

# Research of Pest Diagnosis System Development Tools Based on Binary Tree

Yun Qiu, Guomin Zhou

► **To cite this version:**

Yun Qiu, Guomin Zhou. Research of Pest Diagnosis System Development Tools Based on Binary Tree. Daoliang Li; Yande Liu; Yingyi Chen. 4th Conference on Computer and Computing Technologies in Agriculture (CCTA), Oct 2010, Nanchang, China. Springer, IFIP Advances in Information and Communication Technology, AICT-345 (Part II), pp.300-308, 2011, Computer and Computing Technologies in Agriculture IV. <10.1007/978-3-642-18336-2\_36>. <hal-01562787>

**HAL Id: hal-01562787**

**<https://hal.inria.fr/hal-01562787>**

Submitted on 17 Jul 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# Research of pest diagnosis system development tools based on binary tree

Yun Qiu <sup>1</sup>, Guomin Zhou <sup>1</sup>

<sup>1</sup>Agricultural Information Institute of CAAS, Beijing 100081

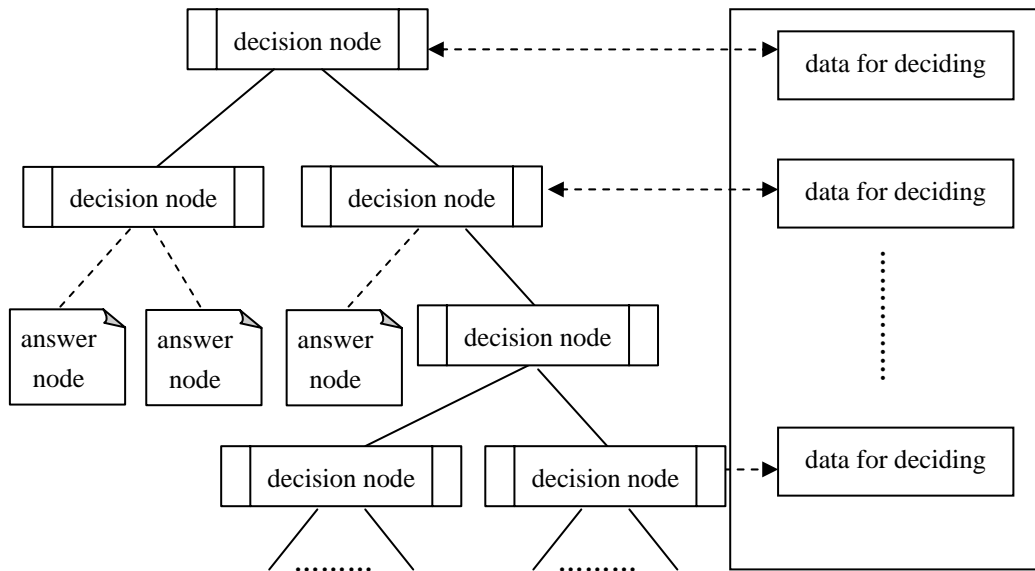
[Qiuyun@mail.caas.net.cn](mailto:Qiuyun@mail.caas.net.cn)

Abstract. A visual pests diagnosis system development tools based on binary tree (abbreviation: knowledge system development tools) was developed by techniques of computer visualization, binary tree, XML database and reasoning. This tool combined tree knowledge expression with rule-based reasoning machine, proposed the knowledge acquisition and the reasoning technique based on binary tree, and solved the problem of knowledge acquisition used for pests diagnosis with an expert system development tools--CLIPS. The reasoning machine with stateless continuous reasoning technology can improve the efficiency of knowledge acquisition and system reasoning. In this article, we introduced the components and working principles of the pest diagnosis system development tools, and described its usage with a simple example.

Key words: pest diagnosis; binary tree; CLIPS system; knowledge system; knowledge expression

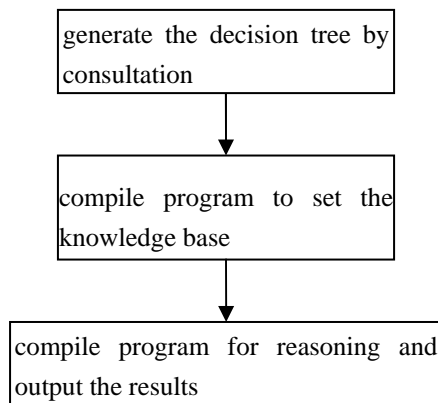
## 1 Introduction

Pest is one of the most terrible threats to crops. The statistics shows that there are 236 million hectares fields are threatened by pests and about 15 percents food are lost in our country every year. An effective measure to reduce crops lost and improve its quality is to get accurate information in time by pest diagnosis and take proper measures in the growth period. Taxa Key and Classification Tree which can be combined into a binary tree key by optimization analysis (shown in Figure1) are often used to summarize the experience of pest diagnosis by plant protection experts. The system based on Binary Tree Key usually includes a series of questions (named “decision node”), the decision makers will answer each questions in sequence till the terminal object (named “answer node or leaf node”) is identified.



**Fig.1.** structure of Binary Tree Key

The development process of expert system which can be expressed by binary tree is shown in Fig.2.



**Fig.2.** the development process of expert system

As shown above, program compiling is a complicated work must be done by professional training engineers. Can we develop a tool to package these difficult programs and make program compiling automatically? The answer is positive. A

kind of visual pests diagnosis system development tools based on binary tree can function properly.

## **2. Materials and methods**

### **2.1. Introduction of CLIPS system**

“CLIPS” is the abbreviation of “C Language Integrated Production System”. It is a rule-based general expert system development tools developed by National Aeronautics and Space Administration (NASA) in 1985. As a knowledge engineering language, CLIPS inherits its traditional writing habits and operating characteristics, and has the basic characteristics of production systems. After deeper development by NASA, CLIPS has several knowledge expression methods and fuzzy reasoning function. The knowledge expression methods include the production system (rule-based), the framework structure, the object-oriented and process programming. The fuzzy reasoning function may improve the accuracy of reasoning results. In addition, CLIPS system has a good openness that can integrate with other systems easily. A dynamic linked functions library-- clips.dll is provided in the software package of CLIPS development environment. The mixed programming can be completed by calling functions in clips.dll. For example, we can embed calculation unit into CLIPS to improve its poor calculate function as well as embed CLIPS into calculation unit to improve its poor logical reasoning. There are three basic elements used for expert system development is provided by CLIPS.

(1) fact-list and instance-list. They usually consist of some self-defined facts and include the data needed for reasoning.

(2) knowledge-base. It includes all rules that usually formed by self-defined.

(3) inference-engine. It reasons according to reasoning mechanisms and control procedures overall.

The fact in fact-lists is the basic form of the data in CLIPS system. Each fact is information in fact-lists. Whether the rule condition is satisfied or not depends on the state fact. A fact is made up with a series of domains separated by spaces. The fact can be inserted or deleted from fact-list before reasoning, and it also can be inserted and deleted as an act section of rule during reasoning.

Forward reasoning is the only rule adopted in CLIPS, it forms in “IF ...THEN...”. Its implementation technology is the famous Rete Algorithm, also named net-matching algorithm. To run the program of expert system is to fire and run each “IF... THEN ...” rule defined in program till getting the specific conclusion. The necessary

condition to fire the rule is each pattern in “IF” sentence matched with the facts in fact-list. It’s called be-activated rule. Be-activated rules may not fire immediately, thus a waiting agenda is required. The waiting agenda will fire the rule with the highest priority and run the action in “THEN” sentence. As thus, CLIPS includes four parts shown in Fig.3.

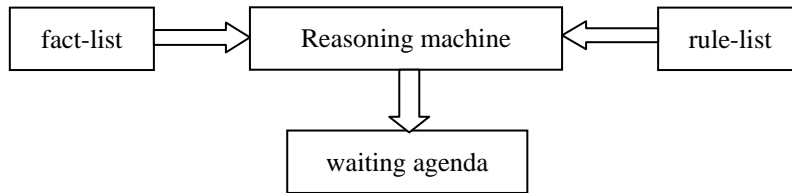


Fig.3. the structure of CLIPS

## 2.2. Components of development tools

Knowledge System Development tools is composed of two sections. One is knowledge editor functions for knowledge editing and uploading. Another is knowledge reasoning components and knowledge service webpage, they plays a role in receiving input information, calling knowledge from knowledge base to reason, and returning results to users by webpage.

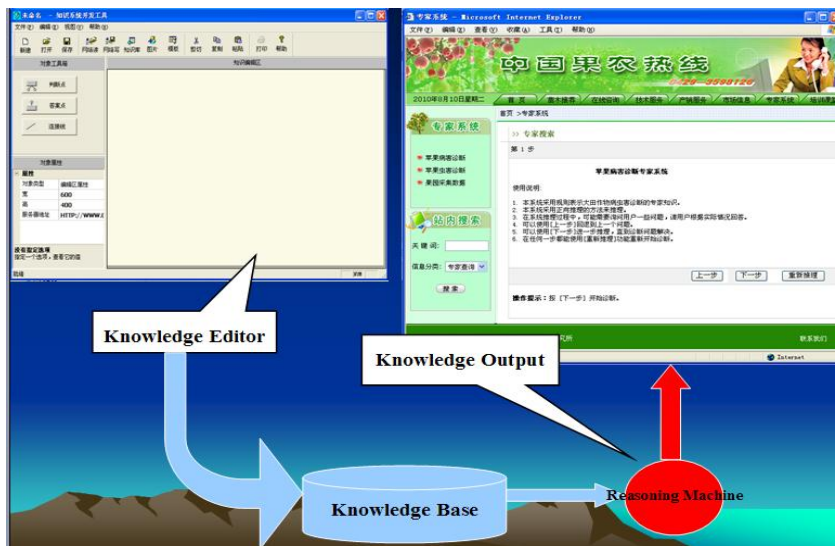


Fig.4. the structure of development tools

There are three main functions with knowledge system development tools: (1) visual knowledge editing function which can edit the knowledge in books and

experts' mind; (2) easy knowledge publishing function which can output knowledge to Internet by knowledge reasoning components and corresponding ASP script without compiling program; (3) flexible webpage customizing function which can customize webpage according to requirements to develop some personalized knowledge system.

### 2.3. Working principles of development tools

Knowledge expression is very important in the knowledge system. The rule-based knowledge expression method of CLIPS is used in this development tools. Each node is a fact. The information style in answer nodes and decision nodes is different, so the different models are needed. The model of answer nodes is expressed as: node <name> answer <value>). In this model, <name> is the name of answer node, and <value> is the answer in storage nodes. The model of decision nodes is expressed as: (node <name> decision <question> <yes-node> <no-node>). In this model, <name> is the name of decision node, and <question> is the question proposed by this node, <yes-node> and <no-node> are the nodes performed respectively for positive and negative answers. An example of knowledge expression is shown below:

```
(deftemplate rule                                (defrule goal-satisfied ""
(multislot if)                                  (declare (salience 30))
(multislot then))                              ?f <- (goal is ?goal)
(defrule propagate-goal ""                     (variable ?goal ?value)
(goal is ?goal)                                (answer ? ?text ?goal)
(rule (if ?variable $?)                       =>
(then ?goal ? ?value))                       (retract ?f)
=>                                             (format                                     t
(assert                                     (goal   "%s%s%n" ?text ?value))
is ?variable)))                             .....
```

However, the knowledge expression method above is difficult for common users. In order to solve this problem a visual editing method is chosen, so the common users can express knowledge by some simple icons. Users can express knowledge by icons. After finished knowledge editing, the development tools will translate knowledge expressed by icons into knowledge expressed by rules. The work principle of development tools is show in Fig.5.

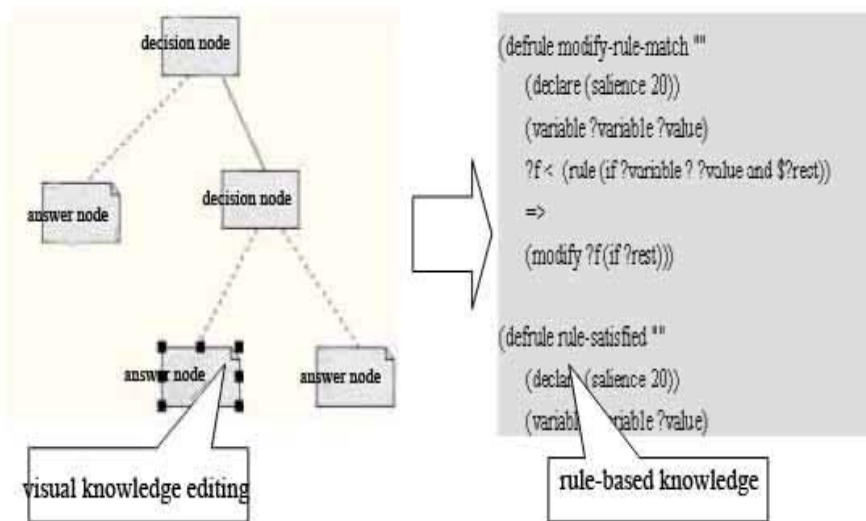


Fig.5. the work principles of development tools

After being edited and translated, knowledge is formed into rule-based ones and stored in knowledge base of web server. The reasoning machine implemented with COM+ technology of Microsoft can call the knowledge stored in knowledge base. The user interface is realized by ASP script. (Fig.6)

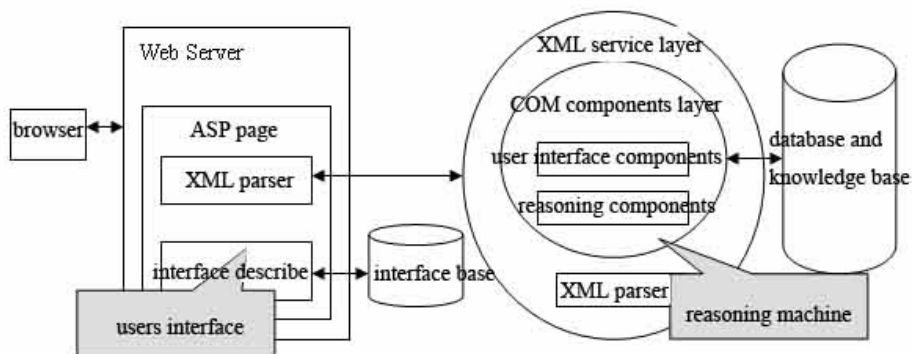


Fig.6. the logic structure of knowledge system

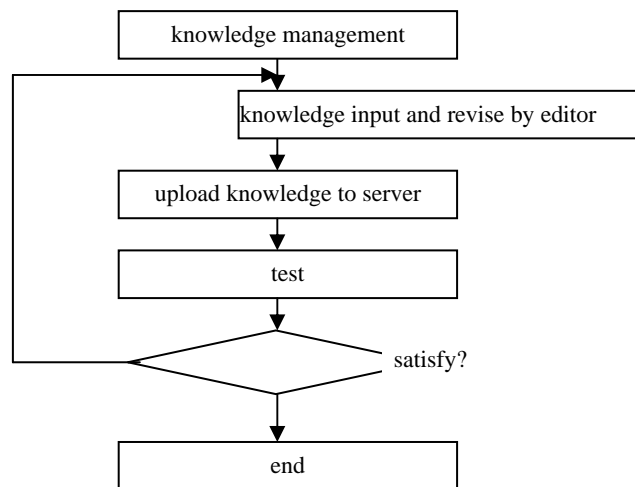
In fact, it isn't as complicated as the structure shown in Fig.6. The development tool has packaged some main functions into COM and provides a standard ASP script. Therefore, users can run the system by the program sentence as "ks

run.asp?kdb = name of knowledge base & title =name of system ” without compiling complicated program.

### 3. Results and Analysis

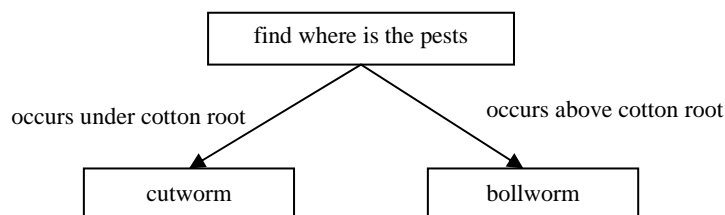
#### 3.1. An example of pest diagnosis

Suppose that we will develop a networked pest diagnosis system named “pest diagnosis experiment system”, the development process is shown in Fig.7.



**Fig.7.** the development process of knowledge system




According to the process above, knowledge management is the first step. Let’s suppose that there is only one pest diagnosis knowledge information in the system, the decision rule is: if it occurs under cotton root then the pest is cutworm, else the pest is bollworm. As shown in Fig.8.



**Fig.8.** chart of pest diagnosis

In this case, we can resort to knowledge system development tools. The pest diagnosis chart can be generated in the knowledge editing area with the icons in



object toolbox, such as the decision node , the answer node  and the connecting line . The pest diagnosis chart in Fig.9 is similar to Fig.8.

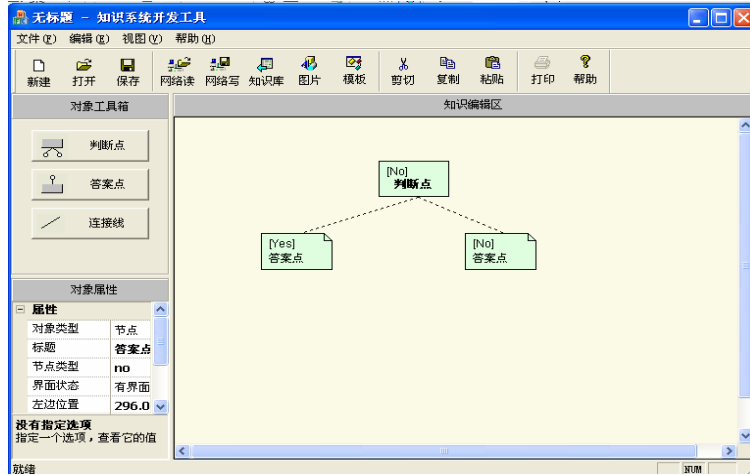



Fig.9. the picture of knowledge system development tools

The decision information provided to users to find the sites of pests in Fig.8 can be inputted and edited with knowledge system development tools. Clicking the decision node  in Fig.9 to open its interface editor, and then input the decision information for users to choose.

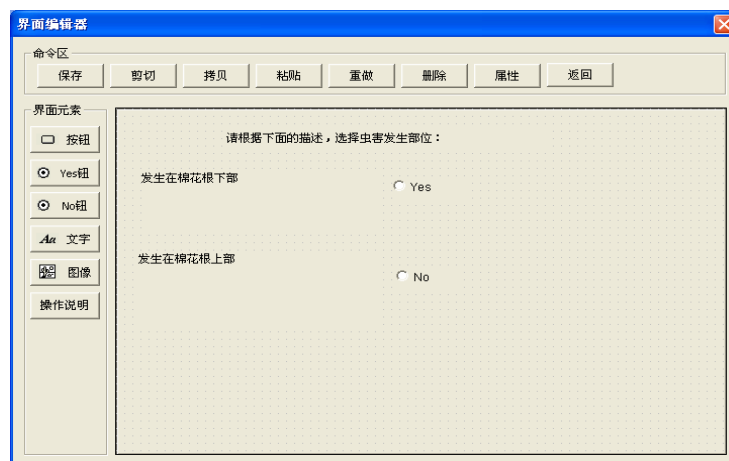


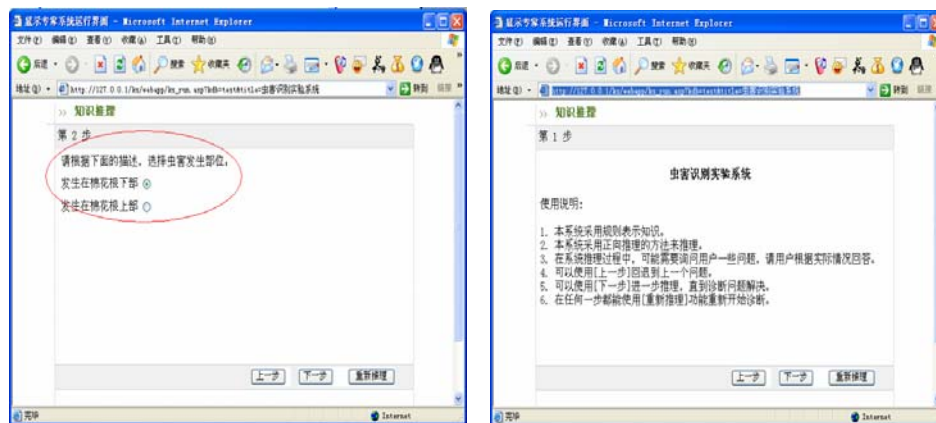


Fig.10. the node edit picture of knowledge system development tools

After editing the nodes and their interface information, saving the knowledge in server with the icon of  (net write) and named “test”. And then generate the rule-based information base that can be recognized by reasoning machine with the icon of  (knowledge base). Lastly, the knowledge base for “pest diagnosis experiment system” is founded and transmitted to the server.

The webpage of knowledge service system is shown in Fig.10 after inputting the address of “http://127.0.0.1/ks/webapp/ks\_run.asp?kdb=test&title= pest diagnosis experiment system” in URL of the browser. Click the button of “next” to



reason.

**Fig.11.** the system of knowledge service system

As thus, the pest diagnosis experiment system is finished.

### 3.2 Result for application

With the knowledge system development tools, we have developed several knowledge systems successfully, such as “the rice diseases diagnose system”, “the rice pests diagnose system”, “the cotton diseases and pests diagnose system”, “the apple tree diseases and pests diagnose system”, and so on. The application shows that the user-friendly interface makes the development tools easy to operate. The speed for knowledge generation using this knowledge system is about 35% faster than traditional way and the accuracy is also increased. Thus, experts of pants protection could generate a pests diagnose system knowledge system based network after simple training. This system is feasible and has a high practical value.

## 4. Conclusion and discussion

In this passage, we develop the visual pests diagnosis system development tools based on binary tree using techniques of computer visualization, binary tree, XML database and reasoning. This tool combined tree knowledge expression with rule-based reasoning machine, proposed the knowledge acquisition and the reasoning technique based on binary tree, and solved the problem of knowledge acquisition used for pests diagnosis with an expert system development tools--CLIPS. The reasoning machine with stateless continuous reasoning technology can improve the efficiency of knowledge acquisition and system reasoning. Application results show that the speed and accuracy of this system are greater than the traditional way. Thus, the development cycle of diseases and pests diagnose system is shortened, and promoted the information technology to be widely used in agriculture.

**Acknowledgements:** Our research is funded by the “863” project of research and application of field crop diseases intelligent diagnosis system (Number: 2007AA10Z237).

## References:

1. LIU Ying-tao. The design and realization of the front-end system of crop plant diseases and insect pests ground survey. *Agriculture Network Information*, 2006(7)
2. Wu He-ling. The introduction and application of Clips--an expert system tool. Beijing: Beijing Institute of Technology Publishing House, 1991
3. Wu He-ling. The introduction and application of Clips—a new expert system tool. 1992,5(3)
4. SHEN Ai-hua. Research and application of forward and backward reasoning based on binary taxonomic key. *Journal of Zhejiang University(Agriculture and Life Sciences)*. 2006,32(5)
5. MING Bo. Design and Realization of the Maize Disease Diagnosis System Based on Image Rules. *Journal of Maize Sciences*. 2009,17(6)
6. Yang Jian-hu. Expert System of Diagnosis and Controlling for Maize Disease, Pest and Weeds in Province. Master Thesis of Jilin Agricultural University. 2006.

7. Michael P. S. Brown, William Noble Grundy, David Lin.: Knowledge-based analysis of microarray gene expression data by using support vector machines. PNAS, vol. 97 no. 1, 262-267(2000).
8. Horowitz, E., Zorat, A.: The Binary Tree as an Interconnection Network: Applications to Multiprocessor Systems and VLSI. Computers, IEEE Transactions on. Volume: C-30 Issue:4, 247 – 253(1981) .