*), criam-se novas hipóteses (*causas*). Uma outra limitação dos modelos apresentados é que os especialistas muitas vezes determinam o quanto uma manifestação está associada a uma desordem de uma forma não-numérica ("muito comum", "moderado", "muito raro", etc.), termos estes que podem ser tratados como tal, no que se chama de "*computing with words*", ou traduzidos em números, usando probabilidade ou outros modelos alternativos, como teoria de possibilidades, teoria de Dempster-Shafer ou fatores de certeza. A seguir será apresentada a abordagem proposta em Massruhá (2003) onde é utilizado o mecanismo de inferência abduativo integrado com lógica *fuzzy*.

Em Massruhá (2003) foi apresentada a arquitetura III - AMZI PROLOG que é uma abordagem integrada para diagnóstico, investigação e tratamento de desordens (Massruhá et al., 2003, 2004, 2005a, 2005b). Nesta nova abordagem, denominada Teoria das Coberturas Nebulosas (TCN), o conhecimento é basicamente modelado através de associações causais e a inferência é abduativa (desordens->manifestações), de maneira a tratar os vários aspectos inerentes aos processos envolvidos em raciocínio clínico. A TCN foi validada para diagnóstico, investigação e tratamentos de doenças de milho. A especificação completa da base de conhecimento e os algoritmos que implementam a TCN estão apresentados em Massruhá (2003). Na Fig. 7 está apresentada a terceira abordagem a ser comentada neste trabalho.

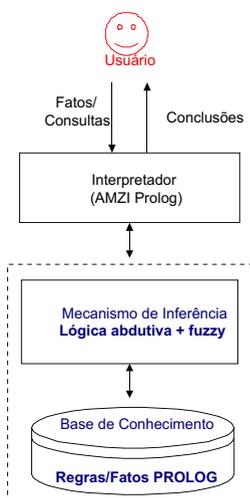


Fig. 7. Arquitetura III - AMZI PROLOG.

A linguagem utilizada na implementação dos algoritmos da TCN foi o AMZI PROLOG devido a sua sintaxe e aderência ao problema (Amzi, 2004). Adicionalmente, a implementação em PROLOG facilitou a comparação da TCN com o TCP original e TCP temporal através das ferramentas que também foram desenvolvidas em PROLOG no escopo do trabalho de Wainer & Rezende (1997). Os programas em PROLOG que compõem a TCN são claramente divididos em três partes: base de conhecimento, mecanismo de inferência e especificação do caso particular. O mecanismo de inferência abduutivo, foi implementado através dos algoritmos: *bipartido_nebuloso*, *exames_nebulosos* e *tratamentos_nebulosos*. Os algoritmos da TCN foram aplicados para as 41 desordens de milho e os resultados obtidos em todos os casos foram compatíveis com o raciocínio do diagnosticador.

Embora a TCN seja uma ferramenta simples e poderosa para raciocínio de diagnóstico, investigação e tratamentos, o modelo proposto tem algumas limitações relacionadas com a sua implementação. A geração da base de conhecimento demanda muito tempo e atenção. Portanto, seria interessante uma ferramenta que pudesse auxiliar na geração automática da base de conhecimento. Outro ponto que poderia ser melhor tratado seria a integração bem como a interface dos três módulos em PROLOG que compõem a TCN (*bipartido_nebuloso*, *exames_nebulosos* e *tratamentos_nebulosos*) que são executados separadamente. Mais detalhes desta arquitetura podem ser encontrados em Massruhá (2003) e Massruhá et al., (2005b). Finalmente, será descrita a abordagem orientada a objetos proposta neste trabalho para suprir estas limitações.

A IV arquitetura - JAVA (J2EE) da aplicação proposta em Massruhá (2003) foi revisada no âmbito do projeto proposto em Massruhá (2004). Após a revisão, a nova arquitetura foi desenhada no modelo três camadas, conforme as mais rigorosas especificações do J2EE. Isto permite que se tenha uma simplificação no processo de manutenção e expansões futuras do sistema. Na Fig. 8 está apresentada a abordagem orientada a objetos proposta neste trabalho.

O modelo J2EE em três camadas contempla:

- **Camada de Apresentação:** responsável pelo layout, navegação e tratamento das ações
- **Camada de Negócios:** responsável por todas as lógicas e regras que definem a aplicação;
- **Camada de Persistência:** responsável pelo armazenamento e consulta ao Banco de Dados.

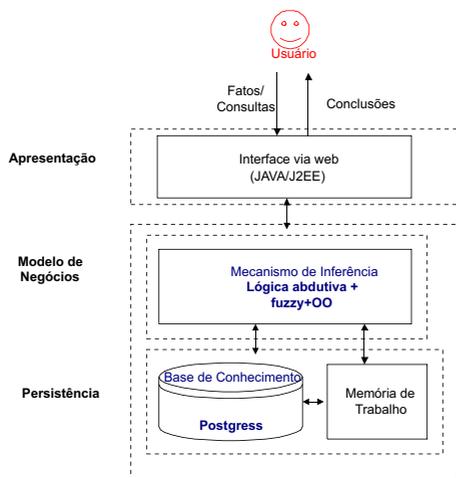


Fig. 8. Arquitetura IV - JAVA (J2EE).

Um sistema feito em camadas, com a estrutura apresentada na Fig. 8, proporciona maior dinamismo para expansões futuras.

O segundo passo, foi montar uma infra-estrutura do projeto (arquitetura) baseada em definições de padrões de projeto e frameworks. Nesta etapa, foram utilizados os mais importantes frameworks (em código livre), para tornar mais produtivo o processo de desenvolvimento, a saber:

- Struts: facilita e padroniza o controle das ações vindas da camada web e sua apresentação (<http://struts.apache.org/>);
- Spring: injeção de dependências, instanciando objetos antecipadamente (<http://www.springframework.org/>);
- Hibernate: facilita a persistência e cria uma abstração para maior independência da base de dados (<http://www.hibernate.org/>).

Para que o projeto pudesse ser implementado em tecnologia J2EE foi necessário inicialmente modelar o problema sob uma perspectiva de Orientação a Objetos (OO), bem como desenhar toda a Base de Dados, que no sistema original era apenas uma base simplificada de conhecimento textual em Prolog, contendo apenas a chave de identificação (id) e nome da entidade.

Na Fig. 9 está representado o modelo de classes gerado nesta etapa a partir da revisão da arquitetura da TCN. O modelo é composto por 6 classes principais (Desordens, Manifestações, Exames, Tratamentos, Problema e Caso) e 12 classes auxiliares (Grafo, Nó, Aresta, Nebulosos, Condições

Controle, Categoria, Fase, Severidade, Pergunta, Cidade, Evento e Usuário). A seguir, na Fig. 10 é mostrado o diagrama final com o modelo de dados projetado que contempla os três módulos: diagnóstico, investigação e tratamentos. O diagrama da Fig. 10 está representado no Modelo de Entidades e Relacionamentos (MER) que cobre todas as necessidades do projeto, em conformidade com a tese e requisitos. Observe que a quantidade de entidades é bastante grande, e que estas possuem muitas relações, expandindo assim as possibilidades e velocidade de acesso em relação ao que existe atualmente em forma textual.

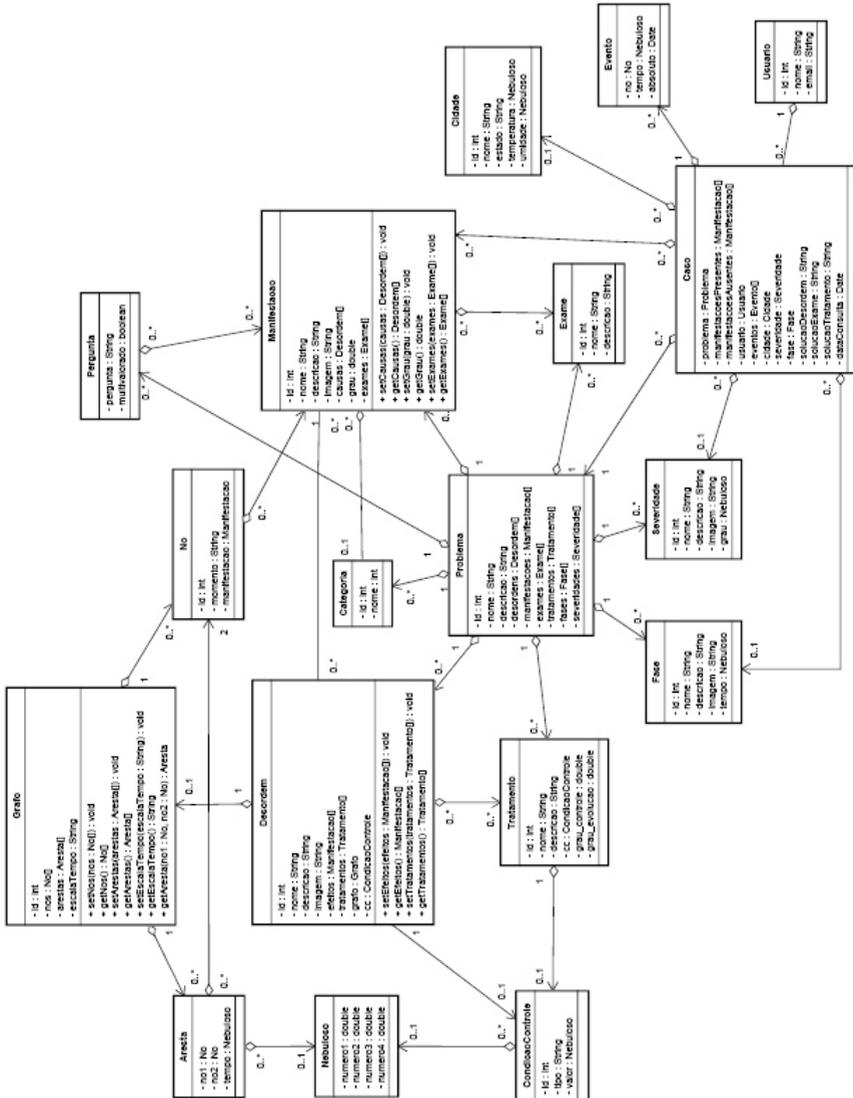


Fig. 9. Modelo de classes (orientação a objetos).

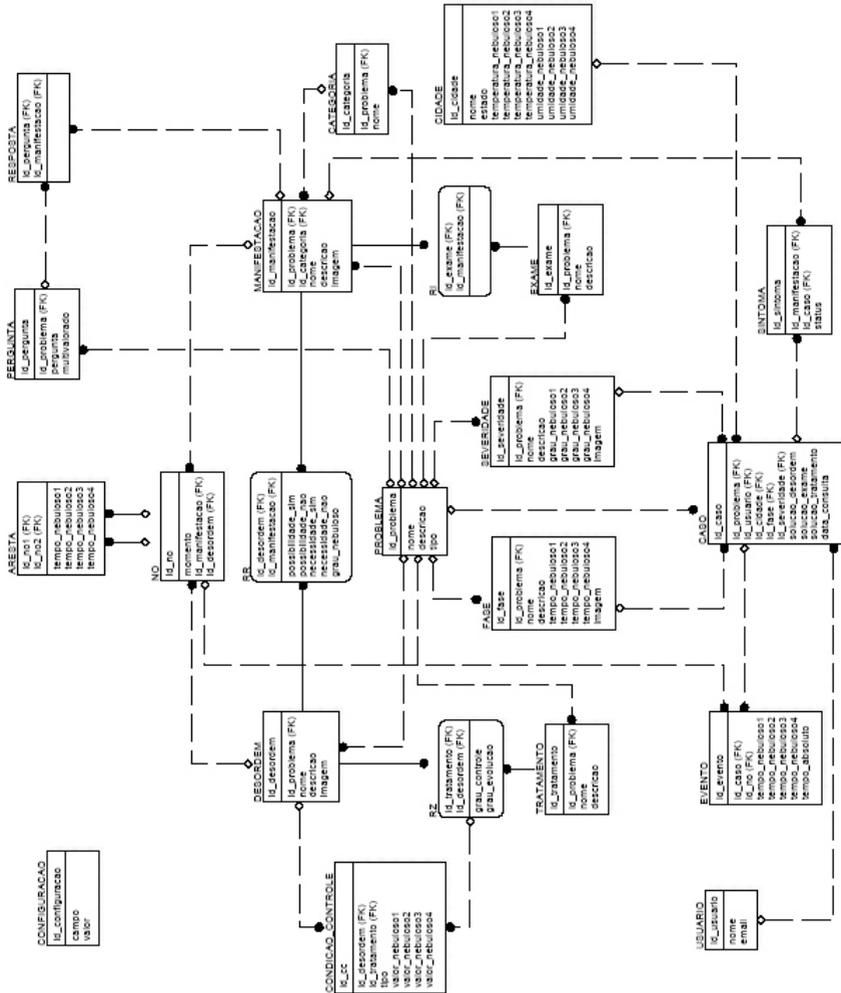


Fig. 10. Modelo de Entidades e Relacionamentos (MER).

Para melhor compreensão das etapas desenvolvidas, segue uma explicação mais detalhada:

Base de Conhecimento: este tópico contempla o tratamento da cadeia do conhecimento que engloba desde da interface gráfica ao registro no banco de dados das operações básicas sobre os objetos de valor, tais como: cadastro, edição, remoção e listagem. Neste ponto foi essencial o conjunto de frameworks composto por: Spring, Struts e Hibernate, o que trouxe agilidade e flexibilidade para mudanças futuras. A base de conhecimento que no protótipo em Prolog considerava apenas as entidades e campos principais, foi ampliada permitindo maior controle e organização dos dados. Informações como: cidade, categoria, usuário, fase, severidade, dentre

outros não eram tratados pelo sistema ou ainda eram tratados como números, o que não era nada intuitivo do ponto de vista de usabilidade para quem cadastra ou consulta o sistema para tomada de decisões. Esta parte no protótipo em Prolog era muito restrita, pois além das deficiências citadas, não possuía interface para cadastro.

Tecnicamente, os predicados auxiliares (que numericamente representam a maioria), tais como os que manipulam os nebulosos, as operações de conjunto, algébricas, dentre outros, quando passados para Java viraram métodos dentro dos objetos tipo TO (Transfer Objects) (Fig. 11). Por sua vez, as operações básicas de manipulação de dados da base, tais como: criação, remoção, edição e listagem foram feitas diretamente nos objetos tipo BO (Business Object), que também utilizam os respectivos DAO (Data Access Object) para o mesmo fim.

Apresentação

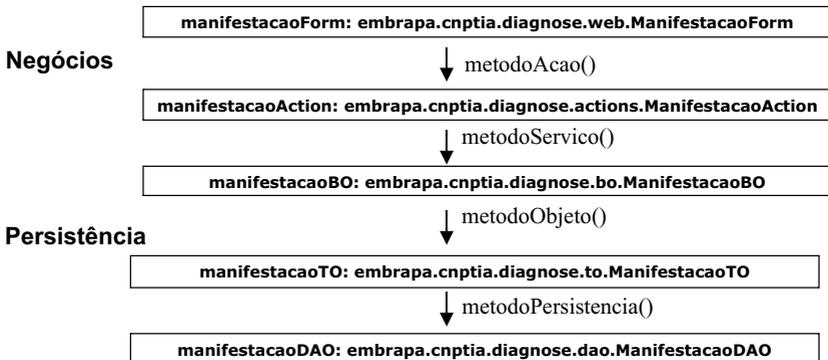


Fig. 11. Diagrama com Arquitetura do Sistema em Camadas.

Métodos de Negócio: existe uma distinção bem clara entre os métodos auxiliares, que extrapolam para as três etapas do processo (diagnóstico, investigação e tratamento) e os métodos específicos, sendo que estes são os mais críticos do ponto de vista de implementação e teste, pois manipulam a essência dos cálculos e tomadas de decisão. Para testar os métodos individualmente (chamando um método por vez) foram criadas classes designadas tecnicamente por *drivers* com o apoio da ferramenta de teste *JUnit* (Sun Microsystems, 2005). Alguns códigos em Prolog estavam quebrados em vários métodos por uma limitação de interações aninhadas desta linguagem. Ao serem transcritos para a tecnologia Java eventualmente foram agrupados, tornando a leitura do código simplificada e sua execução mais eficiente.

Interface Gráfica: para definição da melhor forma de navegação para o usuário, foram experimentadas diferentes telas (e seqüência delas) em um protótipo, que fora utilizado pelo especialista para sugerir um caminho e para validar o fluxo de navegação. Esta parte acabou ficando para o final do projeto, pois apresentava impacto mínimo na base de conhecimento e nos métodos de negócio. A integração das três partes foi tranqüila e demonstrou ter sido correta a escolha pela modularização, facilitando os testes e a qualidade final do sistema. Do ponto de vista estético o sistema possui uma aparência sóbria e bastante adequada ao propósito, tendo sido definida logo no início para que as telas já pudessem ser construídas no modelo correto. Do ponto de vista da usabilidade foi introduzido um grande diferencial, a tecnologia AJAX que permite que sejam enviadas requisições para os servidores, a partir de páginas HTML, sem que haja a necessidade de troca de telas. Tal funcionalidade vem sendo adotada pela Internet em serviços como Google Suggests, GMail, Google Maps e Yahoo Mail. No sistema ela é usada, por exemplo, para a partir do Estado, filtrar as Cidades correspondentes, evitando assim inconsistências. Com este mecanismo, o usuário não precisa mudar de página para ter as informações filtradas em conformidade com o seu caso. A tecnologia AJAX alia o uso de Javascripts e XML (eXtensible Markup Language). Mais detalhes da estrutura da tecnologia AJAX podem ser encontrados em <http://ajaxtags.sourceforge.net/>.

Os principais objetivos da primeira etapa foram alcançados com as definições acima. O ambiente, modelagem e toda a documentação para a codificação do projeto foram criados, abrindo caminho para o desenvolvimento.

Antes do início das atividades de desenvolvimento, foram propostas e introduzidas algumas alterações no ambiente de desenvolvimento do sistema, dentre elas a adoção da ferramenta de desenvolvimento Eclipse e a definição do sistema gerenciador de banco de dados pelo PostgreSQL. O Eclipse foi escolhido por ser o software open-source para desenvolvimento Java mais adotado pelas empresas, além de ser facilmente estendido para desenvolvimento J2EE integrado ao Servidor de Aplicações Tomcat. O Eclipse também conta com inúmeros plugins que facilitam a construção de páginas JSP (Java Server Pages) e de XML (eXtensible Markup Language).

Esta primeira versão do sistema executa na forma de uma aplicação web, com Banco de Dados em PostgreSQL. Sendo de interesse pode-se mudar o Banco de Dados para outro qualquer, apenas alterando as configurações da camada de persistência. Outra simplicidade proporcionada pelo modelo é a possibilidade de se construir uma versão da aplicação para ser distribuída em CD-ROM, neste caso basta alterar a camada de apresentação, mantendo as outras duas intactas.

O PostgreSQL, além de estar de acordo com as atuais diretrizes governamentais relacionadas ao uso de software livre, é robusto e tem uma alta performance, eliminando assim a necessidade de aquisição de licenças de uso de banco de dados proprietário tanto para o ambiente de desenvolvimento como para o ambiente de produção.

A árvore de navegação, ilustrada na Fig. 12, foi construída para organizar o fluxo das informações.

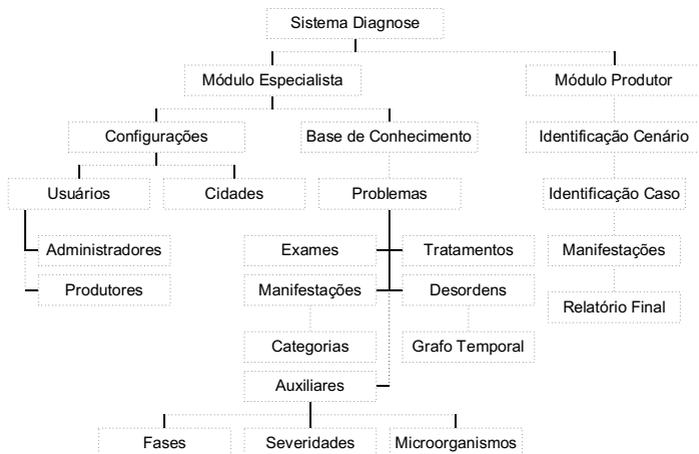


Fig. 12. Árvore de navegação do fluxo de informação do sistema.

Neste ponto fica clara a distinção de papéis entre o especialista e os produtores rurais. Enquanto os primeiros possuem a responsabilidade de configurar e alimentar o sistema com a base de conhecimento, o segundo grupo entra com as informações específicas do caso, obtendo como resultado uma indicação de quais as prováveis desordens encontradas e recomendações de tratamentos.

Resultados e Discussão

Nesta seção são apresentados os resultados obtidos no desenvolvimento da arquitetura IV no âmbito do projeto proposto em Massruhá (2004). Antes de iniciar o processo de desenvolvimento do software, todos os códigos Prolog foram traduzidos manualmente para a tecnologia e lógica JAVA, pois os softwares que se propunham a automatizar este processo foram bem insuficientes. O código traduzido, pelas ferramentas automáticas, além de ficar ilegível, tornava a manutenção impossível, aumentando o trabalho final ao invés de reduzi-lo. Por isso, esta alternativa foi abandonada após alguns testes, com diferentes produtos. A partir desta análise, o sistema foi implementado em JAVA, seguindo as especificações do modelo J2EE.

Para testar o núcleo do sistema, foi necessário realizar testes modulares *bottom-up*. No início foram estudadas as entradas e saídas de métodos auxiliares, tais como: nebulosos, intervalos, etc. Todo o princípio dos testes era o de executar a mesma chamada em Prolog e Java e comparar a saída, que deveria ser idêntica. Na seqüência foram realizados também testes similares com os métodos específicos com cada uma das três etapas do processo: diagnóstico, investigação e tratamento. A partir da conclusão destes testes e da manutenção da base de conhecimento, pode-se afirmar que se tem um sistema completo e que atende às necessidades e expectativas de seus usuários, tanto especialistas como produtores.

Após a conclusão do sistema, o projeto foi validado pela equipe responsável dentro da Embrapa, como forma de cancelar os requisitos desenvolvidos versus o que havia sido planejado.

A base de conhecimento de um sistema especialista para diagnóstico de doenças do milho, que foi desenvolvido na Embrapa Informática Agropecuária em parceria com a Embrapa Milho e Sorgo, foi utilizada como estudo de caso no presente trabalho (<http://diagnose.cnptia.embrapa.br>). Adicionalmente, a base de desordens, foi especificada uma base de conhecimento contemplando nove testes laboratoriais e uma base de conhecimento de tratamentos (corretivos) contemplando somente os fungicidas registrados pelo Ministério da Agricultura. A especificação completa da base de conhecimento está apresentada em Massruhá (2003).

Dentre os resultados obtidos observou-se que a infra-estrutura atual é muito mais robusta, eficiente e flexível (Tabela 5). Um exemplo de flexibilidade que foi introduzida é a de permitir que exatamente o mesmo sistema (em uma única instância) seja capaz de distinguir entre diferentes problemas da agricultura (milho, soja, algodão, etc.) e da pecuária (gado, ovelha, etc..). Para tal, somente é necessário que o próprio especialista cadastre outras bases de conhecimento (com doenças, manifestações, exames, etc) para os respectivos problemas da agricultura e da pecuária.

Tabela 5. Comparação entre as 4 arquiteturas propostas.

	Representação do conhecimento	Mecanismo de Inferência	Base teórica	Ferramentas de Implementação	Internet	Desenvolvedor	Usuário
I	Regras de produção	Dedução	Lógica de predicados de primeira ordem	WeBLS – CGI Prolog	Sim	Engenheiro do conhecimento	Produtor e especialistas
II	Regras de produção	Dedução	Lógica de predicados de primeira ordem	SevCLIPS	Sim	Engenheiro do conhecimento	Produtor e especialistas
III	Regras de produção	Abdução + Lógica nebulosa	TCP + lógica nebulosa	AMZI Prolog	Não	Engenheiro do conhecimento	Produtor e especialistas
IV	OO	Abdução + lógica nebulosa	TCP + lógica nebulosa	JAVA (J2EE)	Sim	Especialista	Produtor e especialistas

Outra flexibilidade é a possibilidade de integração com outros sistemas, tal como o Agritempo (www.agritempo.gov.br) que fornece variáveis como temperatura e umidade, simplificando o uso e aumentando a precisão do sistema, já que uma pessoa não precisa ter estas informações de sua cidade para atingir o aproveitamento máximo do diagnóstico.

Boa parte do sistema foi desenhado de forma a permitir configuração, sem a necessidade de se refazer código o que permitirá um uso mais amplo do sistema (para várias culturas simultaneamente), com fases, severidades e comportamento temporal bastante distintos. Esta parametrização foi o cerne deste trabalho de pesquisa, pois assim, é possível utilizar o sistema não apenas para a cultura de milho (inicialmente proposta), mas para qualquer outra, sem a necessidade de ter que executar diferentes instâncias do mesmo sistema.

Cabe ressaltar ainda que o modelo proposto na arquitetura III, implementado na linguagem de representação lógica PROLOG, foi mapeado para a linguagem JAVA (arquitetura IV) mantendo as propriedades inerentes ao modelo e obtendo um melhor desempenho. Neste primeiro projeto a idéia era informatizar totalmente o processo de diagnóstico e tratamento de desordens, mas na seqüência o objetivo é conseguir realizar estudos coletivos utilizando a base de conhecimento.

Conclusões

O resultado final alcançado, neste projeto de pesquisa, foi satisfatório tanto do ponto de vista de arquitetura da aplicação quanto do ponto de vista da aplicação final.

Em relação a nova arquitetura da aplicação desenhada no modelo três camadas, conforme as mais rigorosas especificações do J2EE, pode-se destacar que: a) mostrou-se mais flexível, robusta e eficiente que o Prolog, sob o ponto de vista de engenharia de software, para a implementação do modelo TCN; b) conforme previsto em Massruhá et al. (2005b), a geração automática da base de conhecimento, através do módulo especialista, demanda muito menos tempo e atenção; c) a integração dos três módulos que compõem a TCN (*bipartido_nebuloso*, *exames_nebulosos* e *tratamentos_nebulosos*) em uma única infra-estrutura tornou o sistema mais eficiente; d) a interface mais amigável dos três módulos que compõem a TCN (*bipartido_nebuloso*, *exames_nebulosos* e *tratamentos_nebulosos*) tornou o sistema viável sob o ponto de vista de usabilidade; e e) finalmente, confirmou-se a hipótese da simplificação no processo de manutenção e expansões futuras do sistema.

No escopo do aplicação, pode-se que concluir que os dois módulos principais do sistema, especialista e produtor, mostraram-se eficiente e robusto. O módulo especialista mostrou-se robusto para o gerenciamento da base de conhecimentos e o módulo produtor bastante eficiente do ponto de vista da lógica para a tomada de decisões.

A essência do problema esteve na definição de quais perguntas fazer ao produtor para que fosse possível determinar a desordem em questão, e também sugerir exames e tratamentos em conformidade com este resultado.

A simplicidade de uso, principalmente no módulo produtor foi perseguida com insistência, dado que os usuários que consultarem o sistema terão disponível um manual on-line e não terão qualquer tipo de treinamento presencial para tal manuseio. Nesta linha, as decisões sempre foram no sentido de retirar possíveis pontos de dúvida ou com múltiplas interpretações.

O projeto entregue permite trabalhar o processo de diagnóstico, investigação e tratamento de quaisquer doenças. Esta generalização é muito importante para a Embrapa, pois a quantidade de conhecimento da Instituição é farto, o que tornará possível explorar o mesmo mecanismo de apoio ao produtor em diferentes áreas do conhecimento rural.

Referências Bibliográficas

AGROFIT: relatório de acaricidas, bactericidas, fungicidas, inseticidas. Brasília, DF: Ministério da Agricultura, Pecuária e Abastecimento, 2003. 145 p. Relatório técnico.

AMZI! *Prolog+Logic Server*: tutorials. Disponível em: <http://www.amzi.com/products/prolog_products.htm>. Acesso em: 24 nov. 2004.

CRUZ, S. A. B.; LEITE, M. A. A.; MOURA, M. F.; SOUZA, E. ServCLIPS: a tool to develop www expert systems. In: WORLD CONGRESS OF COMPUTER IN AGRICULTURE AND NATURAL RESOURCES, 2002, Iguaçú Falls. *Proceedings...* Iguaçú Falls: ASAE, 2001. p. 636-641.

DUBOIS, D.; PRADE, H. Fuzzy relation equations and causal reasoning. *Fuzzy Sets and Systems*, v. 75, n. 2, p. 119-134, Oct. 1995.

DURKIN, J. *Expert systems: design and development*. Englewood Cliffs, NJ: Prentice Hall, 1994. 800 p.

FERNANDES, F.; OLIVEIRA, E. *Principais doenças na cultura do milho*. Sete Lagoas: EMBRAPA-CNPMS; Brasília, DF: EMBRAPA-SPI, 1997. 90p.

GIARRATANO, J. C.; RILEY, G. D. *Expert systems: principles and programming*. 2nd ed. Boston: PWS Publishing, 1994.

HENRION, M.; BREEZE, J. S.; HORVITZ, E. J. Decision analysis and expert systems. *Magazine*, v. 13, n. 8, p. 64-91, 1992.

HUBER, U.G.; DOLUSCHITZ, R. *An overview of expert systems in agriculture*. Bonn: ILB, 1990. 16 p.

LOPES, C. A.; ÁVILA, A. C. de. *Doenças do pimentão: diagnose e controle*. Brasília, DF: Embrapa Hortaliças, 2003. 96 p.

LOPES, C. A.; ÁVILA, A. C. de (Org.). *Doenças do tomateiro*. Brasília, DF: Embrapa Hortaliças, 2005. 151 p.

LUCAS, P. Modeling interactions for diagnosis. In: CESA'96 IMACS MULTICONFERENCE; SYMPOSIUM ON MODELLING, ANALYSIS AND SIMULATION, 1996. Lille, France. *Proceedings...* Lille, France: IMACS, 1996. p. 541-546.

MAHAMAN, B. D.; HARIZANIS, P.; FILIS, I.; Antonopoulou, E.; Sideridis, A. B.; Yialouris, C. P. A diagnostic expert system for honeybee pests. *Computers and Electronics in Agriculture*, v. 36, p. 17-31, 2002.

MAHAMAN, B. D.; PASSAM, H. C.; SIDERIDIS, A. B.; YIALOURIS, C. P. DIARES-IPM: a diagnostic advisory rule-based expert system for integrated pest management in Solanaceous crop systems. *Agricultural Systems*, v. 76, p. 1119-1135, 2003.

MALCOLM, C. *Compendium of corn diseases*. 2nd. ed. St Paul: American Phytopathological Society, 1980. 105 p. (APS. Disease Compendium Series).

MASSRUHÁ, S. M. F. S.; SOUZA, E. de; ROMANI, L. A. S.; CRUZ, S. A. B. Virtual services for agricultural technology transfer. In: EUROPEAN CONFERENCE OF THE EUROPEAN FEDERATION FOR INFORMATION TECHNOLOGY IN AGRICULTURE, FOOD AND THE ENVIRONMENT, 2., September 27-30, 1999, Bonn, Germany. *Role and potential of IT, Intranet and Internet for advisory services*. Bonn: Universität Bonn-ILB, 1999. p. 53-62.

MASSRUHÁ, S. M. F. S.; CRUZ, S. A. B. da; SOUZA, E. de. Diagnose virtual: um sistema para diagnóstico de doenças do milho via web. In: CONGRESSO DA SBI-AGRO, 2. = AGROSOFT 99, 1999, Campinas. Anais ... [S. l.: s. n.], 2000. Não paginado.

MASSRUHÁ, S. M. F. S. *Uma teoria de coberturas nebulosas para diagnóstico, investigação e tratamento*. 2003. 251 p. Tese (Doutorado em Computação Aplicada) - Instituto Nacional de Pesquisas Espaciais, São José dos Campos.

MASSRUHÁ, S. M. F. S.; SANDRI S. A.; WAINER, J. Fuzzy covering theory: an alternative approach for diagnostic problem-solving. In: EUROPEAN CONFERENCE FOR INFORMATION TECHNOLOGY IN AGRICULTURE - EFITA, 4., 2003, Budapeste. *Proceedings...* Budapeste: HAAI, 2003.

MASSRUHÁ, S. M. F. S. (Coord.). *Incorporação de ferramentas inteligentes na Agência de Informação Embrapa*. [Campinas: Embrapa Informática Agropecuária, 2004]. 30 p. (Embrapa. Macroprograma 3 - Desenvolvimento Tecnológico Incremental. Projeto).

MASSRUHÁ, S. M. F. S.; SANDRI, S. A.; WAINER, J. *Ordering manifestations for investigation in incomplete diagnosis*. In: 10th INTERNATIONAL CONFERENCE IPMU, 10., 2004, Perugia, Italy. *Proceedings...* [Perugia], 2004. p. 1153-1160.

MASSRUHÁ, S. M. F. S.; SANDRI, S.; WAINER, J.; MORANDI, M. An integrated framework for clinical problem solving in agriculture. In: CONFERENCE OF THE EUROPEAN FEDERATION FOR INFORMATION TECHNOLOGY IN AGRICULTURE, FOOD AND ENVIRONMENT, 5., WORLD CONGRESS ON COMPUTERS IN AGRICULTURE AND NATURAL RESOURCES, 3., 2005, Vila Real, Portugal. *Proceedings...* Vila Real, Portugal: Universidade de Trás-os-Montes e Alto Douro, 2005a. p. 1400-1407.

MASSRUHÁ, S. M. F. S.; SANDRI, S.; WAINER, J.; MORANDI, M. *Uma abordagem nebulosa para solução de problemas de diagnóstico, investigação e tratamento de desordens*. Campinas: Embrapa Informática Agropecuária, 2005b. (Embrapa Informática Agropecuária. Boletim de Pesquisa e Desenvolvimento, 12).

MICHALSKI, R. S., DAVIS, J. H., BISHT, V. S. AND SINCLAIR, J. B. PLANT/DS: an expert consulting system for the diagnosis of soybean diseases. In: PLANT DISEASES AND PROCEEDINGS OF THE EUROPEAN CONFERENCE ON ARTIFICIAL INTELLIGENCE, 1982, Orsay, France. *Conference proceedings*. p. 82-84 Disponível em: <<http://mars.gmu.edu:8080/dspace/bitstream/1920/1565/1/82-04.pdf>>. Acesso em: fev. 2007

MOURA, M. F.; CRUZ, S. A. B. da. *Estudo de expert system shells para o ambiente de diagnose remota*. Campinas: Embrapa Informática Agropecuária, 2001. 22 p. (Embrapa Informática Agropecuária. Documentos, 7).

PENG, Y.; REGGIA, J. A. *Abductive inference models for diagnostic problem-solving*. New York: Springer Verlag, 1990. 285 p.

POZZA, E. A.; MAFFIA, L. A.; SILVA, C. A. B; BRAGA, J. L.; CERQUEIRA, F. G. TOMEX-UFV: um sistema especialista para diagnose de doenças do tomateiro. In: CONGRESSO DA SBI-AGRO, 1. AGROSOFT 97, 1997, Belo Horizonte. *Anais...* [S. l.: s. n.], 1997. Não paginado.

REIS, E.; CASA, R. T. *Manual de identificação e controle de doenças de milho*. Passo Fundo: Aldeia Norte, 1996. 80 p.

REZENDE, S. O. *Sistemas inteligentes: fundamentos e aplicações*. Barueri: Editora Manole, 2003.

RUMBAUGH, J.; JACOBSON, I.; BOOCH, G. *The unified modeling language reference manual*. New York: Addison Wesley Longman, 1999. 550p.

Sun Microsystems. [*Java 2 Platform, Enterprise Edition (J2EE)*]. Disponível em: <<http://java.sun.com/j2ee/index.jsp>>. Acesso em: 10 jun. 2005.

TABOADA, M.; LAMA, M.; BARRO, S.; MARIN, R.; MIRA, J.; PALACIOS, F. A problem-solving method for unprotocolised therapy administration task in medicine. *Artificial Intelligence in Medicine*, v. 17, p. 157-180, 1999.

TUNEZ, S.; MARIN, R.; AGUILA, I.; BOSCH, A. T. M. An abductive method for solving a treatment problem. In: EUROMICRO CONFERENCE, ENGINEERING SYSTEMS AND SOFTWARE FOR THE NEXT DECADE, 24., 1998, Vasteras. *Proceedings*. Vasteras: IEEE Computer Society, 1998. v. 2, p. 737-744.

WAINER, J.; REZENDE, A. A temporal extension to the parsimonius covering theory. *Artificial Intelligence in Medicine*, v. 10, n. 3, p. 235-255, July, 1997.

WAINER, J.; SANDRI, S. A fuzzy temporal/categorical information in diagnosis. *Intelligent Temporal Information Systems in Medicine*, p. 1-19, 1998.

Yialouris, C. P.; Sideridis, A. B. An expert system for tomato diseases. *Computers and Electronics in Agriculture*, v. 14, p. 61-76, 1996.



Informática Agropecuária

**Ministério da
Agricultura, Pecuária
e Abastecimento**

