

DiagData: uma arquitetura para geração de sistemas inteligentes para predição de doenças de plantas

Raphael Fuini Ricciotti¹

Silvia Maria Fonseca Silveira Massruhá²

Este trabalho descreve a arquitetura de um sistema desenvolvido para utilizar a grande quantidade de dados e informações geradas pela Embrapa na área de manejo e controle de doenças de plantas, por meio de seus experimentos e publicações. No contexto deste trabalho, técnicas de mineração de dados foram utilizadas de modo que o conhecimento extraído destes dados possa ser utilizado para antever a ocorrência de uma doença e evitar a sua disseminação.

A utilização do conhecimento gerado para antever a ocorrência de uma doença é realizada usando duas ferramentas distintas, o Weka (WAIKATO, 2009) para mineração de dados e o Fuzzygen (LIMA; MASSRUHÁ, 2009) para executar o sistema de inferência *fuzzy*. Para utilizar este conhecimento na ferramenta de inferência *fuzzy* é necessário gerar um modelo *fuzzy*, no qual o conhecimento adquirido é inserido no modelo em forma de regras. Como as ferramentas são distintas, a inserção destas regras é feita de forma manual tornando esse processo complicado e demorado. A proposta deste trabalho

¹ Bolsista PBIC/CNPQ; Pontifícia Universidade Católica de Campinas;
raphael@cnptia.embrapa.br

² Pesquisadora; Embrapa Informática Agropecuária; Campinas;
silvia@cnptia.embrapa.br

surgiu com o objetivo de diminuir o tempo gasto no processo de aquisição de conhecimento e modelagem *fuzzy*.

O sistema nomeado DiagData, deve ser capaz de ler os dados, extrair conhecimento pelo algoritmo de árvore de decisão, utilizar esse conhecimento como entrada para o sistema de lógica *fuzzy* e inferir o resultado de forma semi-automatizada. Na etapa de extração do conhecimento e geração de regras é utilizado o algoritmo de árvore de decisão J48 ou C4.5 do Weka. O arquivo de entrada deve estar no formato .csv ou no formato de entrada de dados do Weka. Para a geração do sistema preditivo a partir das regras geradas pelo Weka, é utilizado o sistema de inferência *fuzzy* denominado *Fuzzygen*.

Todo o sistema DiagData foi implementado na plataforma *Java Standard Edition JAVA SE* (SUN MICROSYSTEMS, 2009). A Figura 1 ilustra os principais módulos do sistema, bem como seus principais relacionamentos. No módulo “Interface Integradora”, inicialmente, o usuário tem acesso a todos os passos do sistema, sendo o inicial a entrada dos dados para extrair conhecimento e visualizar as regras geradas, seguido da fase de modelagem *fuzzy* com a inserção das regras, das variáveis e seus valores, e finalmente inferindo o resultado e visualizando graficamente a “defuzzyficação”. Essas são as etapas fundamentais para a geração de um sistema inteligente preditivo. A modelagem, tendo sido processada com sucesso, possibilita ao usuário inferir o resultado baseado nas regras geradas.

No módulo “Árvore Hiperbólica” é disponibilizado ao usuário a visualização das regras em forma gráfica, na forma de árvore hiperbólica e no módulo “Engenho (web)” é disponibilizado o sistema de inferência *fuzzy* com interface web, que foi desenvolvido utilizando Java Server Pages (JSP), Servlet e Apache Tomcat. Observe na Figura 1 que o módulo “Engenho (web)” é executado independente do DiagData.

Este trabalho possibilitou semi-automatizar o processo de geração de regras e modelagem *fuzzy*. O processo de avaliação foi a medição e comparação do tempo gasto para inserir as regras sem o sistema DiagData, com o tempo utilizado para inserir essas mesmas regras com o sistema DiagData.

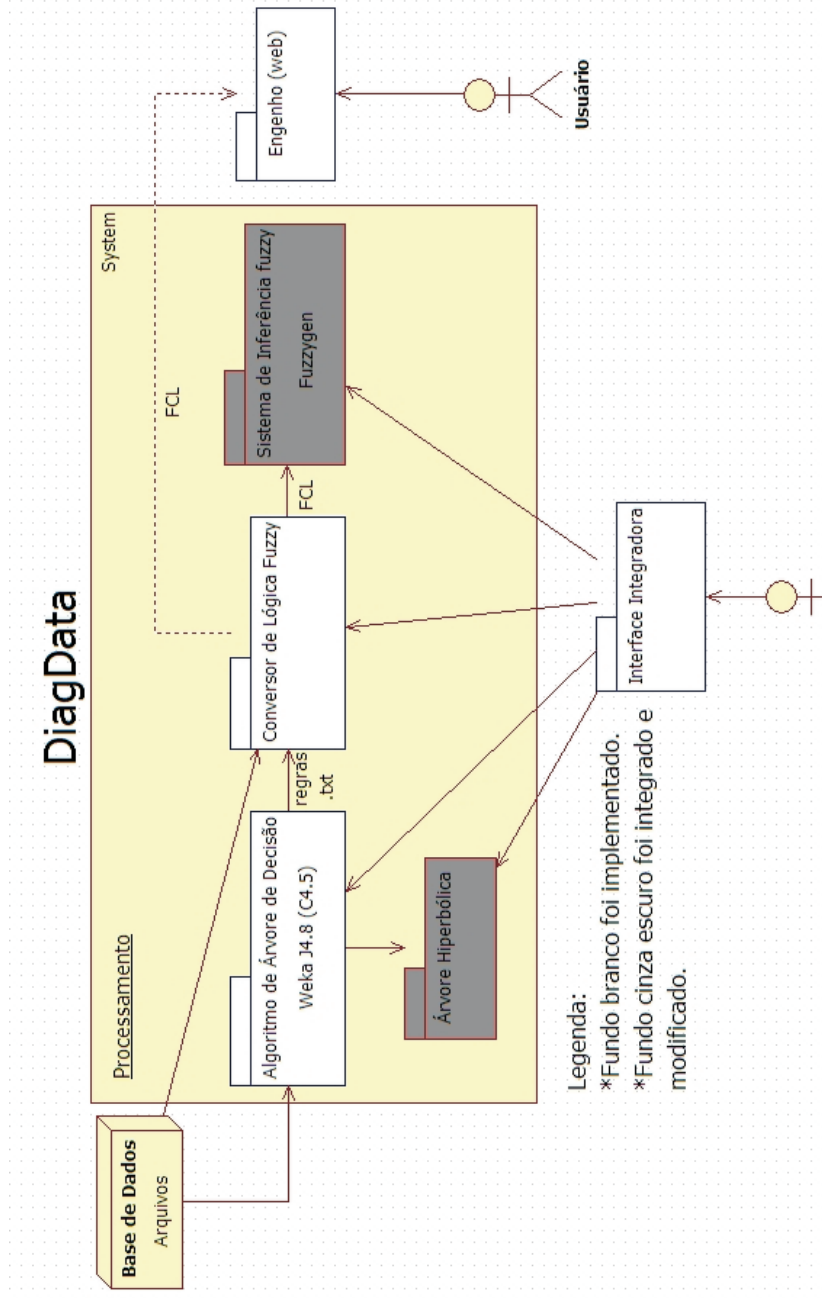


Figura 1. Diagrama de Arquitetura.

A metodologia escolhida para o desenvolvimento deste trabalho foi a XP (*Extreme Programming*) que é uma metodologia de desenvolvimento de software que visa à produção rápida de aplicações com alta qualidade. A abordagem desta metodologia é o desenvolvimento incremental adotada por muitos autores (BECK, 1999, 2000).

A ferramenta foi desenvolvida com o objetivo de utilizar dados de doenças de plantas, mas esta ferramenta é livre de contexto, ou seja, pode-se utilizar qualquer tipo de dados que possam ser minerados usando o algoritmo de árvore de decisão.

Com relação ao objetivo, foi identificado que ele foi alcançado, provendo uma redução de aproximadamente 55% do tempo gasto. Essa porcentagem poderia ser ainda maior se utilizar uma base de dados que gere uma quantidade maior de regras, ou seja, à medida que aumenta a quantidade de regras, a ferramenta mostra-se mais eficiente.

Referências

BECK, K. Embracing change with eXtreme programming. **IEEE Computer**, v. 32, n. 10, p. 70-77, Oct. 1999.

BECK, K. **extreme programming eXplained**: embrace change. Reading: Addison Wesley, 2000. 190 p. (

LIMA, H. P. DE; MASSRUHÁ, S. M. F. S. **Sistema FuzzyGen**: manual do usuário. Campinas: Embrapa Informática Agropecuária, 2009. 24 p. (Embrapa Informática Agropecuária. Documentos, 96).

SUN MICROSYSTEMS. **Java SE**. Disponível em: <<http://java.sun.com/j2se/>>. Acesso em: 11 maio 2009.

WAIKATO. **Environment for knowledge analysis. Version 3.6.0**. Disponível em: <<http://www.cs.waikato.ac.nz/ml/weka/>>. Acesso em: 11 maio 2009.