A MULTI-AGENT PROTOTYPE SYSTEM FOR HELPING MEDICAL DIAGNOSIS

QIAO YANG

A Multi-agent Prototype System for Helping Medical Diagnosis

by

@ Qiao Yang

A thesis submitted to the

School of Graduate Studies

In fulfillment of the

requirements for the degree of

Master of Computer Science

Supervisor: Dr. John Shieh

Department of Computer Science

Memorial University of Newfoundland

January 15th, 2008

St. John's Newfoundland

Abstract

Coordination and negotiation among agents are often necessary for medical diagnoses when a community of experts is called to be involved in a joint diagnosis and treatment for a patient. However, most of existing diagnostic systems are single-agent and rule-based, using probability theory or Bayesian networks. Fuzziness and uncertainty of concepts, facts and rules should also be considered to meet the needs of practical medical diagnoses, especially for Traditional Chinese Medicine.

In this thesis, a novel model of a multi-agent diagnosis helping system (MADHS) is given, where distributed knowledge-based systems are considered as cooperative agents in medical diagnoses. A final diagnosis compatible with both patient's anamnesis and existing medical principles can be reached through a joint decision-making procedure in this model. Fuzziness and uncertainty are incorporated into inference techniques to form the reasoning mechanism of the agents. The model is implemented to create a prototype system using Java, Java Agent Development Framework (JADE), Java Expert System Shell (JESS) and NRC FuzzyJ Toolkit. The prototype system has been tested by medical cases, especially those in Traditional Chinese Medicine (TCM).

It is anticipated that the proposed model and methodologies will be widely used in many applicative areas, such as multi-agent medical diagnosis, medical helping, and other automatic diagnosis and decision-making systems.

Key words: coordination and negotiation among agents, fuzziness and uncertainty in reasoning, joint diagnosis, medical diagnostic system.

Acknowledgments

First of all, I would like to thank my supervisor Dr. John Shieh, whose support and encouragement through all these years was extremely important to the successful completion of this thesis. I am also very grateful to the members of my supervisory committee.

And I could not thank enough my husband Xiaofei Li, my Aunt Ming Yang and Uncle Ziqi Gui, and other family members far away in China.

Finally, this thesis would not have been completed without the support from other wonderful faculties and staffs at Department of Computer Science, Memorial University of Newfoundland.

List of Tables

List of Figures

Table of Contents

1 Introduction

1.1 Overview

The aim of this research is to investigate, develop, design and implement a prototype system for modelling of coordination and negotiation among agents in a multi-agent medical diagnostic system. The final medical diagnoses should be achieved by combining the decisions made by several agents.

This chapter introduces the motivations for developing such a prototype system. This is followed by a brief introduction of the system model. In addition, a list of research objectives and the outline of the thesis will be provided at the end of this chapter.

1.2 Motivation

The goal of an autonomous agent can be satisfied more completely by gaining the collaboration of other agents. Thus, using cooperative multiple agents to improve the system's overall utility is an increasingly pervasive way of conceptualizing automatic diagnosis and decision-making systems.

However, most existing medical diagnostic systems use a single knowledge base, in other words, a single agent during their diagnosing. The web-based interactive medical decision support software EasyDiagnosis [20] is an example of them. Several multi-agent expert system models related to medical diagnosis have been proposed earlier, such as ALIAS [5] and Promedas [22]. But they are still under development. The establishment of a formal and complete prototype of a multi-agent medical diagnostic system remains a challenge.

1.3 A Multi-agent Diagnosis Model

The proposed multi-agent diagnosis helping system (MADHS) contains mainly four different kinds of agents: **Coordinator**, **Examiners**, **Specialists** and **Joint Decision Maker**, as shown by Figure 1. This designing approach is similar to the e-medicine system design presented by Tian and Tianfield [21]. But the Coordinator in MADHS plays several roles at the same time: *broker agent*, *administration agent*, *controller agent* and *interface agent*. The Joint Decision Maker can be considered as a kind of "*decision agent*", while the Examiners and the Specialists can be seen as "*diagnosis agents*" in Tian and Tianfield's design. The functional and communicational details of the above agents will be discussed further in Chapter 2 and Chapter 3.

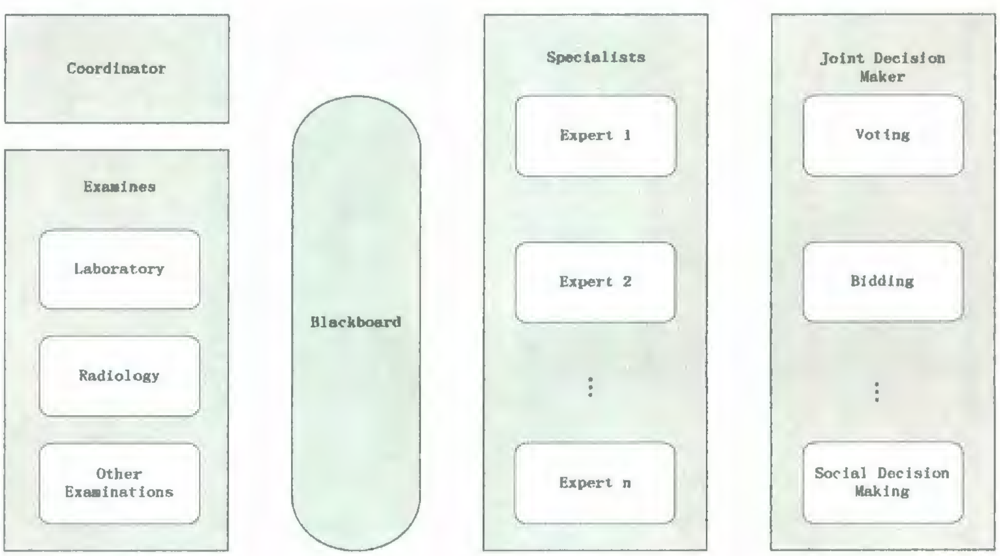


Figure 1 Agents in the proposed multi-agent medical diagnostic system

1.4 Research Objectives

The objectives of this research are:

1) To survey the current state of multi-agent medical diagnostic systems and to find their advantages and limitations;
2) To develop a novel model of coordination and negotiation among agents;
3) To develop methodologies and algorithms of our multi-agent medical diagnostic system;
4) To develop, test and evaluate the prototype system MADHS in a range of practical domains.

1.5 Outline of the Thesis

The remaining chapters of the thesis are organised as follows: Chapter 2 describes related work in the area of multi-agent medical diagnosis. The novel coordination and negotiation model is introduced in Chapter 3, together with other methodologies and algorithms used by it. The implementation of the multi-agent prototype system MADHS is discussed by Chapter 4. Later in Chapter 5, sixteen traditional Chinese medical records are used to test the functions of MADHS in the fields of Traditional Chinese Medicine and western medicine. Finally, results from Chapter 5 are evaluated and compared with the diagnoses from human experts in Chapter 6. The final conclusions and future research directions are discussed in the last Chapter.

2 Related works

2.1 Overview

This chapter will describe the current state of multi-agent medical diagnosis and discuss corresponding advantages and limitations of existing systems. It will also discuss current approaches concerning the aspects of coordination, negotiation, uncertainty management, decision-making, as well as agent-based modeling and simulation of medical processes.

2.2 Modeling Approaches and Architectures of Multi-agent Diagnostic system

In this section, a brief look is taken at the previous modeling approaches of multi-agent medical system. Different design approaches and architectures have been proposed, such as the multi-agent modeling approach of e-medicine mentioned earlier in Chapter 1 [21]. Tian and Tianfield have proposed the functional and non-functional requirements, design of agents and design of multi-agent society in e-medicine. They have divided the agents in a multi-agent society of an e-medicine system into four groups: agents for environment, control, implementation and interface. They have also identified the roles of agents, such as *interface agent*, *broker agent*, *doctor agent*, *administration agent*, *controller agent*, *department agent*, *diagnosis agent*, etc.

The MADHS presented in this thesis is also an e-diagnostic system (which can be seen as a variant of e-medicine). MADHS uses a far more simple and effective structure. The

Coordinator in MADHS can be considered as a combination of *broker agent*, *administration agent*, *controller agent* and *interface agent*. In other words, it plays several roles at the same time. Moreover, because the Coordinator in MADHS has its own knowledge bases for diagnosing, it can also be regarded as a *diagnosis agent*. In fact, each and every Specialist in MADHS can be appointed by the administrator (current user in control) to be the Coordinator of the system. Thus, considering the fixed roles played by the agents in Tian and Tianfield's e-medicine model, more flexibility is given in MADHS.

Ciampolini, Mello, Storari have described a multi-agent diagnostic system called ALIAS [5]. The inner structure of each agent includes two different modules: Abductive Reasoning Module for reasoning and Agent Behaviour Module to handle the multi-agent behaviour knowledge base in a collaborative or competitive way.

ALIAS multi-agent abductive logic framework does have limitations under real context of multi-agent diagnosis. For instance, a general medicine doctor could refer to four specialists; each specialized in a particular area: an osteologist, a neurologist, a cardiologist, and an angiologist. What will happen if there is more than one specialist in each area where the general doctor calls for joint decision-making? For example, what if there are 2 osteologists, 3 neurologists, 4 cardiologists and 5 angiologists? If so, these specialists could not work together in a competitive/collaborative way without differentiate between each of them. A general doctor in ALIAS treats every specialist equally, which is not a good idea among multiple agents, since the qualities of agents may vary. The ALIAS framework has not considered any method to differentiate among participating specialists.

Another aspect of research closely related to the modeling of multi-agent medical diagnostic system is agent-based simulation of hospitals. It is usually the first topic researchers will come up with, when they consider designing a multi-agent medical system. Therefore it has been studied for years. Sibbel and Urban have suggested the simulation of the treatment process in a German hospital [18]. However, most of the existing multi-agent medical diagnostic systems did not follow the same systematic course of treatment as in a real hospital. Therefore, MADHS only involves two steps "Patient Admission" and "Medical History and Preoperative Examinations" of their seven-step simulation (other five steps are "Operation", "Intensive Care", "Intermediate Care", "Postoperative Examinations and Care" and "Patient Discharge"). During these two steps, multiple experts make their diagnoses according to the medical history and examination results of admitted patients.

Unlike other traditional modeling approaches in the hospital domain, Sibbel and Urban's modeling of hospitals did not ignore the importance of human decision-making and behaviors in such systems. But they have not gone any further on some crucial aspects such as joint decision-making, coordination and negotiation methods among the multiple agents, which will be emphasized by this thesis.

Klügl, Oechslein, Puppe and Kirn have also proposed the formal simulation of a hospital [10]. They mainly focused on the scheduling and planning methods for the care of patients. The resulting model can improve the time and resource management problems in a hospital. On the other hand, they also have not considered decision-making and negotiation processes among multiple agents.

Besides, most existing medical diagnostic systems are either single-agent, or based on Bayesian theory (calculating probabilities). Researches have shown that human experts and ordinary users did it very poorly when they were required to give the precise probabilities of the facts used by the rule bases. Calculating uncertainties instead of probabilities in diagnostic systems is becoming a tendency.

2.3 Coordination and Negotiation

Coordination is the process through which individual agents in the same community try to act more coherently. Nwana, Lee and Jennings have classified current coordination techniques into four broad categories: *organizational structuring*, *contracting*, *multi-agent planning* and *negotiation* [11]. The coordination process in MADHS is a combination of organizational structuring and blackboard negotiation. Details of the coordination process will be explained in Chapter 3.

The aim of negotiation is to solve conflicts among agents through interactions. The agents in the same society try to reach a common decision by negotiating with each other.

Blackboard negotiation was first suggested by Werkman's in his paper *Knowledge-based model of negotiation using shareable perspectives* [23]. The negotiation process here in MADHS follows a similar but revised three-phase cycle proposed by Werkman. At the beginning of the first phase, the Coordinator sends an initial plan (or initial proposal) to the JointDecisionMaker. Depending on the chosen joint decision-making algorithm, the JointDecisionMaker modifies the initial plan, adds joint decision-making related information into it, and sends it back to the Coordinator. Then the Coordinator will announce the revised plan to all the registered Specialists who are qualified to participate in the joint diagnosis. The second phase involves individual Specialist evaluating the proposal received from the Coordinator. It may refuse its own task (but generate no counter proposal), or simply reply with a "not-understood" message, or accept its task happily. At phase three, the diagnoses made by available Specialists will be submitted to the blackboard, and reviewed by the JointDecisionMaker. The JointDecisionMaker is responsible to use one of the joint decision-making algorithms and the information stored in the blackboard to produce the final diagnosis.

In Werkman's DFI model, an "arbitrator" helps those agents in need to solve their mutual conflicts through communication, scheduling, setting time limits or other possible techniques. Currently in MADHS, no such "arbitrator" agent or any proper scheduler is implemented. In order to facilitate the negotiation among agents, and to avoid potential conflicts or bottle-necks, one arbitrator (or agent playing the similar role) must be added to MADHS prototype sooner or later.

2.4 Joint Decision-making

Jennings continued working on multi-agent coordination with Hogg, this time, on variable sociability in agent-based decision-making [9].

They proposed a new social decision making framework. This approach incorporates an element of social consideration into each agent's individual decision-making function, and finds a decision of optimistic social benefit to the society.

Jennings and Hogg have developed a socially aware framework for decision making in a multi-agent context. However, their research has not considered the special features and requirements of a diagnostic system. They focus on socially rational decision-making systems in which the available time or resources are limited. When the special requirements of diagnostic system are considered, traditional social welfare functions or utility functions may not be enough to represent the benefit of the entire agent society. In Chapter 3, the social decision-making in the context of multi-agent diagnostic system is discussed in details.

2.5 Certainty Factors and Fuzzy CLIPS

Certainty Factors were first suggested by Shortliffe and Buchanan in their famous medical diagnostic system MYCIN [17]. Unlike probability, which states the likelihood of occurrence, a CF between [-1, +1] represents the extent to which a statement is considered reliable. +1 indicates certainty; 0 indicates no positive or negative belief; -1 indicates negative evidence against this rule or fact. For example, a domain expert (doctor) may state: "if a patient already has abdominal pain, there is a possibility of 80% that he or she also has colitis", and assigns a CF of 0.95 to this rule at the same time.

In MADHS, because a decision of Boolean value (True or False) is given whenever a diagnosis is made, the range of certainty factors becomes [0, 1]. Details of the CF calculation will be explained in Section 3.4.4.

The knowledge bases in most expert systems are constructed by rules. The calculation of certainty factors in chaining rules has already been discussed by Fuzzy CLIPS [12].

In Fuzzy CLIPS, rules are classified to Simple Rules and Complex Rules. Simple Rules are the rules with one antecedent and one consequent. Simple Rules include:

- CRISP Simple Rule

  The word "crisp" is used to describe the facts and rules without fuzzy variables. The equation for CF calculation in this kind of rule is:

  $$CFc = CF_r * CF_f$$

  CF_c above represents the certainty factor of the conclusion; CF_r is the certainty factor of this simple rule; CF_f is the certainty factor of the input crisp fact.

- FUZZY_CRISP Simple Rule

  The equation for the CF calculation is:

  $$CFc = CF_r * CF_f * S$$

  While CF_c , CF_r and CF_f above have the meaning as in the CRISP_Simple rule equation, *S* is a measure of the Similarity between the fuzzy set in the antecedent and the input matching fuzzy set.

- FUZZY_FUZZY Simple Rule

  The fuzzy set asserted by the consequent is determined by the equation below:

$$\mu_{F_c'}(v)=\min(\max(\min(\mu_{F_a'}(u),\mu_{F_a}(u))),\mu_{F_c}(v))$$

In the equation above, F_a represents the fuzzy set determined by the antecedent; F_a' represents the fuzzy set of the input matching fact for the antecedent. F_c is the fuzzy set of the consequent. $\mu_{F_a}(u)$, $\mu_{F_a'}(u)$ and $\mu_{F_c}(v)$ are the corresponding membership functions of these fuzzy sets, where u and v represent the horizontal axes of the membership functions. The whole equation means to get the intersection of input membership function $\mu_{F_a^{'}}(u)$ and antecedent membership function $\mu_{F_a}(u)$ first, and then use the maximum value of the intersection to scale the membership function of the consequent, $\mu_{F_c}(v)$.

Except for that, CF of the conclusion is still calculated by $CFc = CF_r * CF_f$.

Complex Rules include:

- Multiple Consequents

  The rules with multiple consequents are treated as multiple rules with a single consequent.

- Multiple Antecedents

  This kind of rule can be treated as a combination of multiple simple rules with one antecedent. The fact asserted by the consequent at the end of a rule firing can be represented by the equation below:

$$F_c^{'} = F_{c1}^{'} \cap F_{c2}^{'} \cap, \ldots, \cap F_{cn}^{'}$$

  Where $F_{cn}^{'}$ represents the *n*th input fact for the *n*th antecedent, and $F_c^{'}$ represents the fact asserted by the consequent into the knowledge base at the end of rule firing.

  The CF of the consequent is calculated according to MYCIN's model:

$$CF_c = \min(CF_{f1}^{'}, CF_{f2}^{'}, \ldots, CF_{fn}^{'}) * CF_r$$

where CF_c represents the CF of the consequent. CF'_{fn} represents the CF of the *n*th simple rule: if A_n, then *C*, given the input matching fact A'_n. CF_r is the CF of the rule.

In MADHS, similar equations are used in the small Sub Rules. But MADHS is designed especially for Chinese and western medical diagnoses. The rules usually have more medical related features. Thus, the CF calculation in these rules is very different with the method using in Fuzzy CLIPS or MYCIN. Details will be explained in Section 3.4.3.

2.6 Multi-agent Development Toolkits

Shakshuki and Jun have compared several existing multi-agent development toolkits in their work [16], such as JADE, Zeus and Jack.

JADE (Java Agent Development Framework) is a software framework implemented in Java language. It is developed by the research institute of Telecom Italia. Through a set of graphical tools, it simplifies the implementation of multi-agent systems. Multi-agent platforms designed by JADE can be distributed across machines, or using different operating systems. JADE is completely implemented in Java language and the minimal system requirement is the version 1.4 of JAVA (the run time environment or the JDK). The latest version of JADE is JADE 3.5 released on June 25th, 2007.

Zeus toolkit is developed by British Telecom Lab. Zeus provides a graphical environment to build distributed agent systems. It includes a rule engine, planner and visualization tools. The current version that can be downloaded for free is Version 2.0 patch 2 released on January 10^{th}, 2006.

Jack is not open-source and not available for free use to developers. It is a commercial agent oriented development environment of Agent Oriented Software Group. A trial version of JACK Intelligent Agents Framework Version 5.2 can be downloaded from the homepage of the company. The GUI of Jack allows the developers to modify the agent's views, belief sets, capabilities and plans. Unfortunately, Jack does not make use of any pre-existing standard agent communication language. And it is not free for researchers. Thus, Jack was not chosen to be the development toolkit in this thesis.

According to the tests conducted by Shakshuki and Jun, JADE provides the best performance on the message transport system when compared to Jack or Zeus. After thorough comparison of these three development tools, JADE is chosen to be the agent development framework for MADHS.

In a Clinical Practice Guidelines (CPG) implementation for asthma treatment [19], the researchers use Protege-2000 for ontology editing, JESS as the inference engine and NRC FuzzyJ Toolkit as the add-in for JESS that provides fuzzy logic capabilities. Inspired by their work, the programmer of MADHS chooses JESS and FuzzyJ for the designing of the knowledge bases of each agent. JESS 7.1a1 is the latest version available for a free trial. FuzzyJ version 1.10a is also available for free downloading.

2.7 *Other Related Works*

This section briefly mentions some of the other research works related to the development of MADHS.

2.7.1 FIPA ACL and Interaction Protocols

As mentioned in Section 2.5, Java Agent Development Framework (JADE) is chosen to be the development tool of MADHS. JADE uses Agent Communication Language (ACL) messages defined by Foundation for Intelligent Physical Agents (FIPA) to communicate among agents. Using the ACL Message GUI provided by JADE, the users can choose appropriate message types they want to use, such as REQUEST, AGREE and INFORM defined by FIPA Communicative Act Library Specification (SC00037J). They can also choose among the interaction protocols defined by FIPA, including Request (SC00026H), Request When (SC00028H), Contract Net (SC00029H), Broking (SC00033H) and Recruiting (SC00034H).

In MADHS, because most of the users are not supposed to be familiar with any communication languages or protocols, the whole communication procedure, including the protocols used, is already combined into the *initiator* agents and the corresponding *receiver* agents. Thus it will be completely invisible to the users.

2.7.2 Bidding and Voting

Flores-Méndez described the complete model of agent conversation for bidding in his work [7]. Bidding can be chosen as a joint decision-making method, giving the prerequisite that there is something to bid for among multiple agents. Despite the fact that bidding may not be suitable for some complex joint decision-making problems, it can be used as one of the options for certain joint decision-making problems appearing in MADHS. For example, prices of the treatments are not given any consideration in MADHS currently. If the users want to find out the best treatment instead of the most reliable diagnosis of their diseases, then the prices ought to be considered. Under this circumstance, bidding will be a practicable way to get a best price.

Voting is another possible solution for joint decision-making in MADHS. Different voting systems use different types of vote: *single vote*, *multiple vote*, *ranked vote* or *range vote* (*scored vote*).

The joint decision-making method currently used in MADHS is a variation of *range voting*. In original range voting, the score of a candidate (possible disease) would be the sum or average rating from the voters (Specialists) who did rate this candidate (diagnosed this disease). In MADHS, the joint decision-making method can be described as a kind of *weighted range voting*. Because the *diagnosis agents* in MADHS have different qualities, they deserve to have different weights attached to their opinions when the society is making a joint decision.

2.7.3 Knowledge Representation

Belief net (Bayesian net) has contributed greatly in the field of AI [15]. The main bottleneck of the technique is the inference in large networks. Although complexity in singly-connected networks is polytime, it is shown to be NP hard for more general multiply connected networks. Many tools are available for Belief net users, such as MATLAB and Microsoft MSBNx.

Belief net could be one of the possible representations for knowledge bases of domain experts in multi-agent medical diagnostic system.

2.8 Summary

This chapter has briefly introduced the existing multi-agent medical diagnosis models, related methodologies and toolkits. Corresponding advantages and limitations of every approach are also discussed. In the next chapter, a novel model of multi-agent medical diagnostic system will be presented.

3 The Model of Multi-agent Diagnosis Helping System

3.1 Overview

This chapter describes the novel model of a multi-agent diagnosis helping system.

After the overview, the second section of this chapter introduces roles of agents in the diagnostic system. The subsequent section describes the communication among agents. It is followed by explanations of the coordination and negotiation going on in MADHS. The fourth section discusses the methodologies used in the building process of this fuzzy expert system. The last section introduces a range of joint decision-making methods supported by MADHS. Also, a brief summary is given at the end of this chapter.

3.2 Model of MADHS

In this section, a novel model of Multi-agent Diagnosis Helping System (MADHS) is introduced. Figure 2 on the next page gives a general view of the system model, including the types of agents, and the directions of communications among agents.

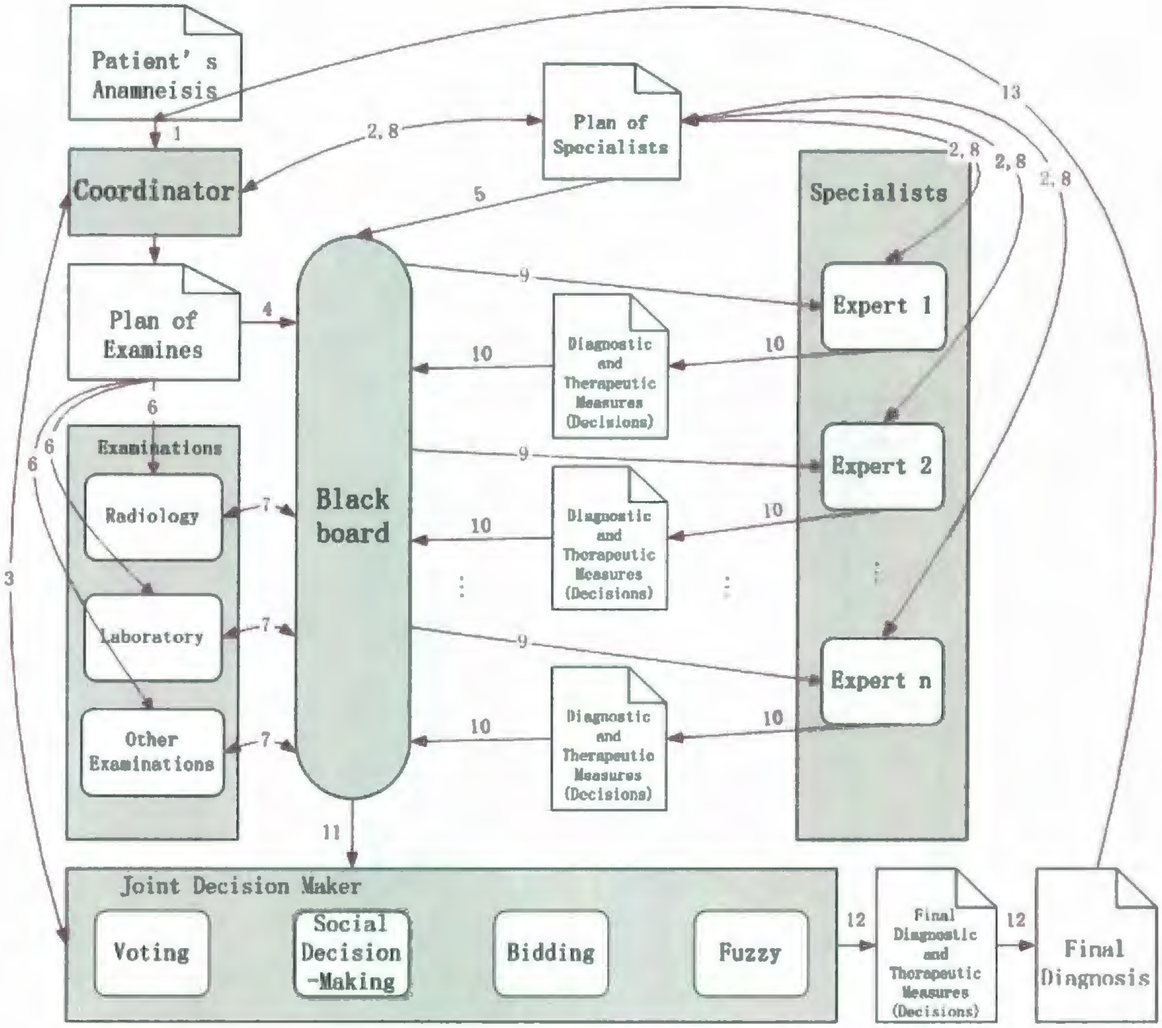


Figure 2 The novel model of the Multi-agent Diagnosis Helping System

3.2.1 Agents

Although JADE has already provided a Java class named Agent as a common base for agents, MADHS still need various user-defined Java classes. The types of agent classes appeared in MADHS are as follows:

1) Coordinator:

Coordinator is an agent class designed especially for the following tasks:

a) To determinate an initial plan of diagnosis or treatment according to the patient's anamnesis.

   That means the Coordinator must diagnose the patient by itself, and then asks other agents to form a team of Specialists and Examiners.

   When the Coordinator diagnoses the patient at first, the knowledge base used by it can be simpler than that of other Specialists'. The questionnaires handed out by the Coordinator can ask more basic, more general questions about the patient's symptoms. Then the Coordinator will use the initial diagnosing results to narrow down the range of diseases which the patient probably has, and forms a team of Specialists and Examiners for this patient.

b) To discuss with the Specialists about whether or not they want to participate in the joint diagnosis, their opinions to the initial plans and their available schedules.

c) To communicate with the Joint Decision Maker. Once the Coordinator has decided the initial plan and schedules (of examinations and Specialists), it informs the Joint Decision Maker. The Joint Decision Maker chooses the proper decision-making method according to the initial plan. It also decides the corresponding parameter values (such as the weight added to the diagnostic result of each agent) according to the chosen decision-making method.

d) To assign the sub tasks to domain experts (Examiners and Specialists) according to the plan revised by the Joint Decision Maker.

In fact, a class Coordinator can be further divided into two separate classes. One part is just a common Specialist like others, doing the initial diagnosis described in a) above. The other part is a real "Coordinator", who manages the control flow (finite state machine) and the communication paths, doing the remaining responsibilities in a), b), c) and d). Separating these two parts has one obvious advantage: the "Coordinator" part can be dynamically assigned to any participating Specialist before the joint diagnosis begins. Using the GUI provided by MADHS, a patient who is diagnosing himself/herself can appoint one of the available Specialists to be the Coordinator, if he/she thinks the default Specialist doesn't have enough authority. A participating Specialist can also initiate a negotiation by sending messages to other experts before the joint diagnosis to vote him/herself or another Specialist to be the Coordinator of the team.

However, the model of MADHS is not centralized. The Coordinator only controls the global coordination and the top-level communication paths. Because of the hierarchical structure of the agent society, other independent agents (Specialists and Examiners/Lab Agents) in the system can initiate their own partial coordination, negotiation, and communication. For example, if one of the Specialists needs more lab tests that are not listed in the "Plan of Examinations" to be done during its diagnosing procedure, it can send extra requests to the lab agents, automatically or manually (by the user), following its own "Behaviour" classes, and then get extra lab results directly from them. No central control from the Coordinator appears in this situation.

2) Examinations:

The Lab Agents will conduct the examinations for the Specialists as they required. After the Coordinator has decided the plan of examinations according to the patient's anamnesis, the Lab Agents of the system are motivated to do the examinations and send the results back to the blackboard. But as mentioned in **1)**, other participating Specialists can ask the Lab Agents to do extra lab tests directly.

3) Specialists:

Specialists are the agents who contribute to the diagnosis or treatment in special domains. The coordination among those Specialists could be either competitive or collaborative, depending on which kind of decision-making method is being used in the Joint Decision Maker.

4) Joint Decision Maker:

The Joint Decision Maker (JDM) is one of the complicated parts of our model. The inner structure of the JDM will be discussed later in Section 3.5.

The decision-making method applied in this agent should be carefully chosen. The users of the anticipated system should be able to choose among Voting, Social Decision-making and Bidding, or use their own decision-making method.

a) Voting:

Voting can be used as one of the joint decision-making methods. For example, when all the specialists have similar qualifications, voting will be a better choice than the other decision-making methods.

Currently, the joint decision-making method applied in the Joint Decision Maker can be called *weighted range voting*. Details of this method will be explained in Section 3.5.

b) Social Decision-making:

Social Decision-making can also be a good choice of decision-making methods. However, the users have to figure out what is the optimistic social benefit for the society of specialists and how to measure it before using this method.

c) Bidding:

When the aim of the joint decision-making is to select the best choice among the results provided by the participating agents, bidding can be chosen as the appropriate joint decision-making method. For instance, when the user wants to find out the best price of treatment instead of the most reliable diagnosis, then bidding is the best method to solve the problem.

After the making of one joint decision, the Joint Decision Maker should be able to judge whether there are any more joint decisions to make and whether to go back to the Coordinator again and do some iterations of the whole procedure.

5) **Blackboard**

The blackboard of MADHS is a hierarchically organized global memory (or database) that stores the following Meta data, and information:

a) plans (schedules) of Specialists and examinations

b) examination results from Lab Agents

c) individual diagnoses generated by Specialists

The details of the blackboard structure will be introduced in Section 3.3 while discussing the coordination and negotiation techniques.

6) **Existing JADE Agents**

Any multi-agent system developed by JADE will have a DF (Directory Facilitator) agent, an AMS (Agent Management System) agent and a RMA (Remote Monitoring Agent) agent by default [2]. The AMS provides the naming service and guarantees that each agent in the platform has a unique name; The DF agent provides the yellow book of the registered services; The RMA appears when the user starts the GUI of JADE. It is an agent management tool, in which a user can easily manage and send messages to all the agents in the platform.

3.2.2 Communications among Agents

In order to explain the communication process among the agents, step numbers are added to the communication directions in Figure 2:

1) **Step 1:**

The Coordinator diagnoses the patient first, and forms a team of Specialists according to its diagnosis.

2) **Step 2:**

The Coordinator initiates a negotiation, sends out REQUEST messages to each and every team member to ask if they will be available within a certain time limit. The Specialists being asked should check their own schedules. If a Specialist is available to diagnose the current patient within the time limit, it sends its available time slots back to the Coordinator; if not, it refuses the request from the Coordinator.

3) **Step 3:**

The Coordinator plans the actions of the Specialists according to their schedules. Thus, it forms a global initial plan. Then it sends the initial plan to the Joint Decision Maker. The Joint Decision Maker modifies the initial plan, adding parameter values according to the chosen joint decision-making method, like the weights in the Voting process. After that, it sends the revised plan back to the Coordinator.

4) **Step 4:**

The Coordinator stores Plan (Schedules) of Examinations to the blackboard.

5) **Step 5:**

The Coordinator stores Plan (Schedules) of Specialists to the blackboard.

6) **Step 6:**

The Coordinator initiates cooperation, sends out REQUEST messages to the Lab Agents according to the Plan of Examinations. A Lab Agent being asked will answer the diagnosis request with:

a) A **failure**, if it fails to fill the request.

b) An **inform-done**, if it successfully completes the request.

c) An **inform-result**, if it successfully completes the request and notifies the initiator (Coordinator) of the results.

7) **Step 7:**

The Lab Agents send the lab results to the blackboard. If they failed to receive the requests from the Coordinator earlier, they can check the Plan of Examinations stored in the blackboard to find out what actions they should take.

8) **Step 8:**

The Coordinator initiates another conversation, sends out REQUEST messages to the team members according to the Plan of Specialists. A Specialist being asked will answer the diagnosis request with:

a) A **failure**, if it fails to fill the request.

b) An **inform-done**, if it successfully completes the request.

c) An **inform-result**, if it successfully completes the request and notifies the initiator (Coordinator) of the results.

9) **Step 9:**

The Specialists get the lab results they needed from the blackboard.

If they failed to receive the diagnosis requests from the Coordinator earlier, they can check the Plan of Specialists stored in the blackboard to find out what actions they should take.

10) Step 10:

The Specialists send the diagnoses and corresponding therapies to the blackboard.

11) Step 11:

A finite state machine in the Coordinator arranges the next behaviour to be activated, as will be described in Section 4.3. In Step 11, under the request of the Coordinator, the Joint Decision Maker gets the diagnoses produced by the Specialists from the blackboard, uses the chosen joint decision-making method on them, and forms a final diagnosis.

12) Step 12:

The Joint Decision Maker demonstrates the final diagnosis for the current patient to the user.

13) Step 13:

The Coordinator clears the blackboard and the facts temporally stored in the knowledge bases. Then it will automatically start over again from Step 1. If a user is not satisfied with the final diagnosis, he/she can choose to diagnose the same patient again. Because the available team members and their corresponding knowledge bases may vary at that time, and a user could adjust the CFs given to the facts (patient information), the result of the next joint diagnosis may be different too.

3.3 Coordination, Cooperation and Negotiation in MADHS

The definition of coordination is described in Nwana, Lee and Jennings work as follows [11]:

"Coordination is a process in which agents engage in order to ensure a community of individual agents acts in a coherent manner."

Because the participating agents distribute all over the platform, coordination is extremely important in MADHS. The registered experts have different medical resources, capabilities and limitations. For instance, the Specialists can diagnose the patient according to their own knowledge bases. But they also need to ask the Examiners (Lab Agents) to perform the related laboratory tests and send back the results.

The coordination strategy *organizational structuring* is modified and used in MADHS. The roles and responsibilities of the participating agents are pre-defined to form a hierarchical organization. The user-defined agent classes (and their corresponding roles) have been explained earlier in Section 3.2.1, while introducing the system model. The inner structure of the Coordinator and the Blackboard will be illustrated in Figure 3 of this section.

Meta-level Information Exchanges, like partial global plans (PGPs), is intended to be applied in MADHS as the coordination method in the future. If this coordination method is chosen, one expert's diagnosis will provide useful predictive information for other experts (acquaintances) in the same community.

To achieve coordination, cooperation is not a necessary premise. It may even cause incoherent behaviours [11]. But in MADHS, distributed experts have to cooperate through communication in order to coordinate. As have been mentioned earlier, the Specialists can not conduct the laboratory tests by themselves. Besides devising a Plan of

Examinations at the beginning of the diagnosis, the second (and current) solution offered by the programmer of MADHS is to allow the Specialists automatically request help from Lab Agents whenever a laboratory test is needed. This is a typical way of cooperation.

Negotiation is the process where final agreement is achieved by communication and mutual selection. Blackboard negotiation is used as a coordination method in MADHS. Similar to Werkman's classic blackboard architecture (introduced in Section 2.3), the multi-partitioned blackboard in MADHS is the public information source of the agent community, as shown in Figure 3 below:

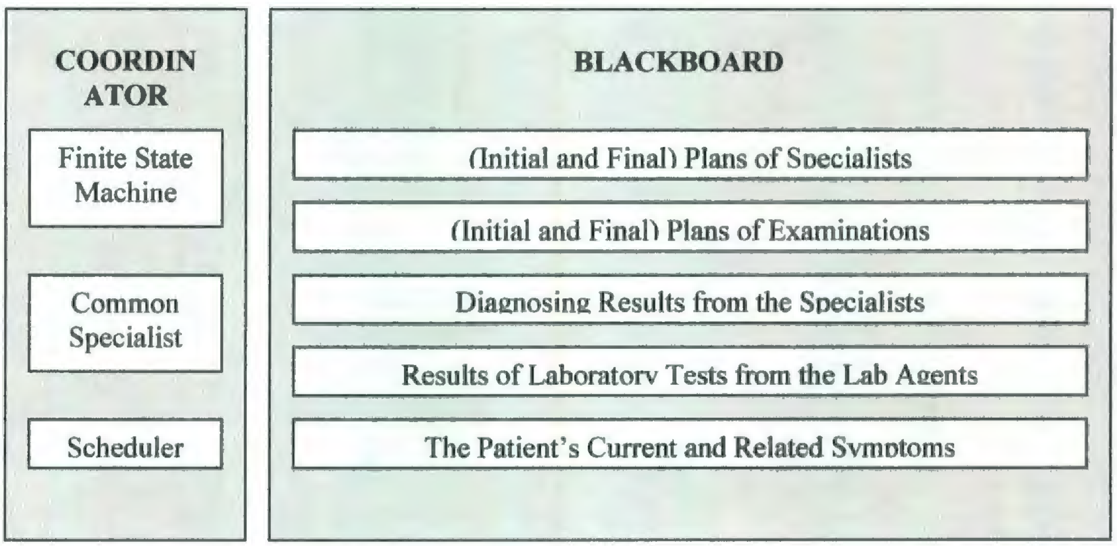


Figure 3 The Coordinator and the blackboard negotiation in MADHS

Figure 3 clearly illustrates the inner structure of the Coordinator. It contains three main components:

1) A finite state machine

The finite state machine in the Coordinator controls the intra-platform communication paths. The structure of this finite state machine will be described in Section 4.3, where the implementation of the intra- and inter-platform communication is discussed in details.

2) A common Specialist

The common Specialist in the Coordinator has the rule bases for diagnosing like other Specialists. Because it is only responsible for the initial diagnosis of a patient, its rule bases can be simpler. Actually the structure of the Coordinator is very flexible. The components 1) and 3) of the Coordinator can be removed and attached to any Specialist in the platform. Then that Specialist becomes the new Coordinator of the agent society.

3) A Scheduler

The Scheduler within the Coordinator schedules the readings from and writings to the blackboard.

Figure 3 also shows the partitions (sub systems) in the blackboard, including the patient's current and related symptoms; initial proposals from the Coordinator; final plans negotiated by the Specialists and revised by the Joint Decision Maker; shared diagnosing results and shared laboratory results. The global or partial knowledge mentioned above will provide sufficient information for the agent community to support their further decisions.

3.4 Building a Fuzzy Expert System

MADHS is also a fuzzy expert system. In order to build this prototype system, the designers should first specify the problems, define linguistic variables and determine corresponding fuzzy sets, as described in 3.4.2. Then they should elicit and encode fuzzy rules, as described in 3.4.4. The equations of certainty factor calculation in MADHS are introduced in Section 3.4.3. The knowledge representation and inference technique used in the system are introduced in Section 3.4.4.

3.4.1 Expert System Development Team

An expert system development team consists of: the knowledge engineers, the domain experts, the project managers, the programmers and the end-users.

The domain experts play a very important part in obtaining expertise. However, because the time and resources limit, the knowledge engineer of MADHS could not interview real domain experts. The expert knowledge used in this prototype system is mainly elicited from Chinese and western medical manuals of clinical diagnosis. MADHS is a small prototype system. The author of this thesis is responsible for the knowledge engineering designing, encoding and testing of the prototype system.

The end-users of MADHS can be the patients who want to diagnose themselves in case of emergency, the doctors who want to get some advices from other specialists, and the specialists who want to participate in the joint diagnoses of certain patients. The developer of MADHS must consider their different requirements when designing the prototype system, especially when designing the graphical user interface (GUI).

3.4.2 Fuzzy Logic in Knowledge Acquisition

When a patient comes to see a doctor, he/she usually describes his/her symptoms and medical history using certain fuzzy concepts: "I ate *little* recently", "I had a *high* fever yesterday", "I have had gastritis for a *long* time". Those fuzzy concepts like "little", "high" or "long" have no clear boundaries. For some people, two months can be called a long time; while for others, a long time means more than one year.

In this section, further studies are conducted on the inevitable fuzziness in MADHS, especially in the traditional Chinese medical diagnoses part. Fuzziness and uncertainty are two distinct concepts employed in fuzzy expert systems. The next section will discuss the measurement of uncertainty in MADHS.

According to Zadeh's research on fuzzy set theory [25], a Fuzzy Variable (Linguistic Variable) defines the components used to describe a fuzzy concept. It consists of the name of the variable, units of the variable, the universe of discourse (UOD) and fuzzy terms. Each fuzzy term consists of a term name and a Fuzzy Set.

A Fuzzy Linguistic Expression is defined by fuzzy variables, fuzzy modifiers and operators. For example, the linguistic expression "very little or normal food intake" consists of the term *little* and *normal*, the fuzzy modifier *very*, and the fuzzy set union operator *or*.

The designer of MADHS is also responsible for the knowledge engineering process. Knowledge Acquisition is the first stage of Knowledge Management. Domain knowledge is elicited from different Chinese and western medical manuals for the encoding of corresponding knowledge bases. Fuzzy concepts represented by natural language in the manuals and patient anamneses are organized into fuzzy linguistic expressions by the knowledge engineer. After this knowledge engineering process, the users of MADHS are able to choose between fuzzy words and numbers when they want to input patient anamneses.

The knowledge engineering process is shown by an example. In traditional Chinese medical diagnosis, "poor appetite" is a symptom of the disease "Deficiency of Spleen-Qi". The formal linguistic expression of "poor appetite" is "little food intake" in natural language.

According to the medical information for food intake published in Merck Manual [1]:

1) Protein, carbohydrate and fat provide 90% dry weight of the diet, and 100% energy of the diet;

2) There are 4 calories in a gram of protein;

   There are 4 calories in a gram of carbohydrate;

   There are 9 calories in a gram of fat.

3) Ordinary calories intake: 1000 to 4000 calories per day for a person.

   Daily calorie intake of various groups of people is shown in Table 1.

Table 1 Daily calorie intake of different groups

Class No.	*Group of People*	*Age*	*Daily Calorie Intake (Calorie)*
Class 1	sedentary women		1600
	young children	1-8	
	older adults	>=50	
Class 2	older children	8-13	2000
	active adult women		
	sedentary men		
Class 3	active adolescent boy	13-18	2400
	young men		

4) In daily calorie intake, 15% comes from proteins, 55% comes from carbohydrate, and 30% comes from fat.

According to the statistics listed in 2) 3) and 4), the daily protein, carbohydrate and fat intakes of different groups can be calculated by the following equations: (This step is done by the knowledge engineer of MADHS.)

$$amount\ of\ protein = \frac{amount\ of\ calories \times 15\%}{4} \quad (g)$$

$$amount\ of\ carbohydrate = \frac{amount\ of\ calories \times 55\%}{4} \quad (g)$$

$$amount\ of\ fat = \frac{amount\ of\ calories \times 30\%}{9} \quad (g)$$

The minimum of daily food intakes (dry weight) are calculated by adding these three food intakes together. According to 1), the normal daily food intakes (dry weight) are calculated by the equation below:

$$daily\ food\ amount\ (normal) = \frac{daily\ food\ amount\ (\min)}{90\%}$$

The results of the calculations are shown in Table 2 Daily food intakes of different groups:

Table 2 Daily food intakes of different groups

Class No.	*Daily Calorie Intake (Calorie)*	*Daily Protein Intake (g)*	*Daily Carbohydrate Intake (g)*	*Daily Fat Intake (g)*	*Daily Food Intakes :Min (g)*	*Daily Food Intakes: Normal (g)*
Class 1	1600	60	220	53.3	333.3	370
Class 2	2000	75	275	66.7	416.7	463
Class 3	2400	90	330	80	500	555.6

Finally, the knowledge listed from 1) to 4) and above is elicited and represented properly to form the membership functions for the fuzzy variable "food intake", as shown in Figure 4.

The knowledge engineer of MADHS will encode the membership functions using classes and methods provided by the Fuzzy Jess (FuzzyJ) toolkit. ZFuzzySet, PIFuzzySet and SFuzzySet are three FuzzySet classes defined by FuzzyJ. The membership function of ZFuzzySet usually has a "Z" shape curve with a 1 at the left edge and 0 at the right edge. PIFuzzySet is used to build a "bell" shape FuzzySet with a 0 at the left and right edges of the curve and a 1 at the middle. The membership function of SFuzzySet has an

"S" shape curve with a 0 at the left edge and 1 at the right edge. The complete definitions of ZFuzzySet, ZFunction, SFuzzySet, SFunction and PIFuzzySet can be easily found in the on-line tutorial of NRC FuzzyJ Toolkit.

For patients of Class 1, the definition of foodAmt written by Fuzzy Jess is:

```
foodAmt = new FuzzyVariable("foodAmount", 0.0, 1000.0, "g/day");
foodAmt.addTerm("little", new ZFuzzySet(0.0,333.3));
foodAmt.addTerm("normal", new PIFuzzySet(370.0,47.0));
foodAmt.addTerm("much", new SFuzzySet(416.7,1000.0));
```

For patients of Class 2, the definition of foodAmt written by Fuzzy Jess is:

```
foodAmt = new FuzzyVariable("foodAmount", 0.0, 2000.0, "g/day");
foodAmt.addTerm("little", new ZFuzzySet(0.0,416.7));
foodAmt.addTerm("normal", new PIFuzzySet(463.0,47.5));
foodAmt.addTerm("much", new SFuzzySet(500.0,2000.0));
```

For patients of Class 3, the definition of foodAmt written by Fuzzy Jess is:

```
foodAmt = new FuzzyVariable("foodAmount", 0.0, 3000.0,"g/day");
foodAmt.addTerm("little", new ZFuzzySet(0.0,500.0));
foodAmt.addTerm("normal", new PIFuzzySet(555.6,56.0));
foodAmt.addTerm("much", new SFuzzySet(610.0,3000.0));
```

For example, for patients of Class 2, Fuzzy Jess code “foodAmt.addTerm("normal", new PIFuzzySet(463.0,47.5));” means adding a fuzzy term to the fuzzy variable foodAmt. The membership function of this fuzzy term is a PIFuzzySet, whose midpoint is 463.0, and the curve width is 47.5. 463.0 and 47.5 are directly elicited from the data shown in Table 2.

The definitions above are illustrated by

Figure 4:

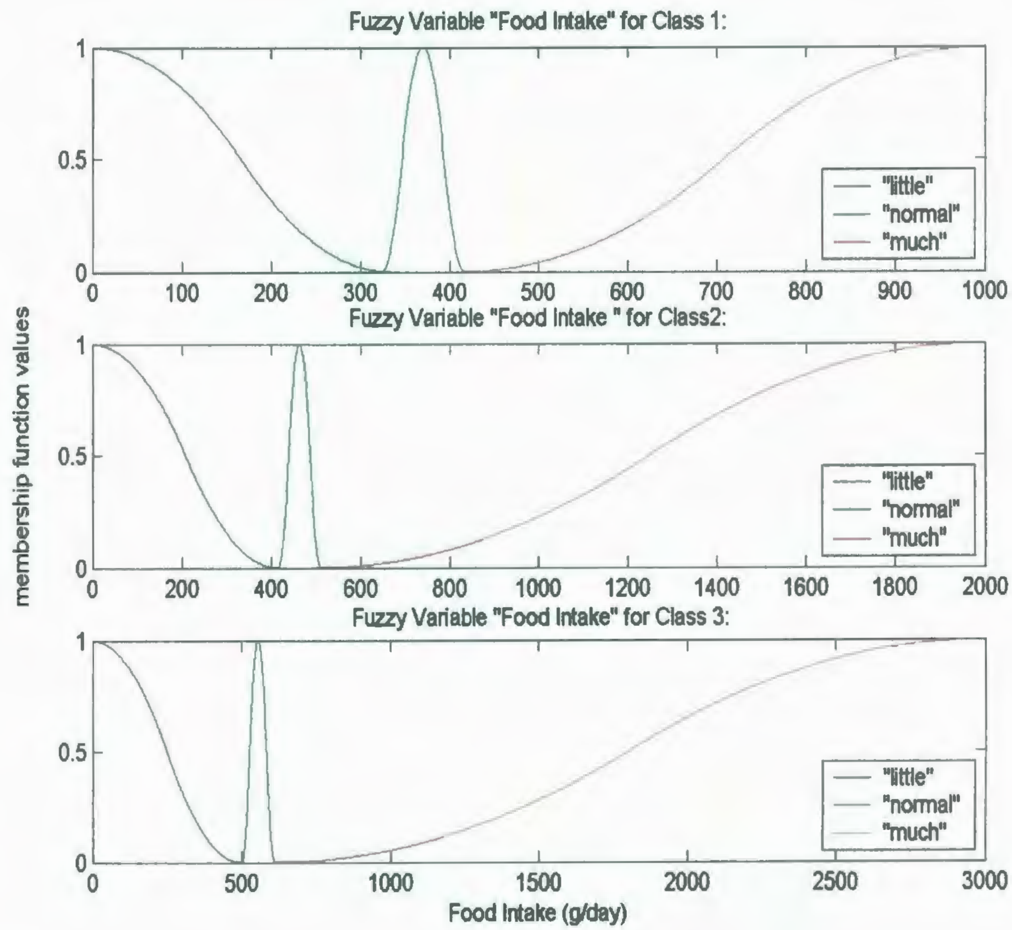


Figure 4 Fuzzy variable “Food Intake” for Class 1, 2, and 3

According to the standards provided by the Merck Manual, there are small overlaps between the fuzzy sets for "little" and the fuzzy sets for "normal", as well as between the fuzzy sets for "normal" and the fuzzy sets for "much"; although they are too tiny to be clearly seen in Figure 4. These overlaps represent part of the fuzziness of the fuzzy terms. The knowledge engineer can adjust the current settings of these fuzzy sets later if he/she wants to emphasize the overlaps more.

3.4.3 Certainty Factors

When the knowledge engineer of MADHS interviews domain experts to get their knowledge (rules), or designs various questionnaires for patients to get input facts, he or she should be prepared for unreliable or incomplete answers from the experts and patients. In a fuzzy expert system, this level of uncertainty can be represented and calculated using the Certainty Factors of the rules and facts.

The definition of Certainty Factor has been introduced in Section 2.5, together with the calculation of CFs in FuzzyCLIPS.

As mentioned earlier in Section 2.5, negative certainty factors (between [-1, 0]) are not used here in MADHS. For both positive (True) and negative (False) decisions, the values of certainty fall between [0, 1], which means from no confidence to total confidence. No matter what range of CF is used, the experts are supposed to provide the CFs of the rules. The patients are supposed to provide the CFs of the input facts. It is the responsibility of an expert system to calculate the CFs of the consequent and every intermediate decision made during the decision-making process.

In order to bring the subjective certainty factors closer to precise measurement of probabilities, the designers of MYCIN decided to ask the domain experts for the *belief* and *disbelief* of a hypothesis first, and then calculated the corresponding CF from these given beliefs.

On the contrary, the knowledge engineer of MADHS doesn't stop the domain experts from giving certainty factors directly, without calculation of *beliefs* and *disbeliefs*. In MADHS, disbeliefs are not measured and directly set to zero. How much a user/expert really believes in a fact /rule is the most import thing being taken care of here.

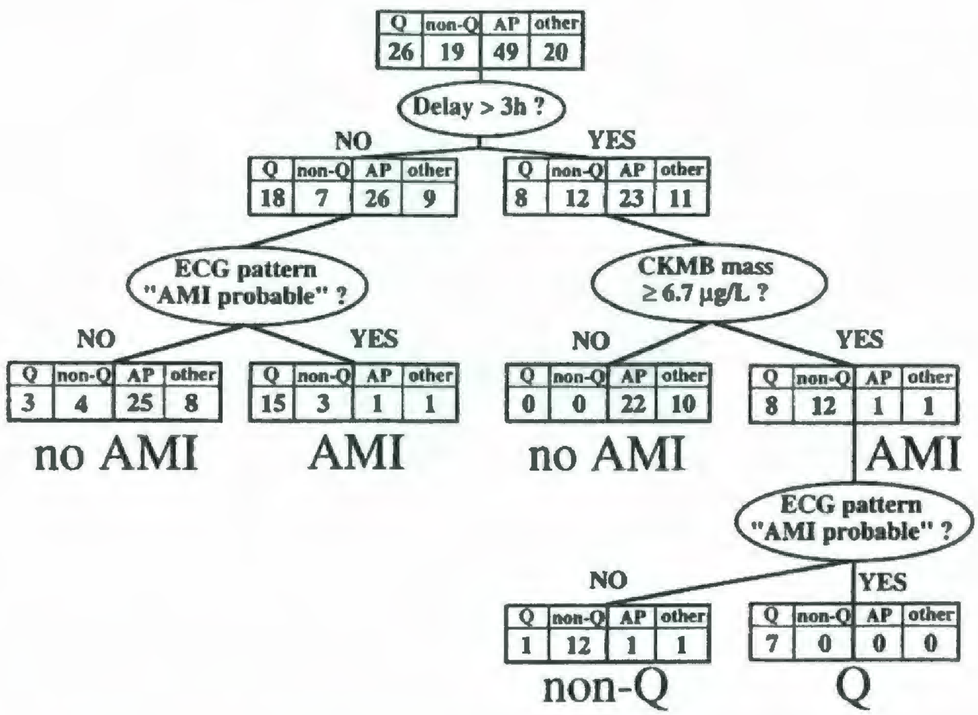
Because the CF calculation is closely related to the knowledge representation method and inference techniques used by MADHS, the details will be explained in the next section. For those simple rules in the knowledge bases, the calculation of CF is similar to the theory used by Fuzzy CLIPS. But for complex rules with multiple antecedents or multiple consequents, the equations are quite different.

3.4.4 Structure of the Knowledge Bases

When encoding the fuzzy sets, fuzzy rules and procedures to perform fuzzy inference, possible knowledge representation methods like Decision Trees will be very helpful.

A node (or a "leaf") in a classical decision tree takes one or several options (or properties) as input, and outputs a "Yes" or "No" decision as result. Decision trees can be used in medical diagnoses as visual and analytical tools to represent decision-making methods and their consequences. Usually those consequences could be chance event outcomes, resource costs, and utilities.

A figure below shows a brief example of a traditional decision tree used in the early diagnosis of acute myocardial infarction for nontraumatic chest pain patients at hospital admission:



A decision tree for the early diagnosis of acute myocardial infarction, from one online article provided by CHEST, the official journal of the American College of Chest Physicians.

The decision tree above is a typical result of *classification and regression tree analysis* (CART), which is widely used in today's medical practices. The integers below every "Yes" or "No" decisions in the figure represent the learning dataset (classified numbers of patients) used during the building of the tree. Decision trees of this kind can be easily transformed into corresponding rule bases with a little help from the knowledge engineers.

When used in clinical diagnosis, a classical decision tree usually starts from making a decision on one main symptom of the patient. According to different results ("Yes" or "No", "True" or "False") of the first step, this tree will go towards separate directions and make decisions on other symptoms. Eventually it will lead to several potential diseases which the patient possibly has.

While currently most of these classical decision trees still use logical values, in real-life diagnoses, especially in traditional Chinese medical practices, there are situations where symptoms of patients can not be described easily by Boolean values. In those cases, fuzziness can be combined into decision tree structures to form Fuzzy Decision Trees. A fuzzy decision tree has the same or similar structure as the traditional decision tree, but with fuzzy variables on the internal tree nodes. A leaf node of a fuzzy decision tree is associated with one or more class labels.

However, the knowledge engineer of MADHS has leant from existing medical manuals that fuzzy decision trees can grow too complicated to be gone through if they completely follow the form of traditional decision trees. That means if a fuzzy decision tree starts from one symptom of a patient, and ends up with several potential diseases, the size of the tree will become considerably large, because all symptoms of these potential diseases should be included into this decision tree. Thus, this tree will have too many levels, branches and nodes.

Currently in MADHS, inference trees are used instead of fuzzy decision trees to show the inference techniques in the knowledge bases. That means fuzzy rules are organized into tree-like structures to facilitate the calculation of certainty factors.

As for the tree structure: the symptoms (Nodes) are placed at the lowest level of an inference tree. Several Nodes can form a Rule at the second lowest level. Several Rules with/with out more Nodes can form another Rule at a higher level. A Rule on the lower level is called a "Sub Rule" of the Rule on a higher level. The rule which leads to the final diagnosis is at the root/top of the tree.

As for the reasoning process: an inference tree starts from decisions on symptoms of a disease, and ends up with a final decision about that disease at the root. The reasoning goes from Nodes to higher-level Rules, and finally leads to the conclusion/root. The decisions on the lowest level (about Nodes) will form the decisions on higher levels (about Sub Rules). The decisions about Sub Rules will form the decisions about Rules and Main Rules.

Figure below 5 shows part of an inference tree, with the lowest and the second lowest level Nodes and Rules. The complete inference tree is shown in Figure 9.

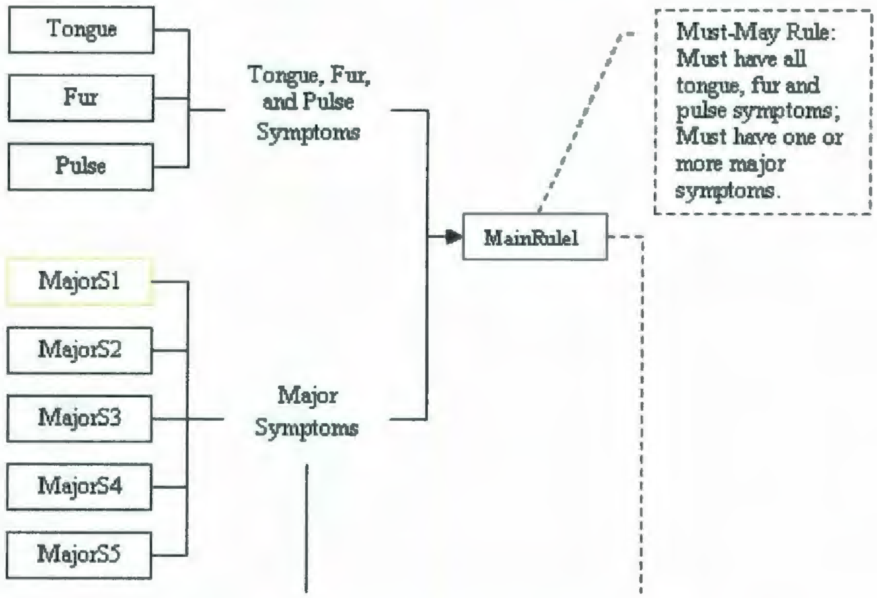


Figure 5 Part of an inference tree in MADHS

In Figure 5, MajorSn, Tongue, (Tongue) Fur, and Pulse are symptoms of a disease. MainRule1 is a Rule on the higher level, which is closely related to these eight symptoms. The content of MainRule1 is: the patient must have all tongue, fur and pulse symptoms, while having one or more Major symptoms.

Figure 5 can also be easily translated into a rule base. In order to facilitate programming, "has one or more Major symptoms" can also be treated as a Sub Rule of MainRule1. If so, the small part of inference tree shown in Figure 5 can be translated into two rules in Jess, one Sub Rule and one Main Rule:

Sub Rule:

```
(defrule Rule_Has_More_Than_One_Major_Symptoms_TRUE
"Rule_Has_More_Than_One_Major_Symptoms_TRUE"
(declare (salience 250))
(or (Record (name "Anorexia_Poor_Appetite") (decision TRUE) (CF ?x))
(Record (name "Distention_and_Fullness_of_Gastric_Cavity_and_Abdomen")
(decision TRUE) (CF ?y))
(Record (name "Pain_of_Gastric_Cavity_and_Abdomen") (decision TRUE)
(CF ?z))
(Record (name "Favor_Warmness_and_Relief_with_Pressure") (decision TRUE)
(CF ?p))
```

```
(Record (name "Aversion_to_Cold_and_Coldness of_Limbs") (decision TRUE)
(CF ?q)))
=>
(open
multi/framework/patientManagement/002/RulebaseS2Deficiency of Spleen Yan
g.txt outRouter "a")
(bind ?HMTOMS (new framework.decisiontree.CrispNode))
(call ?HMTOMS setName "Has_More_Than_One_Major_Symptoms")
(call ?HMTOMS setResultCF 1.0)
(call ?HMTOMS setResultDecision TRUE)
(call ?*MainRule1* addMustHaveSymptom ?HMTOMS)
(assert (Has_More_Than_One_Major_Symptoms Done))
```

Main Rule:

```
(defrule MainRule1 "MainRule1"
(declare (salience 100))
(Pale_and_Enlarged_Tongue Done)
(White_Fur_or_Slippery_Tongue Done)
(Deep_Slow_and_Adynamic_Pulse Done)
(Has_More_Than_One_Major Symptoms Done)
=>
(call ?*MainRule1* setResultCF)
(assert (Record (name "MainRule1") (decision (call (call ?*MainRule1*
getResult) getDecision)) (CF (call (call ?*MainRule1* getResult)
getCF))))
(assert (MainRule1 Done))
```

Note that the salience of Sub Rule is declared as 250, while the salience of Main Rule is declared as 100. Saliences should be set during programming. A rule with higher salience will be fired first in Jess knowledge bases. So the Sub Rule at the lower level will be fired before the Main Rule on the higher level.

Fuzziness of an inference tree is shown by the fuzzy variables and certainty factors assigned to the internal nodes. MajorS1, which is marked by color yellow in Figure 5, is a fuzzy node with corresponding fuzzy variable "foodAmd".

The following part of this section will explain how inference trees are formed from small nodes (leaves) to the trunk for knowledge bases in MADHS.

3.4.4.1 Node

Class **Record**:

Class "Record" is the smallest storable unit. Every Node, Rule and Rulebase in MADHS has a Record in it to store its name, decision and the result of CF calculation at its level. Various kinds of Records are listed as follows:

1) Regular Record

   Sub class of the class "Record". No additional attributes and methods. It makes the code easier to read, so the designers can differentiate between regular records and the lab records.

2) Lab Record

   Sub class of the class "Record". It has an extra String member "diagnosis" to record the lab results reported by the Lab Agents.

Class "Record" and its sub classes are illustrated by the UML diagrams in Figure 6:

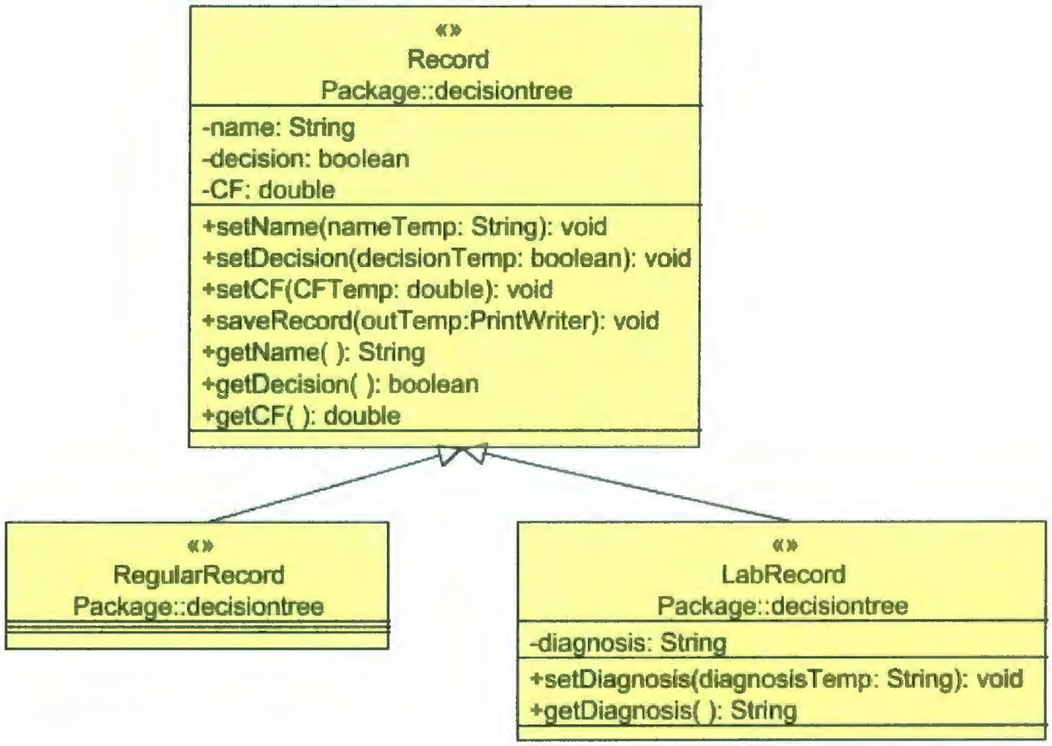


Figure 6 Class "Record" and its sub classes

Class **Node**:

Class "Node" is the smallest unit of decision-making. A Node in MADHS can represent a premise of a Rule, or a conclusion of a Rule.

Subclasses of the class Node are:

1) Crisp Node

   When a Node doesn't have any fuzzy member with it, it is called a "Crisp" Node. The word "crisp" was also used in Fuzzy CLIPS to describe the facts and rules without fuzzy variables.

2) Fuzzy Node

When a Node has a fuzzy variable "FuzzyNodeCF" with it, it is called a Fuzzy Node. A "FuzzyNodeCF" represents the certainty factor assigned directly by the experts to a premise, or by the user to an input fact.

3) Lab Node

Lab Node has an extra variable "additionalInfo" to record the lab results reported by the Lab Agents.

Class "Node" and its sub classes are illustrated in Figure 7.

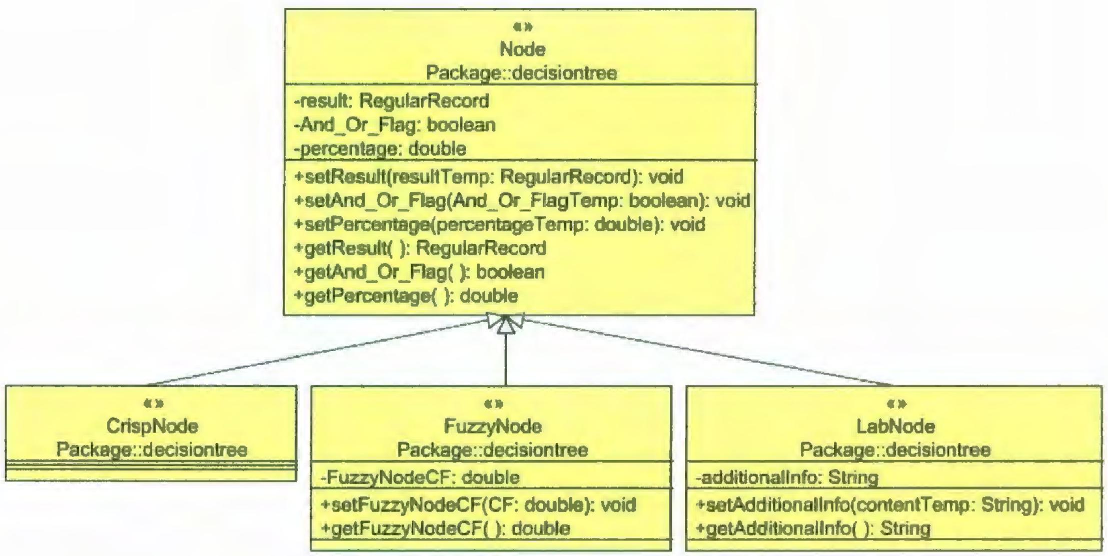


Figure 7 Class "Node" and its sub classes

3.4.4.2 Rule

Class **Rule**:

The antecedents and consequents of Rules are formed by various kinds of Nodes.

In Fuzzy CLIPS, rules are classified to Simple Rules and Complex Rules.

Simple Rules include:

- CRISP_Simple Rule
- Fuzzy_Simple Rule
- Fuzzy_Fuzzy Simple Rule

Complex Rules include:

- Multiple Consequents
- Multiple Antecedents

The CF calculation in these types of rules in FuzzyCLIPS has been explained in Section 2.5. Note that the patients are supposed to provide the CFs of the input facts. The experts are supposed to provide the CFs of the rules. It is the responsibility of an expert system to calculate the CFs of the consequent and every intermediate decision made during the decision-making process.

In MADHS, Rules are classified according to the types of criteria used in traditional Chinese and western medical diagnoses. Each type of Rule has its own method of CF calculation.

For example, in western medicine, the criteria for the diagnosing of "Rheumatic Heart Disease" were first published in 1944, and then revised by the American Heart Association and other groups [8]. According to the first main rule elicited from their works (which is highlighted by color blue in Table 3), two major criteria or one major and two minor criteria support the diagnosis of rheumatic fever. This Rule is called a "Major_Minor_Rule", because it deals with the Major criteria and Minor criteria of diagnosis. The criteria and rules important to the diagnosis of Rheumatic Heart Disease are listed in Table 3:

Table 3 Diagnosis of "Rheumatic Heart Disease"

Disease Name:	Rheumatic Heart Disease
Major Criteria:	• Carditis (Inflammation of Heart Muscle) (laboratory test needed); • Migratory Polyarthritis; • Sydenham's Chorea (sub symptoms: Facial Grimacing, Hypotonia, Loss of Fine Motor Control, Gait Disturbance); • Erythma Marginatum; • Subcutaneous Nodules.
Minor Criteria:	• Fever; • Arthralgia; • Laboratory Abnormalities (laboratory test needed); • Electrocardiogram Abnormalities (laboratory test needed).
May-Have Symptoms:	• Abdominal Pain; • Epistaxis.
Laboratory Tests:	• Blood Test: Increased Erythrocyte Sedimentation Rate; Increased C reactive Protein; Leukocytosis; • Electrocardiogram; • Culture for Group A Strep: Positive, Elevated or Rising Antistreptolysin O title; • X-Ray; • Ultra-Sound.
Main Rules:	1. >=2 major criteria or >=1 major criteria and >=2 minor criteria 2. May have Abdominal Pain and Epitaxis
Treatment:	...

In traditional Chinese medicine, the classification of Rules also depends on the types of criteria used in diagnoses. The symptoms and rules used in the diagnosis of "Deficiency of Spleen-Qi" are elicited from a Chinese medical manual [4] and listed in Table 4. The second main rule in the table (which is also highlighted in blue) belongs to the

Must_May_Rule, because it deals with the Must-Have symptoms and May-Have symptoms in the diagnosis.

Table 4 Diagnosis of "Deficiency of Spleen-Qi"

Disease Name:	Deficiency of Spleen-Qi
Major Criteria:	• Anorexia and Poor Appetite; • Distension and Fullness of Gastric Cavity and Abdomen; • Loose Stool.
Minor Criteria:	• Sallow Complexion; • Lassitude of Limbs; • Short Breath and Speech Less; • Hypodynamia; • Emaciation of Body, or Enema of Limbs.
Must-Have Symptoms:	• Slow Start; • Long Duration.
May-Have Symptoms:	• Eating Disorder; • Tired Body; • Emotion Disorder; • Weak After Protracted Disease; • Pale and Tender Tongue, or Teeth-Printed; • White Fur; • Moderate and Weak Pulse.
Laboratory Tests:	Stool Test
Main Rules:	1. The patient must have at least 2 major symptoms and at least 1 minor symptom. 2. The patient must have all the must-have symptoms, and some of the may-have symptoms;
Treatment:	…

After a quick view of the two examples above, the types of Rules in MADHS (or the subclasses of the class "Rule") are introduced as follow:

1) Major_Minor_Rule:

A Major_Minor_Rule is a Rule dealing with Major symptoms (criteria) and Minor symptoms (criteria) in the diagnoses.

In Table 3, main rule 1 is a typical Major_Minor_Rule. This type of Rule has its special attributes that are critical to the CF calculation. These attributes are illustrated in Figure 8. *CF_Rule_Major_Minor_Rule* represents the CF of this Rule; *thresholdMajor* is the smallest number of major symptoms a patient should have in order to be diagnosed to have this disease; *thresholdMinor* is the smallest number of minor symptoms; *countMS* is the actual number of major symptoms appearing on this patient; *countAS* is the actual number of associate (minor) symptoms appearing on this patient.

The equations of CF calculation in Major_Minor_Rules are:

a. CF calculation for Major symptoms:

$$CF_{Major} = \frac{\sum_{n=1}^{N_{MajorS}} CF_n}{N_{MajorS}} * (1 - \lambda_{MajorSOR}) + \frac{\sum_{m=1}^{N_{MajorSOR}} CF_m}{N_{MajorSOR}} * \lambda_{MajorSOR} \quad (1)$$

In Major_Minor_Rules, especially in the Rules for traditional Chinese medical diagnoses, sometimes there are "Or" relations among Major symptoms (and among Minor symptoms).

For example, a traditional Chinese medical manual (or a Chinese doctor) may list the Minor symptoms of the disease "Deficiency of Spleen-Yang" as follows: "Shallow Complexion; Dysgeusia and No Thirst; Loose Stool; Dysuria; Or Heavy Limbs; Or Edema of Body; Or Leukorrhagia." [4] Those symptoms with an "Or" before them are treated as less important than other symptoms without an "Or" in traditional Chinese medicine.

Thus, in order to calculate the CF of all the Major symptoms, a weight $\lambda_{MajorSOR}$ (< = 0.5) is given to the Major symptoms with an "Or" before them. The other Major symptoms use $1-\lambda_{MajorSOR}$ as the weight.

Besides, in Equation 1, $N_{MajorSOR}$ represents the number of Major symptoms in an "Or" relation, while N_{MajorS} represents the number of other Major symptoms actually appearing on the patient.

b. CF Calculation for Minor symptoms:

$$CF_{Minor} = \frac{\sum_{n=1}^{N_{MinorS}} CF_n}{N_{MinorS}} * (1-\lambda_{MinorSOR}) + \frac{\sum_{m=1}^{N_{MinorSOR}} CF_m}{N_{MinorSOR}} * \lambda_{MinorSOR} \quad (2)$$

To calculate the CF of all the Minor symptoms, $\lambda_{MinorSOR}$ (< = 0.5) is given to the Minor symptoms with an "Or" before them. The other Minor symptoms use $1-\lambda_{MinorSOR}$ as the weight.

In Equation 2 above, $N_{MinorSOR}$ represents the number of Minor symptoms in an "Or" relation, while N_{MinorS} represents the number of the remaining Minor symptoms which are actually appearing on the patient.

Finally, the certainty factor asserted with the consequent of a Major_Minor_Rule is calculated by Equation 3 below:

$$CF_{Major\_Minor} = (CF_{Major} * \lambda_{Major} + CF_{Minor} * (1 - \lambda_{Major})) * CF_{Major\_Minor\_Rule} \quad (3)$$

Because Major symptoms are more important than Minor symptoms in the diagnosing of diseases, a weight λ_{Major} is given to the CF of Major symptoms calculated by Equation 1, while $1 - \lambda_{Major}$ is the weight given to the CF of Minor symptoms calculated by Equation 2. $CF_{Major\_Minor\_Rule}$ in Equation 3 represents the CF given to this Rule by the domain expert.

As mentioned earlier in Section 2.5 and Section 3.4.3, in most of the time, negative CFs are not calculated in MADHS. The reason can be explained clearly here by an example. For instance, in Table 3, the first rule of "Rheumatic Heart Disease" requires that a patient with this disease should have two or more major symptoms; or one or more major symptoms and two or more minor symptoms. If a patient has already got enough numbers of major symptoms and minor symptoms, then the negative CFs of the remaining symptoms should not affect the CF calculation of this Rule.

2) Must_May_Rule:

A Must_May_Rule is a Rule dealing with Must-Have symptoms and May-Have symptoms in the diagnosis of a disease.

In Table 4, main rule 2 is a typical Must_May_Rule. The class "Must_May_Rule" has a special attribute illustrated in Figure 8: *CF_Rule_Must_May_Rule* represents the CF of this Rule.

The equations of CF calculation for Must_May_Rules are:

a. CF calculation of Must-Have symptoms:

$$CF_{Must} = \min(CF_1, CF_2, \ldots, CF_{N_{MustS}}) \quad \textbf{(4)}$$

If a symptom is described as a "Must-Have" one in the diagnosing of a disease, then this symptom can be seen as one of the premises in order to get to the conclusion that the patient really has this disease. As mentioned earlier in Section 2.5, Equation 4 treats a Must_May_Rule with multiple Must-Have symptoms as a combination of several one-antecedent rules. In Equation 4 above, N_{MustS} represents the number of Must-Have symptoms described by the domain expert in the original interview.

b. CF calculation of May-Have symptoms:

$$CF_{May} = \max(CF_1, CF_2, \ldots, CF_{N_{MayS}}) \quad \textbf{(5)}$$

If a symptom is described as a "May-Have" one in the diagnostic criteria of a disease, the symptom is called a May-Have symptom. Currently, the CF of the May-Have symptoms is calculated by Equation 5. N_{MayS}

represents the number of May-Have symptoms listed out by the domain expert in the building process of the rule base.

Using CF_{Must} calculated by Equation 4 and CF_{May} calculated by Equation 5, the certainty factor asserted with the consequent of a Must_May_Rule is:

$$CF_{Must\_May} = (CF_{Must} * \lambda_{Must} + CF_{May} * (1 - \lambda_{Must})) * CF_{Must\_May\_Rule} \quad (6)$$

Because Must-Have symptoms are far more important than May-Have symptoms in the diagnoses, a weight λ_{Must} is given to CF_{Must}, while $1 - \lambda_{Must}$ is the weight given to CF_{May}. $CF_{Must\_May\_Rule}$ is the CF given to this Rule.

3) Sometimes_Have_Rule:

An expert of traditional Chinese medicine may list the symptoms of "Deficiency of Stomach-Yin" as follows: "The patients with this disease sometimes have (a history of) Late Febrile Disease, or Eating Disorder, or Stagnation of Liver-Qi and Liver-Fire, or Damage of Stomach Yin." A patient just needs to have one of these symptoms to get to the consequent of this Sometimes_Have_Rule. So the certainty factor of the consequent is calculated by:

$$CF_{SH} = \max(CF_1, CF_2, \ldots, CF_{N_{SHS}}) * CF_{SHR} \quad (7)$$

In Equation 7, N_{SHS} represents the number of sometimes-have symptoms listed out by the domain expert for the disease. CF_{SHR} is the CF of the Sometimes_Have_Rule assigned by the domain expert.

4) Lab_Rule:

This type of Rule is designed for the laboratory tests.

$$CF_{Lab} = \max(CF_1, CF_2, \ldots, CF_{N_{LabS}}) * CF_{Lab\_Rule} \qquad \textbf{(8)}$$

In Equation 8, N_{LabS} represents the number of sub tests needed to be done in a laboratory test. For example, a Barium Enema includes five sub tests for Heart, Lung, Bony Throax, Mediastinum and Great Vessels, respectively. If any of these sub tests show "abnormal" (CF>0), then the final lab result will also be "abnormal". The CF of the lab result is calculated by multiplying the maximum of sub CFs with the CF of the Lab_Rule (CF_{Lab_Rule}).

5) Rule_with_Percentage:

In western medicine, a doctor can use the analysis of anamneses as a source of his/her expert knowledge. A typical analysis report of the disease "Ulcerative Colitis" is shown on the next page as an example [13]:

Analysis of 198 Inpatients with Ulcerative Colitis in Chengdu

Objective: To investigate the pathogenic characteristics of inpatients with ulcerative colitis (UC) in recent 10 years in Chengdu.

Methods: To analyse 198 cases collected from three hospitals.

Results: The ratio of male to female was 1.63/1. The mean age was 45.6 years old. The mean duration of the disease was 3.74 years. The cases of mild and moderate severity and left-sided colitis were the most common, accounted for

79.85% and 49.5%, respectively. The major clinical symptoms were diarrhea (89.9%), pus or blood stool (83.2%). The major colonoscopic appearances were mucosal erythema (94%), erosion or frank ulceration (73%). The misdiagnostic rate was 23.7%.

Conclusion: The incidence of UC increased in recent years in our area, especially in the middle age group. Mild severity is the most common and it's rather easy to be misdiagnosed.

As listed above, statistics assigned to symptoms can be employed to guide the CF calculation for a diagnosis. These statistics are simply percentages or frequencies, neither probabilities nor uncertainties. But to a certain extend, a percentage/frequency can represent the importance of the corresponding symptom during the diagnosing of a disease. This is the rationale for CF calculation in equation (9) and (10) for Rule_with_Percentages.

Sometimes, a doctor can not provide a precise percentage for every symptom of a disease. In certain medical manuals, various "grades" are provided instead to describe the frequencies of symptoms. The expert or knowledge engineer can assign statistics to the symptoms according to their grades. For example, Table 5 on the next page shows the diagnosis criteria of "Ulcerative Colitis" from two medical manuals [24][26]. The "grades" used in the table are explained below it.

Table 5 Diagnosis of "Ulcerative Colitis"

Disease Name:	Ulcerative Colitis
Criteria (Symptoms)	• Abdominal Pain; (Grade 1) • Diarrhea; (Grade 1) • Pain with Pressure; (Grade 2) • Anorexia and Poor Appetite; or Low Fever; (Grade 1) • Swelling of the Colon Tissue (lab); • Rectal Bleeding (lab); (Grade 1) • Ulcerations or Erytheme of the Surface of the Colon (lab); • Digital Rectal Examination.(Grade 2)
Laboratory Tests:	• X-Ray; (Grade 2) • Stool Test: Blood and Pus; (Grade 2) • Sigmoidoscope or Colonoscopy; (Grade 1) • Stool Cultures; (Grade 3) • Blood Test; (Grade 2) • Blood Chemistry Test. (Grade 3)
Main Rules:	None
Treatment:	...

Grade 1: Most of the patients have this symptom/need this lab test. The expert or knowledge engineer can assign a percentage to this grade, like 100%.
Grade 2: Many patients have this symptom/need this lab test. The expert or knowledge engineer can assign a percentage to this grade, like 70%.
Grade 3: Only some of the patients have this symptom/need this lab test. The expert or knowledge engineer can assign a percentage to this grade, like 30%.

According to the adjectives used in the description of the disease, the number of grades used to classify the symptoms may vary from manual to manual.

Using the statistics or "grades" as part of the expert knowledge, Rule_with_Percentage calculates the CF of the asserted conclusion as follows:

$$sum = \sum_{m=1}^{NumOfGrade} PercentageOfGrade_m \qquad (9)$$

$$CF_{WP} = \sum_{n=1}^{N_{WPS}} \frac{CF_n * (Percentage_n / sum)}{lengthOfGrade_n} \qquad (10)$$

Suppose there are n input facts matching the n antecedents of a Rule_with_Percentage. And these n facts are classified into m grades in a medical analysis.

In Equation 9, $PercentageOfGrade_m$ represents the statistics (like 89.9% in the analysis report, or 70% in Table 5) assigned to Grade m.

In Equation 10, CF_n represents the certainty factor of the nth fact, CF_{WP} represents the CF of the whole Rule, N_{WPS} represents the number of symptoms being added to this Rule as antecedents; $Percentage_n$ is the statistic assigned to symptom n; $LengthOfGrade_n$ represents the number of symptoms in the same "grade" where symptom n belongs to.

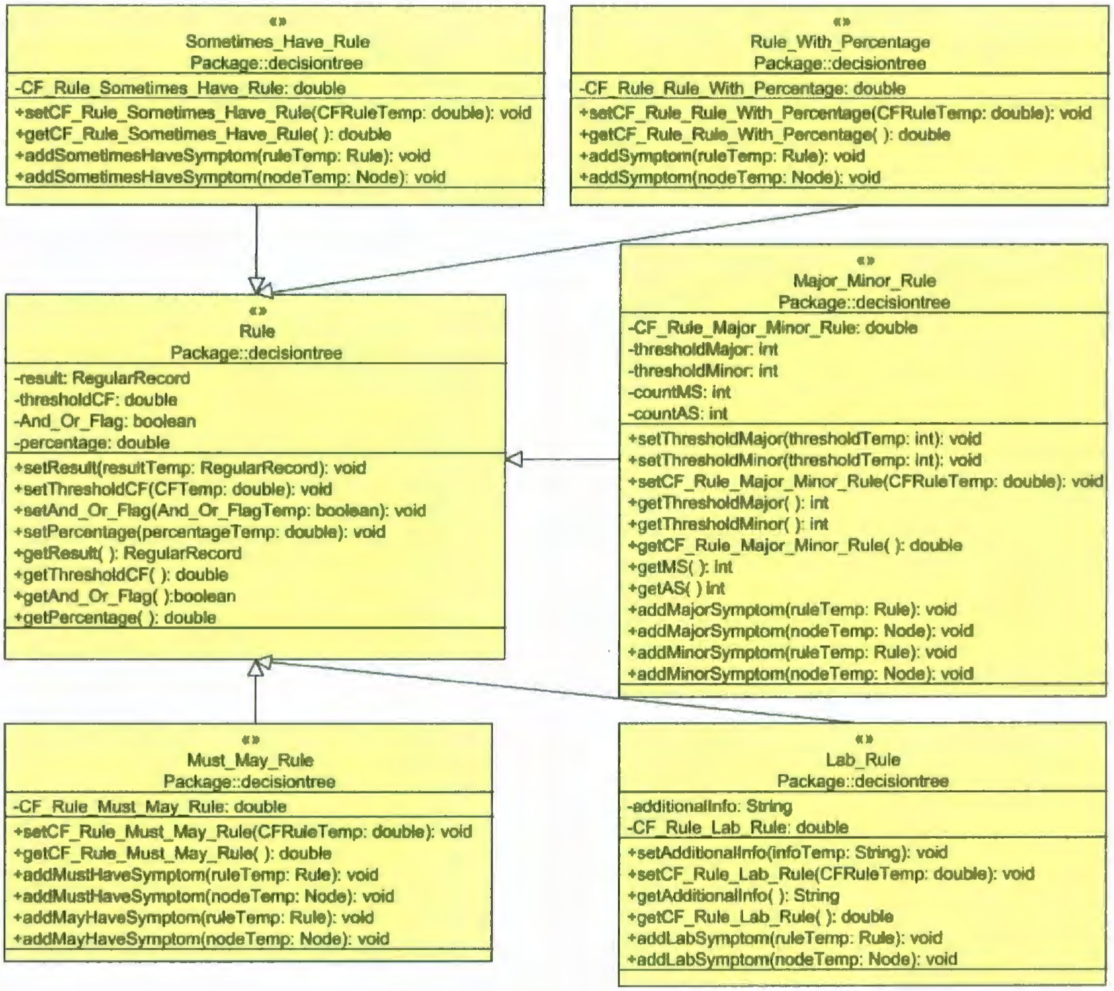


Figure 8 Class "Rule" and its subclasses

3.4.4.3 Rulebase

Class **Rulebase**:

Every cooperative agent (expert or examiner) who participates in the multi-agent diagnoses has its own rule bases that are different from other agents.

In MADHS, there are mainly four different kinds of rule bases:

- Java rule bases (for Traditional Chinese Medicine)

- Jess rule bases (for Traditional Chinese Medicine)
- Java rule bases (for western medicine)
- Java rule bases (for labs)

Java rule bases (for Traditional Chinese Medicine, western medicine, and lab) use java programming language to write the knowledge bases. No special inference engine is used. Those Java rule bases are constructed by java classes like Nodes and Rules, which were introduced in the above sections.

Because different types of rule bases are used in our prototype system, currently, class Rulebase does not have a unified object-oriented form. When Rule objects are organized together to form a Rulebase, conditional statements like "if, then, else" are used instead to form a decision-tree-like structure. So at the Rulebase level, actually a kind of procedural programming is used. It should be modified in later development phases.

When using Java rule bases, the knowledge bases get patient information from corresponding databases or file system, and use them as input facts. A rule base will use these input facts to check the LHS (left hand side) of every rule, and decide whether or not to fire the rule and make a decision.

One major disadvantage is easily discovered here: redundant Nodes. Figure 9 illustrates the rule base for one disease named "Deficiency of Spleen-Yang" in the Coordinator. It also shows the inference technique applied to Java rule bases.

MainRule1 in this graph points out that the patients with this disease must have all tongue, fur and pulse symptoms, while having one or more Major symptoms. In order to fire this rule, all three Nodes for the tongue, fur and pulse symptom should be checked; and all five Nodes for the Major symptoms should be checked too.

MainRule2 states that the patients with this disease have two or more Major symptoms and two or more Minor symptoms. In order to fire this rule, all seven Nodes for the Minor symptoms should be checked; and all five Nodes for the Major symptoms should be checked again.

At this phase of development, no algorithm has been applied to reduce this redundancy in the checking process of the LHS of different rules. Methods that can be used to solve this problem are listed as follows:

a) Nodes could be organized better to avoid this redundancy.
b) Remember past LHS test results during the rule activations. Search the blackboard or the related records of the patient for existing result first before checking one node/rule.

Jess rule bases (for Traditional Chinese Medicine) use Jess inference engine and corresponding knowledge base encoding. The inference trees are converted into 'if-then' rules and then input as knowledge base to Jess inference engine. Expert system building tools like Jess use a very efficient inference technique known as the Rete (Latin for *net*) algorithm [3]. In the Rete algorithm, the inefficiency described above is reduced by remembering past test results in the rule activations.

Rete has one obvious advantage: it allows the rules to share nodes across a pattern network. Redundant/same nodes in the LHS (left-hand-side) part of different rules are combined into just one node. That means each unique node in the rule base is tested for just one time.

However, since the results of rule activations need to be remembered, when using Rete, Jess' memory usage is very considerable. In Chapter 5 and Chapter 6, during the test of the prototype system, the average memory usage of Jess rule bases is around 2000K more than that of Java rule bases.

On the other hand, because of those redundant nodes, Java rule bases in MADHS will run apparently slower than Jess rule bases. Although both Java and Jess rule bases can diagnose a certain disease in less than 1 second.

The agents especially composed for the testing are introduced below:

1) Coordinator

The Coordinator has its own Java Rulebases for four stomach diseases and four heart diseases. In the testing of the prototype system (Chapter 5), the rules for diagnosing each disease are elicited from traditional Chinese medical books. The rules for all eight diseases in the Coordinator are elicited from a famous traditional Chinese medical manual [4].

For example, the rule base (showing inference tree at the same time) of the disease "Deficiency of Spleen-Yang" in the Coordinator is illustrated by Figure 9.

In Figure 9, the yellow rectangles represent the Fuzzy Nodes. The purple rectangles represent the Rules and Sub Rules. The green rectangles represent the Nodes that need information from other agents. The dashed lines represent the "Or" relations.

2) Specialist1

Specialist1 is the name given to one expert agent participating in the testing of the prototype (described in Chapter 5). It can also diagnose four stomach diseases and four heart diseases. The rules for these eight diseases are elicited from a different traditional Chinese medical manual [27].

The Java Rulebase of "Deficiency of Spleen-Yang" used by Specialist 1 is illustrated by Figure 10. The parts within the dashed blue rectangle are the four nodes different with the rule base in Figure 9.

3) Specialist 2

Specialist2 is the name given to another expert agent participating in the test. The rules for the eight diseases are elicited from four traditional Chinese medical manuals [4][24][26][27]. Different from other experts, Specialist 2 uses Jess Rulebases only. The rule bases of all eight diseases in the knowledge base of Specialist2 are implemented by Jess (combined with Fuzzy Jess packages). Rete algorithm is used as the inference technique.

The rule base of "Deficiency of Spleen-Yang" used by Specialist 2 is almost the same with the rule base used in the Coordinator, except for one node "Stool Test".

4) Laboratory 1

Figure 11 shows the Java Rulebase of "Stool Test" in Laboratory 1(an Examiner), which provides the lab result for the rule base in Figure 9 and Figure 10.

The rules of the laboratory tests in Laboratory 1 and Laboratory 2 are elicited from two western medical laboratory manuals [6] [14].

Since knowledge engineering is one of the main contributions of this research, java rule bases for eight diseases (four heart diseases, four stomach diseases) have been designed and generated for the Coordinator and Specialist1. Each rule base includes 2 to 3 main rules, up to 3 sub rules, and 9 to 20 crisp or fuzzy nodes. The average scale is about 600 to 1000 lines of java code.

Eight Jess rule bases have been designed and generated for Specialist 2. Each Jess rule base has a similar scale as mentioned above, but is separated into two parts: "facts" (50 to 70 lines of Jess code) and "rules" (400 to 600 lines of Jess code). An example of Jess rule bases is shown in Appendix D of this thesis.

In order to diagnose diseases in western medicine, another eight Java rule bases have been constructed for Specialist 3. Each of them includes 1 to 4 main rules, up to 4 sub rules, 4 to 15 crisp/fuzzy nodes, and 1100 to 1500 lines of java code.

The average scale of the java rule bases used in the lab agents (Lab1 and Lab2) is: 1 lab rule, 1 to 26 lab nodes, about 250 to 850 lines of java code in each rule base.

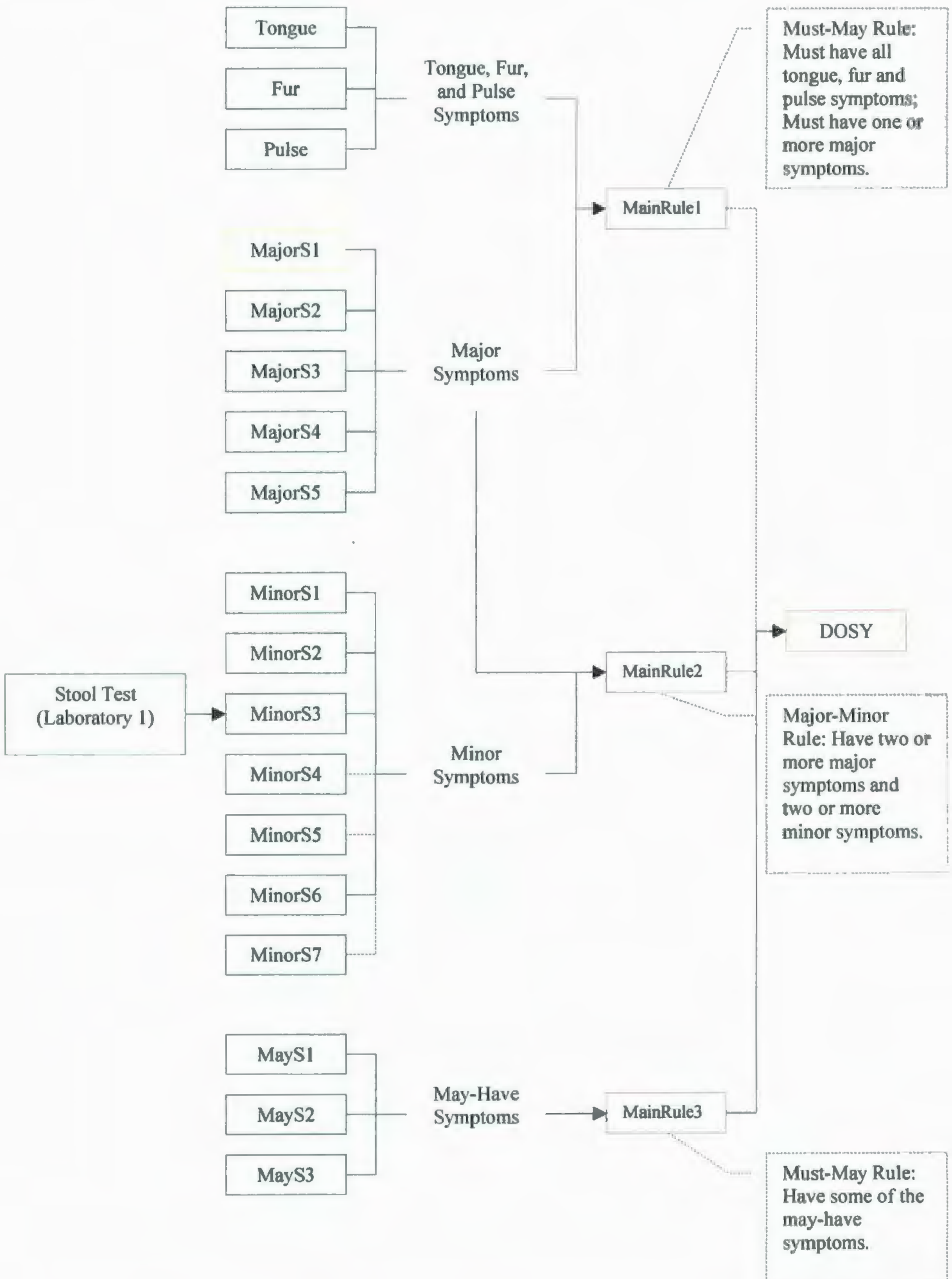


Figure 9 Inference tree for the disease "Deficiency of Spleen-Yang" in the Coordinator

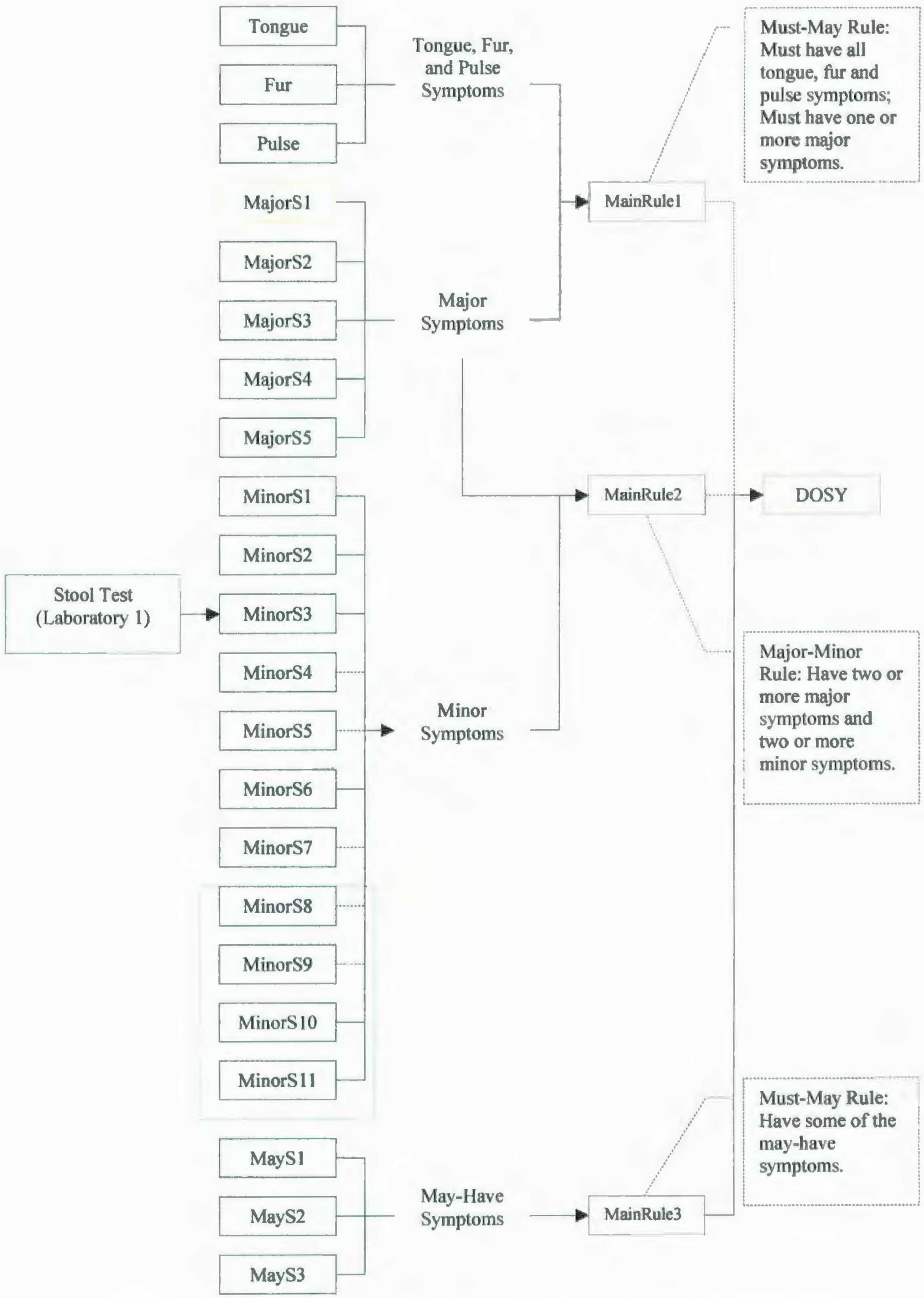


Figure 10 Inference tree for the disease "Deficiency of Spleen-Yang" in Specialist 1

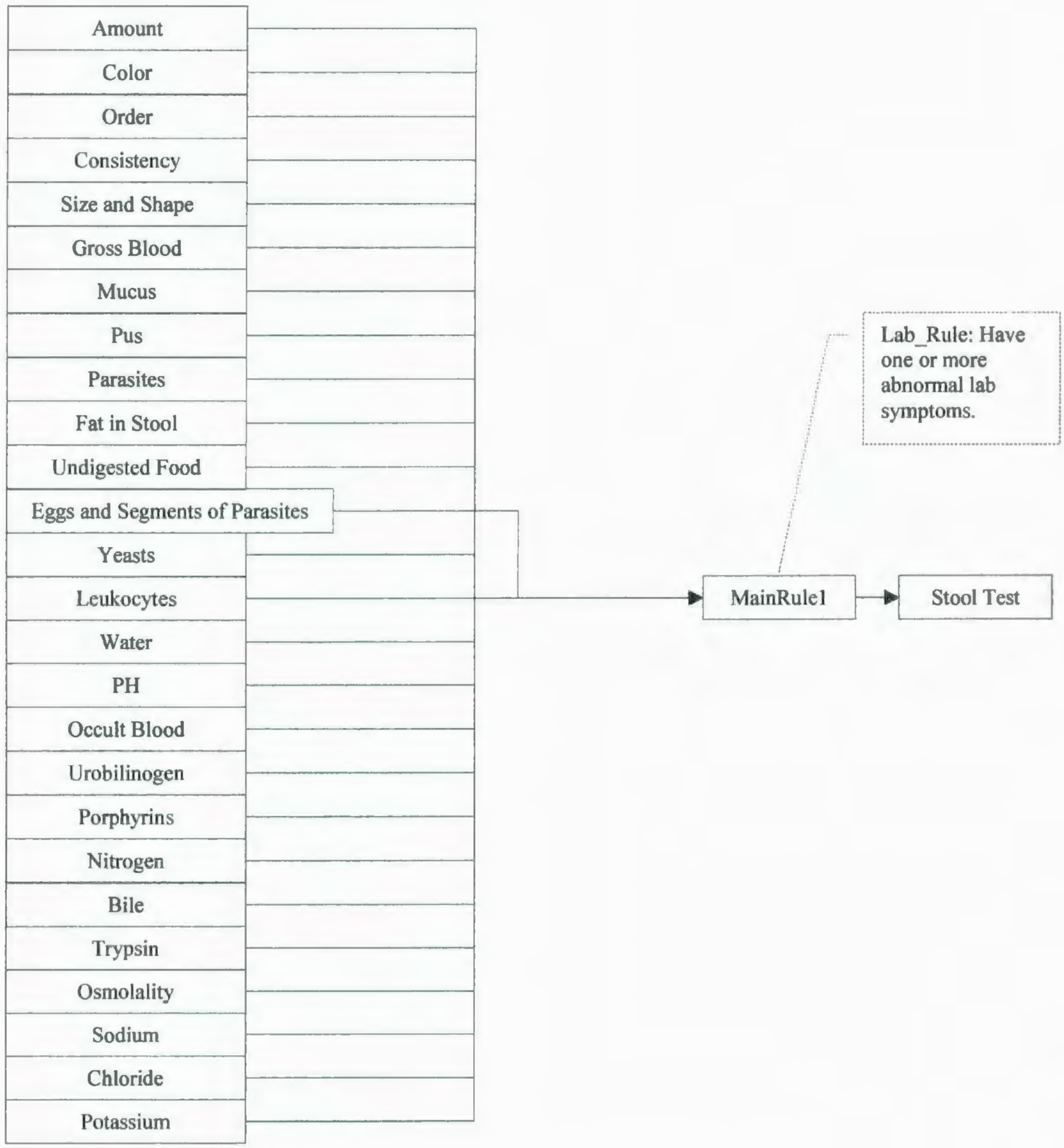


Figure 11 Inference tree for the Stool Test in Laboratory 1

3.5 Joint Decision-Making Methods

The main components of the Joint Decision Maker (JDM) are shown in Figure 12 below:

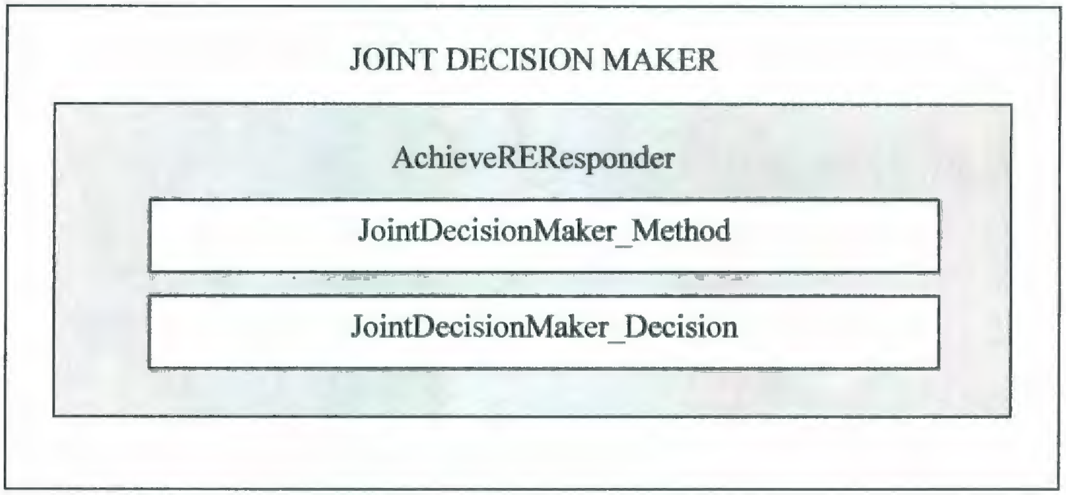


Figure 12 The inner structure of the Joint Decision Maker

The components of the JDM are introduced as follows:

1) JointDecisionMaker_Method;

As said by its name, this part of the JDM is responsible for invoking the interface to allow users of MADHS to choose an appropriate joint decision-making method, and set values of the parameters for this method at the same time.

For instance, if a user chooses to use *weighted range voting* as the joint decision-making method, then this part of the JDM allows the user to set the weight of each and every agent in the team of experts.

After this step, the chosen joint decision-making method and its parameters are attached to the initial plan of joint diagnosis to form the final plan of joint diagnosis.

2) JointDecisionMaker_Decision;

This part of the JDM is responsible for making the final decision according to the pre-chosen joint decision-making method and the diagnosing results from participating experts.

As mentioned previously in Section 2.4, earlier joint decision-making methods like Jennings and Hogg's framework mainly focused on socially rational decision-making systems where the available time or resources are limited. Traditional social welfare functions or utility functions can't be used properly here to represent the "benefit" of the agent society in MADHS.

However, inspired by their work, the idea of "rank", "authority", and corresponding "weights" attached to each and every agent in the society is adopted in MADHS. Currently, the only available joint decision-making method for the users to choose is a revised version of *range voting*, which can be called as *weighted range voting*.

In classical *range voting*, the voters give every candidate a score within a certain range. The sum of the scores is the rating of the candidate. In *weighted range voting* here, the score (CF of the diagnosing result) of each candidate (possible disease) is multiplied by the weight of the Specialist who made the decision. Then the "weighted" scores are added up, and the average of the sum is taken as the final rating of the candidate. The CF of the final diagnosis is calculated by Equation 11 and Equation 12. Note that currently only the CFs of the positive diagnoses are calculated in the equations.

$$CF_{pi} = \begin{cases} CF_i & \text{if the diagnosis is positive} \\ 0 & \text{if the diagnosis is negative} \end{cases} \qquad (11)$$

$$CF_f = \frac{\sum_{i=1}^{N} weight_i * CF_{pi}}{\sum_{i=1}^{N} weight_i} \quad (12)$$

N in equation 12 is the number of Specialists participating in the joint diagnosis. $weight_i$ represents the weight given to the decision of Specialist *i*.

For example, Patient Wang (with unknown stomach problems) has been diagnosed by the Coordinator at the beginning of the joint diagnosing. The possible diseases listed by the Coordinator are: "Deficiency of Spleen-Qi" with a CF of 0.76, "Deficiency of Spleen-Yang" with a CF of 0.80, "Deficiency of Stomach-Qi" with a CF of 1.00. The Coordinator has also formed a team of Specialists for each and every possible disease, as indicated by Table 6.

Then the Joint Decision Maker will ask the user to choose a joint decision-making method and assign corresponding values for the parameters. Suppose a user chooses "weighted range voting" as the joint decision-making method, and assigns a weight for each team member of every possible disease, as shown in Table 6:

Table 6 Weights assigned to the Specialists in the teams for the possible diseases

Disease/Team Members	*Coordinator*	*Specialist1*	*Specialist2*
Deficiency of Spleen-Qi	1.00	0.80	0.70
Deficiency of Spleen-Yang	1.00	0.90	0.80
Deficiency of Stomach-Qi	1.00	0.95	0.85

Later, when the Joint Decision Maker wants to make a joint diagnosis, it will calculate the final CFs of the candidate diseases according to "weighted range voting" method and the weights chosen by the user at an earlier time. In this way, MADHS can differentiate participating experts according to their abilities, experience and authorities. If a user is not able to evaluate the team members according to their qualities, the JDM will use default settings provided by the knowledge engineer.

Suppose Patient Wang's diagnosing results by the team members are shown in Table 7:

Table 7 Diagnosing results from the Specialists in the teams for the possible diseases

Disease/Team Members	*Coordinator*	*Specialist1*	*Specialist2*
Deficiency of Spleen-Qi	True, CF 0.76	True, CF 0.76	True, CF 0.78
Deficiency of Spleen-Yang	True, CF 0.80	False, CF 1.00*	True, CF 0.80
Deficiency of Stomach-Qi	True, CF 1.00	True, CF 1.00	True, CF 1.00

*In "weighted range voting", because currently only the CFs of the positive diagnoses are calculated, if a diagnosis is negative (False), then the corresponding CF is calculated as zero. So negative CF 1.00 here is calculated as positive CF 0.00 in the equations below.

The Certainty Factors of the final diagnoses are calculated according to Equation 11 and Equation 12:

For Deficiency of Spleen-Qi:

(0.76*1.00+0.76*0.80+0.78*0.70)/(1.00+0.80+0.70) = 0.7656

For Deficiency of Spleen-Yang:

(0.80*1.00+0.00*0.90+ 0.80*0.80)/(1.00+0.90+0.80)= 0.5333

For Deficiency of Stomach-Qi:

(1.00*1.00+1.00*0.95+1.00*0.85)/(1.00+0.95+0.85) = 1.00

3.6 Summary

This chapter has described the novel model of MADHS in details, including the coordination, negotiation and cooperation methodologies used by the prototype system, and the step-by-step calculation of Certainty Factors in the joint decision-making process. The implementation of this novel model will be illustrated in the next chapter.

4 Implementation

4.1 Overview

Chapter 4 describes the implementation of the MADHS model. The MADHS model is implemented using Java language, Java Agent Development framework (JADE), Java Expert System Shell (JESS) and Fuzzy Jess package. JADE is chosen because it can provide convenient and stable services to implement a multi-agent model, including various agent management services. But MADHS didn't follow any existing communication protocols provided by JADE. Besides, while most existing multi-agent diagnostic systems only combine JESS inference engine with JADE platform, Fuzzy Jess is also used by MADHS at the same time to add fuzziness into the system.

In this chapter, the structure of the prototype system is introduced in the following order: Section 4.2 describes the agent platform of MADHS; the implementation of communication and cooperation among agents is explained in Section 4.3. Database and file management in the system is described in Section 4.4. The user interface of MADHS and its functions will be explained in Appendix A of this thesis.

4.2 Agents and Containers

The four basic types of user-defined agents in MADHS have been illustrated in Figure 1 in Section 1.3: Coordinator, Examiners (Lab Agents), Specialists and Joint Decision

Maker. Also, Figure 2 in Section 3.2 has illustrated the communication and cooperation process among those agents. The final diagnoses are achieved by combining the decisions made by registered Specialists.

Figure 13 below shows the distributed multi-agent platform of MADHS:

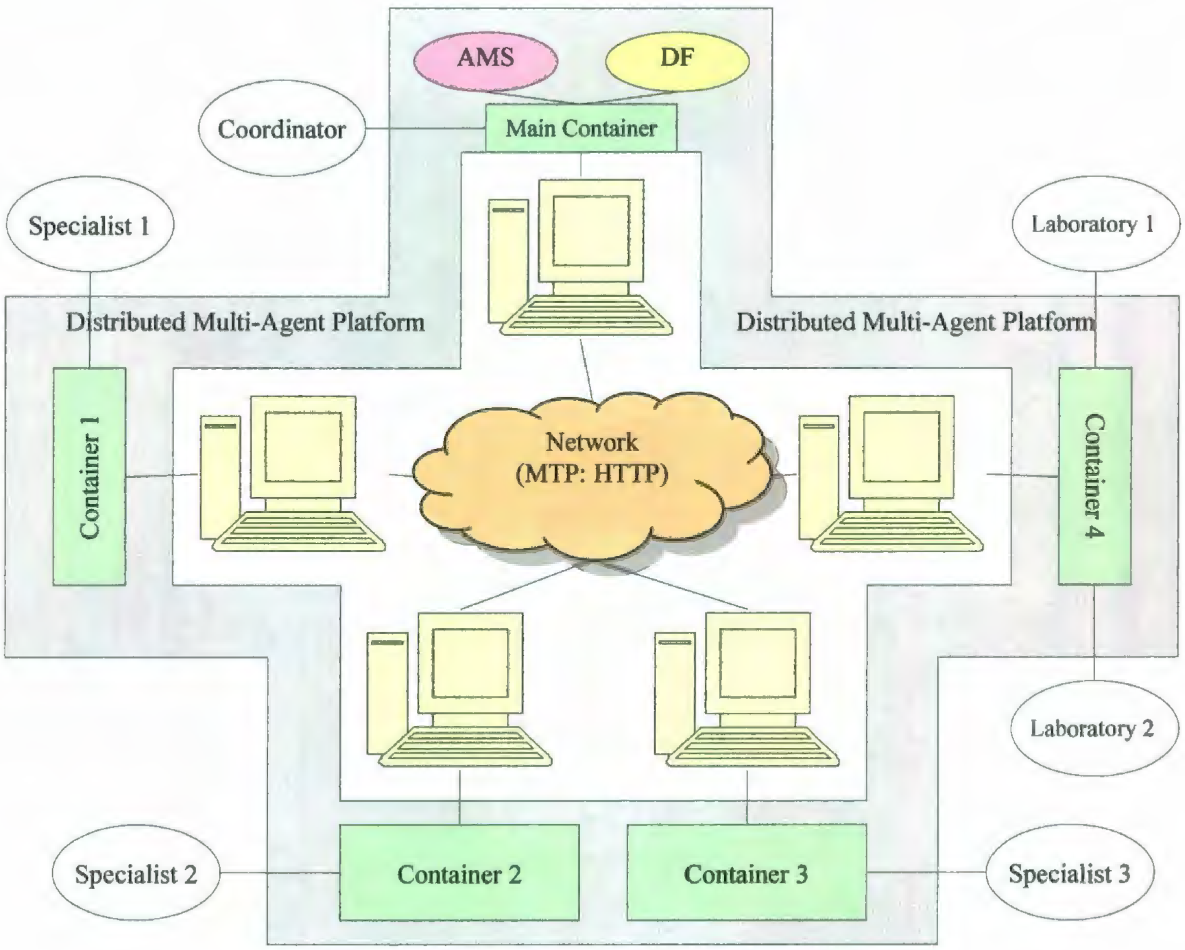


Figure 13 Agents distributed over Containers in the MADHS

JADE has already provided several useful tools to facilitate the development of a multi-agent platform, including Directory Facilitator (DF), Remote Monitoring Agent (RMA) Agent Management System (AMS) and methods to create Containers and Agents [2].

The Agent Management System (AMS) is a part of the main container of JADE. It owns an AID database and guarantees that each agent in the platform has a global unique AID (in order to facilitate the exchanging of messages). AMS also provides functions to create or kill any local/remote agents. Shown by Figure 14, AMS is automatically activated at the start time of MADHS, as part of the JADE run-time environment, together with other registered agents. So AMS and its functions can be used by the users of MADHS directly.

The programmer of MADHS is mainly responsible for writing the codes of user-defined application agents (in Java) and the knowledge bases attached to them (in Java or Jess, if necessary). The services provided by these agents are written in Java, while using the Behavior templates defined by JADE and Initiator/Responder templates defined by FIPA protocol. All services should be programmed first, and then registered to the DF agent (which will automatically start with the main container of the platform).

Besides composing user-defined agents, and using the functions of JADE to create agents and containers, the programmers are also responsible to organize the cooperation and communication among the agents.

In the development phase of MADHS, each expert (Specialist or Examiners) is created under a separate satellite container, so its code can be re-edited and re-compiled without affecting running agents in other containers. After this phase, they can be migrated and booted from one or several containers. Their services/functions will not be affected at all.

Under most circumstances, a user must boot MADHS from a computer already connected to the internet. The chosen machine is by default the host of the multi-agent platform. For example, suppose the local name of the machine is QIAOYANG, the booting command can be written as:

java jade.Boot -gui –host Coordinator: framework.coordinator.Coordinator

Then satellite containers can be created likewise. Some of the satellite containers can live on remote computers, but all satellite containers are registered and controlled by the host of the platform. Using RMA on a machine other than the host is not recommended here in MADHS, because through the remote RMA agent a user can easily destroy the whole platform.

While testing the prototype system only one computer is available. The application agents are created under one main container on the same computer. The resulting structure under Remote Monitoring Agent (RMA) is shown in Figure 14.

Using existing tools provided by JADE, the programmers can easily move the Specialists and Examiners into other Containers or other machines available.

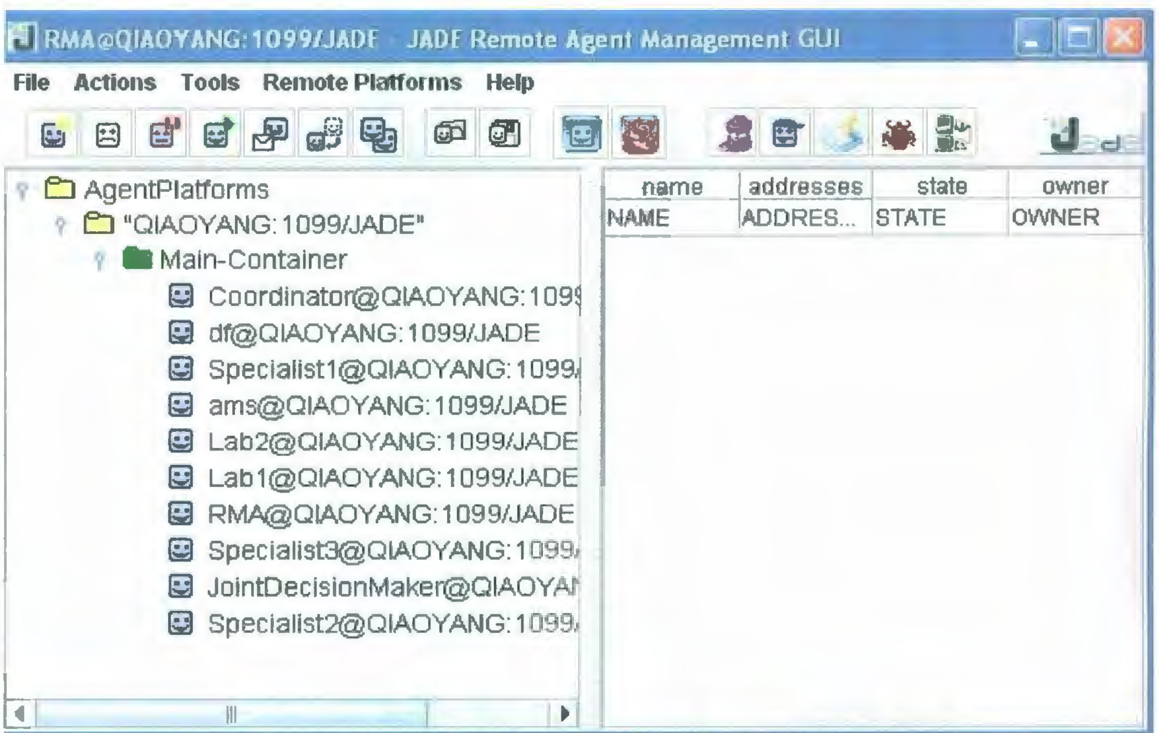


Figure 14 The appearance of the platform when testing MADHS

4.3 Inter-Platform and Intra-Platform Communication

JADE supports two Message Transport Protocols (MTPs): IIOP and HTTP. HTTP is the default inter-platform MTP.

Because by default the main container of a JADE platform runs its RMI server on part 1099, so in Figure 14, the full addresses of all the application agents have "@computer name: 1099/JADE" attached. Also by default, the HTTP MTP should be running on port 7778. The users can change the RMI port using –port command line, and change the MTP port using the options of –mtp command line.

While the inter-platform communication mainly depends on the default settings of JADE, the intra-platform communication paths of MADHS is controlled by a newly-designed Finite State Machine in the Coordinator, as shown in Figure 15 below:

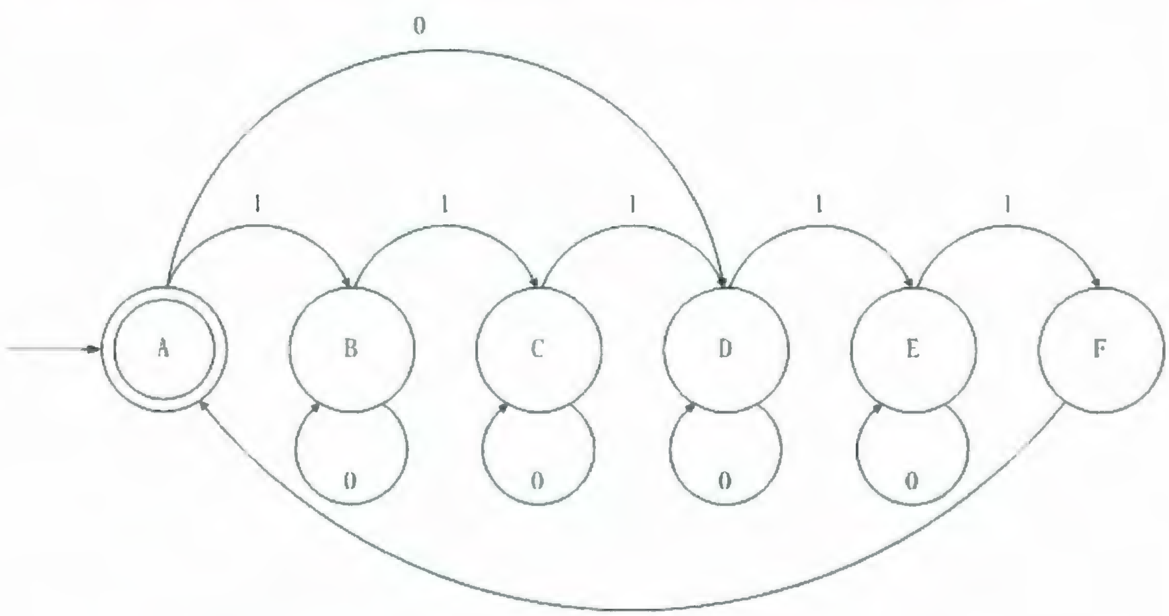


Figure 15 The finite state machine in the Coordinator of MADHS

In Figure 15, State A to State F represents the following behaviours in Table 8:

Table 8 The states of the Finite State Machine in the Coordinator

State	*Name*	*Type of Behaviour*	*Functions*
A	Start	OneShotBehaviour	Set the arguments of msgAboutInitialPlan. If successfully finished (output "1"), go to state B; else (output "0"), go to state D.
B	SendInitialPlan	AchieveREInitiator	Send msgAboutInitialPlan to the JointDecisionMaker. The JDM will add the joint decision-making method to the initial plan and send it back. If successfully finished (output "1"), go to state C; else (output "0"), repeat state B.
C	Initiator	SequentialBehaviour formed by AchieveREInitiator[][]	For every possible disease (in Traditional Chinese Medicine), send out msgAboutDiagnosis to the team members, ask them to diagnose the disease and send back the results. If successfully finished (output "1"), go to state D; else (output "0"), repeat state C.
D	Initiator2	SequentialBehaviour formed by AchieveREInitiator[]	For every possible disease (in western medicine), send out msgAboutDiagnosis2 to the team members, ask them to diagnose the disease and send back the results. If successfully finished (output "1"), go to state E; else (output "0"), repeat state D.
E	FinalDecision	SequentialBehaviour formed by AchieveREInitiator	Send msgAboutJDM to the JointDecisionMaker. Ask the JDM to make the joint decision according to the plan. If successfully finished (output "1"), go to state F; else (output "0"), repeat state E.
F	Exit	OneShotBehaviour	Exit the finite state machine. Return to the first state.

JADE uses FIPA (Foundation for Intelligent Physical Agents) standard interaction protocols to build the agent conversations. The developers using JADE can choose among various Initiator and Responder behaviours/templates. AchieveREInitiator and AchieveREResponder is a pair of Initiator and Responder provided by JADE. The programmer of MADHS chose them to implement the finite state machine shown in Figure 15. The simple methods provided by this pair of classes must be overwritten in order to handle expected and unexpected messages during conversations. Most of other Initiator/Responder templates in FIPA are designed for specific protocols. They are not very suitable in our situation.

In Table 8 above, the SequentialBehaviours in the Finite State Machine of the Coordinator are formed by AchieveREInitiators. The messages sent out by these AchieveREInitiators will use FIPA-Request protocol (FIPA Communicative Act Library Specification SC00026H). The AchieveREResponders in the Specialists and Lab Agents will respond to the messages according to this protocol.

While the agents are communicating with each other, the Sniffer Agent of JADE is recording the conversation at the same time. Every message sent to and directed from the chosen agents is recorded and displayed in the Sniffer GUI. Figure 16 on the next page shows a sequence of messages in one test of the prototype system, after the Coordinator of MADHS is started. Because this snapshot is taken during the testing, it will be explained later in Section 5.2.1.

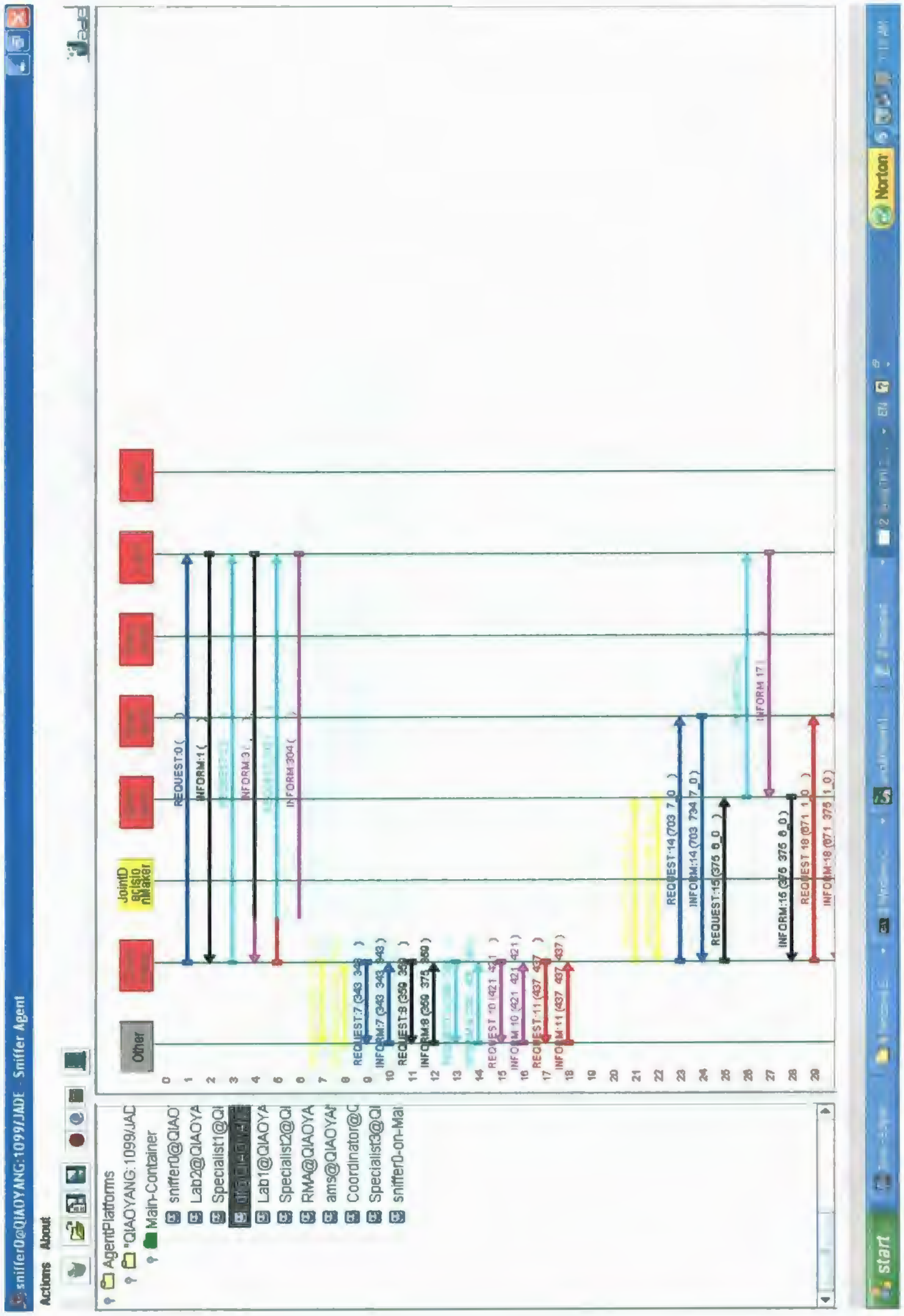


Figure 16 The Sniffer GUI when running MADHS platform

4.4 Database Management

In MADHS, JDBC technology is planned to be used. It will convert program requests to database operations. Currently, file system is used instead. Files of patients and registered experts (Specialists and Laboratories) are stored separately, as shown in Figure 17 and Figure 18.

Figure 17 Directory "patientManagement"

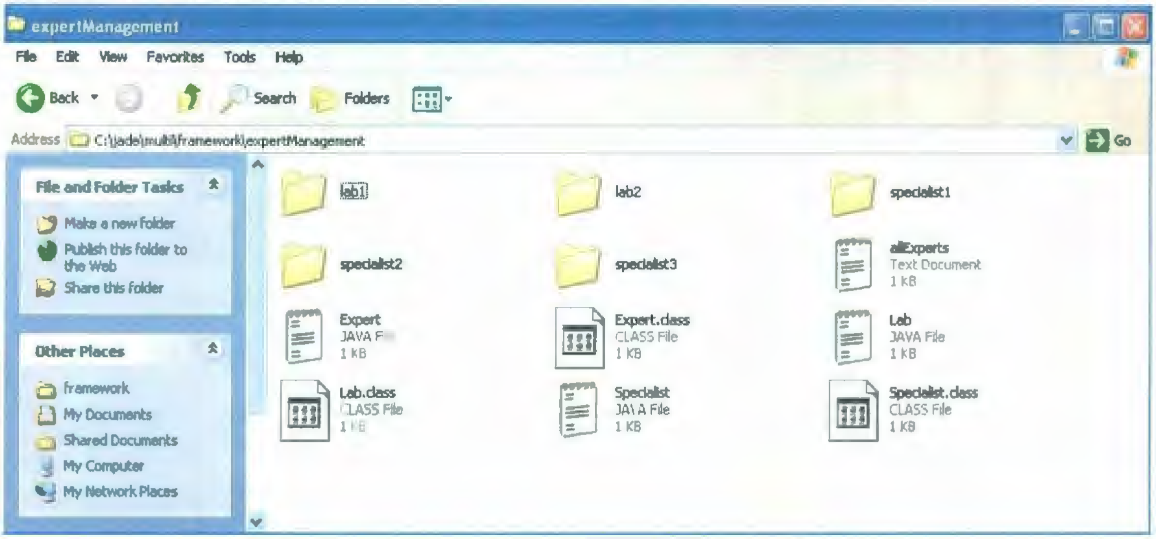


Figure 18 Directory "expertManagement"

Two menus provided to patient and expert file management will be introduced in Appendix A of this thesis, when the GUI of MADHS is introduced.

4.5 Summary

This chapter has introduced the implementation of a MADHS prototype, including the building of the agent platform, communication and cooperation among experts, database and file management. The graphic user interface of MADHS will be introduced in Appendix A. In next chapter, the prototype system will be tested using traditional Chinese and western medical examples.

5 Examples and Tests

5.1 Overview

In the previous chapter, the functions of the prototype system have been illustrated using the figures and examples. This chapter will focus on the testing of the prototype system using traditional Chinese medical examples and western medical examples.

The tests with Chinese and western medical examples are designed as follows:

Objective: To test the multi-agent diagnosing functions of the prototype system with traditional Chinese medical and western medical records (anamneses) collected from medical textbooks and medical manuals. These patients have already been diagnosed. The results produced by MADHS can be compared with the known diagnoses provided by the human experts.

Methods: To diagnose 16 cases collected from 3 different traditional Chinese medical books, including 11 men (68.75%) and 5 women (31.25%), with an average age of 32. Eight of them have stomach diseases (50%); the other eight patients have heart diseases (50%). The medical records of all 16 patients are listed in Appendix C of this thesis. These patients are chosen on purpose. The stomach diseases and heart diseases they have are very confusing.

5.2 Tests with Traditional Chinese Medical Examples and Western Medical Examples

5.2.1 Tests of Coordination and Communication

Take the diagnosing procedure of patient Wang (No. 001) for example. The Sniffer Agent of JADE is also started when the test begins, in order to monitor the traffic of messages among the participating agents. Figure 16 in Section 4.3 has illustrated the communication process after starting MADHS to diagnose this patient. This figure is analyzed and explained as follows:

(Figure 19 is the same runtime snapshot of the Sniffer Agent as Figure 16, this time with notations. In this figure, a pair of REQUEST/INFORM messages started by the same initiator will share the same message number.)

(1) The first sequence of messages shows the communication between the Coordinator and Laboratory 1. After starting MADHS, the Coordinator must diagnose patient Wang itself at first. The knowledge base of "Deficiency of Spleen-Qi" asks Lab 1 about the patient's stool test, using a REQUEST message (No. 0). It is replied by an INFORM message (No. 1) from Lab1, sending back the result of the stool test. The knowledge base of "Deficiency of Spleen-Yang" asks the same question (No. 2), and is replied by the same answer (No. 3). Then the knowledge base of "Deficiency of Stomach-Yin" asks Lab 1 about the patient's stool test, and gets the corresponding answer form the laboratory.

(2) The next sequence of messages shows the communication between the Coordinator and the DF agent of the platform. After diagnosing the current

patient, the Coordinator discovers that patient Wang has "Deficiency of Stomach-Qi" (CF 1.00), "Deficiency of Spleen-Qi" (CF 0.70) and "Deficiency of Spleen-Yang" (CF 0.21). It begins to form a team of Specialists for each of these three potential diseases.

The first three REQUEST/INFORM pairs (No. 6, No. 7 and No. 8) exchanging between the Coordinator and the DF agent represent the first round of DF search. For each of the three possible diseases, the Coordinator sends a REQUEST message to the DF agent, asking how many agents are available to participate in the joint decision-making; The DF agent replies with the number of available agents.

The following three REQUEST/INFORM pairs (No. 9, No. 10, No. 11) represent the second round of DF search required by the Coordinator. This time the Coordinator asks for the names (addresses) of the available agents for each of the three possible diseases. According to the results of this searching, the Coordinator is able to organize the next round of coordination and cooperation.

(3) The Coordinator requests the Joint Decision Maker to provide an appropriate joint decision-making method. The JDM answers with an INFORM message (No. 12), sending back revised initial plan with chosen joint decision-making method and corresponding parameters, like weights in voting.

(4) The Coordinator sends a REQUEST message (No. 13) to Specialist 1, who is available to diagnose the potential disease with the highest CF: "Deficiency of Stomach-Qi". Specialist 1 sends back the diagnosing result using an INFORM message (No. 13).

(5) A similar REQUEST message (No. 14) is sent to Specialist 2, who is also in the team for diagnosing "Deficiency of Stomach-Qi". Specialist 2 sends back its diagnosis with an INFORM message (No. 14).

(6) The Coordinator sends a REQUEST message (No. 15) to Specialist 1, who is available to diagnose the second potential disease "Deficiency of Spleen-Qi". Specialist 1 asks Lab1 to conduct the stool test (No. 16, No. 17) first. Then it will send back its diagnosis to the Coordinator using an INFORM message (No. 15). The Lab agent (Lab1) also saves a copy of the lab result to the file system, and sent another copy to the blackboard. Coordinator sends a REQUEST message (No. 18) to Specialist 2, who is also available to diagnose "Deficiency of Spleen-Qi"...

That is what can be seen from this runtime snapshot. The complete multi-agent communication process is too long; therefore it is recorded only by snapshots. Fortunately JADE has also provided the function of saving a complete message list.

Figure 19 shows that the Finite State Machine in the Coordinator (introduced in Section 4.3) functions well. It also demonstrates the hierarchical structure among the cooperative agents.

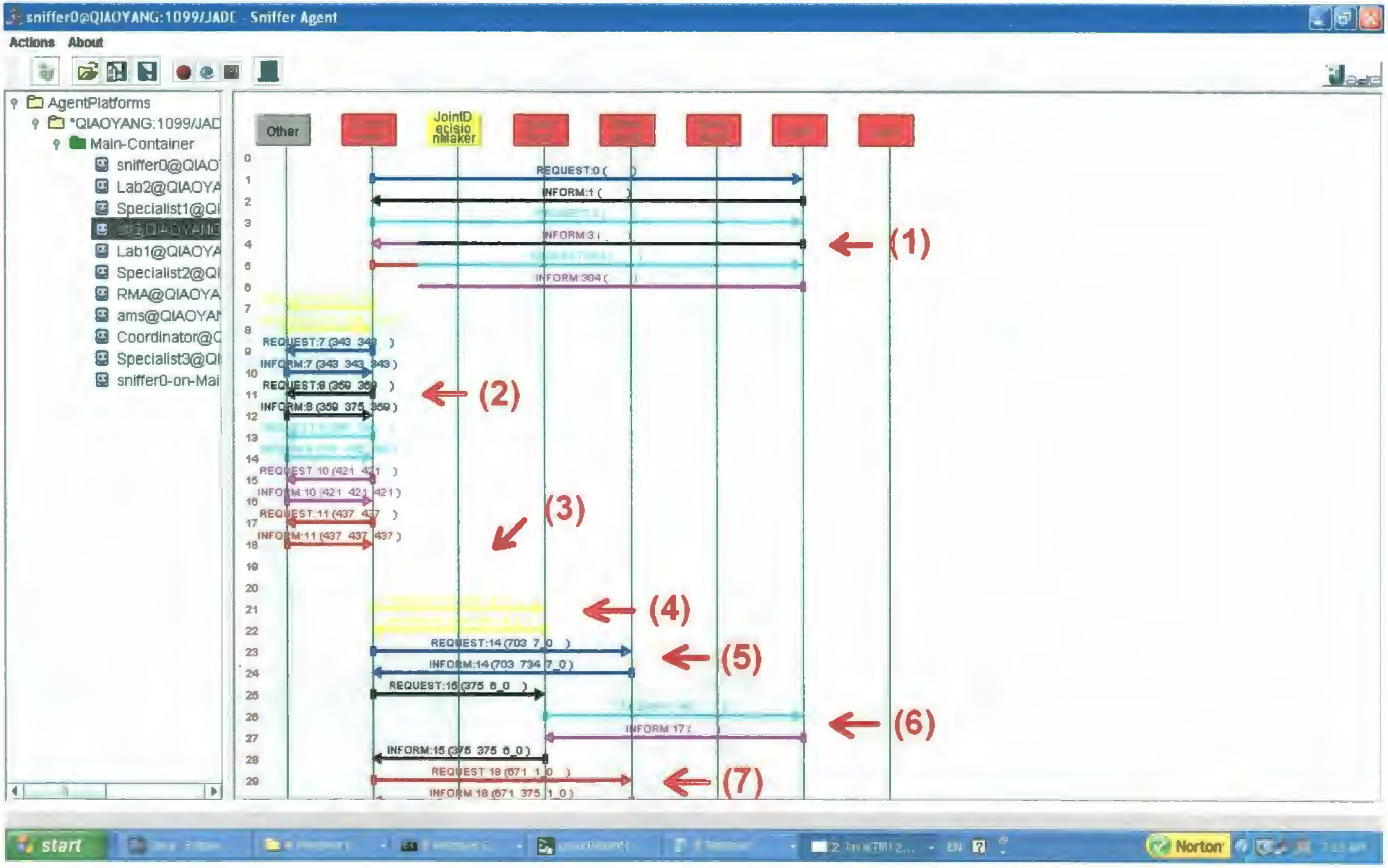


Figure 19 Analysis of the traffic of messages

5.2.2 Tests of Joint Decision-making

As mentioned earlier in Section 3.5, currently the only available joint decision-making method in MADHS is the *weighted range voting* method.

Table 9 below shows the weights assigned to the team members of each potential disease at the runtime of the tests:

Table 9 Weights assigned to the team members in the voting method

Disease/Team Members	*Coordinator*	*Specialist1*	*Specialist2*	*Specialist3*
Deficiency of Spleen-Qi	1.00	0.80	0.70	1.00
Deficiency of Spleen-Yang	1.00	0.90	0.80	1.00
Deficiency of Stomach-Qi	1.00	0.95	0.85	1.00
Deficiency of Stomach-Yin	1.00	0.90	0.70	1.00
Deficiency of Heart-Qi	1.00	0.90	1.00	1.00
Deficiency of Heart-Yang	1.00	0.95	1.00	1.00
Insufficiency of both Heart-Qi and Heart-Blood	1.00	0.90	1.00	1.00
Insufficiency of both Heart-Yin and Heart-Yang	1.00	1.00	1.00	1.00

The weights in the table are assigned by the user during the runtime of the Joint Decision Maker according to the authorities of the Specialists. The registered Specialists and their corresponding medical resources have been introduced earlier in Section 3.4.4.3.

Besides the two types of tests introduced in this section, more comparative tests between MADHS and existing diagnostic systems were also considered. But other diagnostic systems are often based on Bayesian networks and probability theory. The certainty factors produced by MADHS can not be compared with the probabilities produced by other expert systems. Moreover, other diagnostic systems usually use one unique decision tree for one main symptom of a patient, like Easy Diagnosis [20]. If a patient has *n* different main symptoms, then there will be *n* different decision trees for the user to go through during one diagnosis. And the results produced by *n* decision trees should be compared, and combined somehow, if they are different from each other. All of the 16 patient records in our tests have more than one main symptom. That means several large decision trees must be gone through from their roots in order to diagnose just one patient. And there are no existing algorithms to combine the diagnosing results. So these comparative tests were cancelled at last.

5.3 Result Analysis

In this section, the results of the testing are illustrated by tables at first, and then analyzed in detail.

5.3.1 Results of the Testing

Table 10 on the next page shows the results of the testing, where 16 different patients' anamneses are used. The patients' anamneses can be found in the file folder: **jade\multi\framework\patientManagement**, using their corresponding patient numbers.

Characters C1 to C8 in Table 10 represent following diseases in the Traditional Chinese Medicine (TCM):

C1: Deficiency of Spleen-Qi	C2: Deficiency of Spleen-Yang	C3: Deficiency of Stomach-Qi	C4: Deficiency of Stomach-Yin
C5: Deficiency of Heart-Qi	C6: Deficiency of Heart-Yang	C7: Insufficiency of both Heart-Qi and Heart-Blood	C8: Insufficiency of both Heart-Yin and Heart-Yang

Characters W1 to W8 in Table 10 represent following diseases in the western medicine:

W1: Colitis	W2: Gastritis	W3: Duodenal Ulcer	W4: Gastric Ulcer
W5: Myocarditis	W6: Premature Beat	W7: Tachycardia	W8: Rheumatic Heart Disease

Characters L1 to L10 in Table 10 represent following laboratory tests in the western medicine:

L1: Blood Test	L2: Gastric Analysis	L3: Stool Test	L4: Colonoscopy
L5: Culture for Group A Strep	L6: ECG	L7: EGD & GI	L8: Holter Monitor
L9: X Ray	L10: Ultra Sound		

The "w" before the double numbers in Table 10 represents the weight of a specialist in a joint decision-making process.

In the column "Final Diagnosis", the color green highlights the results of the highest CFs in the traditional Chinese medical diagnoses, while the color yellow highlights the results of the highest CFs in the western medical diagnoses.

Take patient 001 for example. On the first row, first column, the first line of record is "C1, T, 0.76, w1.00". It means the Coordinator has diagnosed that patient 001 has "Deficiency of Spleen-Qi", the CF of this diagnosis is 0.76, and the weight of this decision in the joint decision-making will be 1.00. The Coordinator has also diagnosed that patient 001 has "Deficiency of Spleen-Yang" and "Deficiency of Stomach-Qi". But patient 001 does not have "Deficiency of Stomach-Yin". So later in the joint diagnosis, no team of Specialists will be formed for "Deficiency of Stomach-Yin".

Table 10 Results of testing

	Coordinator	*Specialist1*	*Specialist2*	*Specialist3*	*Lab1*	*Lab2*	*Final Diagnosis*	
001	C1, T, 0.76, w1.00 C2, T, 0.80, w1.00 C3, T, 1.00, w1.00 C4, F, 1.00, w1.00	C1, T, 0.76, w0.80 C2, F, 1.00, w0.90 C3, T, 1.00, w0.95	C1, T, 0.78, w0.70 C2, T, 0.80, w0.80 C3, T, 1.00, w0.85	W1, T, 0.25, w1.00 W2, T, 0.05, w1.00 W3, T, 0.35, w1.00 W4, T, 0.05, w1.00	L1, F, 1.00 L2, F, 1.00 L3, F, 1.00	L4, F, 1.00 L7, F, 1.00 L9, F, 1.00	C1, T, 0.77 C2, T, 0.53 C3, T, 1.00	W1, T, 0.25 W2, T, 0.05 W3, T, 0.35 W4, T, 0.05
002	C1, T, 0.99, w1.00 C2, T, 1.00, w1.00 C3, F, 1.00, w1.00 C4, F, 1.00, w1.00	C1, T, 0.99, w0.80 C2, F, 1.00, w0.90	C1, T, 0.99, w0.70 C2, F, 1.00, w0.80	W1, T, 0.68, w1.00 W2, T, 0.22, w1.00 W3, T, 0.29, w1.00 W4, T, 0.29, w1.00	L1, F, 1.00 L2, F, 1.00 L3, T, 1.00	L4, F, 1.00 L7, F, 1.00 L9, T, 1.00	C1, T, 0.99 C2, T, 0.37,	W1, T, 0.68 W2, T, 0.22 W3, T, 0.29 W4, T, 0.29
003	C1, F, 1.00, w1.00 C2, T, 0.80, w1.00 C3, F, 1.00, w1.00 C4, F, 1.00, w1.00	C2, T, 1.00, w0.90	C2, T, 0.80, w0.80	W1, T, 0.43, w1.00 W2, T, 0.19, w1.00 W3, T, 0.24, w1.00 W4, T, 0.24, w1.00	L1, F, 1.00 L2, F, 1.00 L3, T, 1.00	L4, F, 1.00 L7, F, 1.00 L9, F, 1.00	C2, T, 0.87	W1, T, 0.43 W2, T, 0.19 W3, T, 0.24 W4, T, 0.24

004	C1, T, 0.38, w1.00 C2, T, 1.00, w1.00 C3, F, 1.00, w1.00 C4, F, 1.00, w1.00	C1, T, 0.38, w0.80 C2, T, 1.00, w0.90	C1, T, 0.37, w0.70 C2, T, 1.00, w0.80	W1, T, 0.55, w1.00 W2, T, 0.31, w1.00 W3, T, 0.32, w1.00 W4, T, 0.32, w1.00	L1, F, 1.00 L2, F, 1.00 L3, T, 1.00	L4, F, 1.00 L7, F, 1.00 L9, F, 1.00	C1, T, 0.38 C2, T, 1.00	W1, T, 0.55 W2, T, 0.31 W3, T, 0.32 W4, T, 0.32
005	C1, F, 1.00, w1.00 C2, F, 1.00, w1.00 C3, F, 1.00, w1.00 C4, T, 1.00, w1.00	C4, T, 1.00, w0.90	C4, T, 1.00, w0.70	W1, T, 0.13, w1.00 W2, T, 0.34, w1.00 W3, T, 0.24, w1.00 W4, T, 0.24, w1.00	L1, F, 1.00 L2, F, 1.00 L3, T, 1.00	L4, F, 1.00 L7, F, 1.00 L9, F, 1.00	C4, T, 1.00	W1, T, 0.13 W2, T, 0.34 W3, T, 0.24 W4, T, 0.24
006	C1, F, 1.00, w1.00 C2, F, 1.00, w1.00 C3, F, 1.00, w1.00 C4, T, 0.90, w1.00	C4, T, 0.90, w0.90	C4, T, 0.90, w0.70	W1, T, 0.25, w1.00 W2, T, 0.47, w1.00 W3, T, 0.48, w1.00 W4, T, 0.48, w1.00	L1, F, 1.00 L2, F, 1.00 L3, F, 1.00	L4, F, 1.00 L7, F, 1.00 L9, F, 1.00	C4, T, 0.90	W1, T, 0.13 W2, T, 0.48 W3, T, 0.33 W4, T, 0.63
007	C1, F, 1.00, w1.00 C2, F, 1.00, w1.00 C3, T, 1.00, w1.00 C4, F, 1.00, w1.00	C3, T, 1.00, w0.95	C3, T, 1.00, w0.85	W1, T, 0.25, w1.00 W2, T, 0.22, w1.00 W3, T, 0.82, w1.00 W4, T, 0.51, w1.00	L1, F, 1.00 L2, F, 1.00 L3, F, 1.00	L4, F, 1.00 L7, F, 1.00 L9, F, 1.00	C3, T, 0.98	W1, T, 0.25 W2, T, 0.22 W3, T, 0.82 W4, T, 0.51
008	C1, F, 1.00, w1.00 C2, F, 1.00, w1.00 C3, T, 1.00, w1.00 C4, F, 1.00, w1.00	C3, T, 1.00, w0.95	C3, T, 1.00, w0.85	W1, T, 0.25, w1.00 W2, T, 0.14, w1.00 W3, T, 0.35, w1.00 W4, T, 0.05, w1.00	L1, F, 1.00 L2, F, 1.00 L3, F, 1.00	L4, F, 1.00 L7, F, 1.00 L9, F, 1.00	C3, T, 1.00	W1, T, 0.25 W2, T, 0.14 W3, T, 0.35 W4, T, 0.05
009	C5, T, 1.00, w1.00 C6, T, 0.70, w1.00 C7, F, 1.00, w1.00 C8, F, 1.00, w1.00	C5, T, 1.00, w0.90 C6, T, 0.70, w0.95	C5, T, 1.00, w1.00 C6, F, 1.00, w1.00	W5, F, 1.00, w1.00 W6, F, 1.00, w1.00 W7, F, 1.00, w1.00 W8, F, 1.00, w1.00	L1, F,1.00	L5, F,1.00 L6, F, 1.00 L8, F,.1.00 L9, F, 1.00 L10, F, 1.00	C5, T, 1.00 C6, T, 0.46	None
010	C5, T, 1.00, w1.00 C6, T, 0.50, w1.00 C7, T, 0.50, w1.00 C8, F, 1.00, w1.00	C5, T, 0.50, w0.90 C6, T, 0.50, w0.95 C7, T, 1.00, w0.90	C5, T, 1.00, w1.00 C6, F, 1.00, w1.00 C7, F, 1.00, w1.00	W5, T, 0.79, w1.00 W6, T, 0.47, w1.00 W7, T, 0.25, w1.00 W8, F, 1.00, w1.00	L1, F, 1.00	L5, F, 1.00 L6, T, 1,00 L8, F, 1.00 L9, T, 1.00 L10, F, 1.00	C5, T, 0.84 C6, T, 0.33 C7, T, 0.48	W5, T, 0.79 W6, T, 0.47 W7, T, 0.25
011	C5, F, 1.00, w1.00 C6, F, 1.00, w1.00 C7, F, 1.00, w1.00 C8, F, 1.00, w1.00			W5, T, 0.81, w1.00 W6, T, 0.33, w1.00 W7, T, 0.60, w1.00 W8, F, 1.00, w1.00	L1, F, 1.00	L5, F, 1.00 L6, F, 1.00 L8, F, 1.00 L9, F,1.00 L10, F, 1.00		W5, T, 0.81 W6, T, 0.33 W7, T, 0.60
012	C5, T, 0.30, w1.00 C6, T, 0.70, w1.00	C5, T, 0.30, w0.90 C6, T, 0.70, w0.95	C5, T, 0.30, w1.00 C6, T, 0.70, w1.00	W5, T, 0.65, w1.00 W6, T, 0.67, w1.00	L1, F, 1.00	L5, F, 1.00 L6, T, 1.00	C5, T, 0.30 C6, T, 0.70	W5, T, 0.65 W6, T, 0.67

	C7, F, 1.00, w1.00 C8, F, 1.00, w1.00			W7, T, 0.25, w1.00 W8, F, 1.00, w1.00		L8, F, 1.00 L9, F, 1.00 L10, F, 1.00		W7, T, 0.25
013	C5, T, 1.00, w1.00 C6, T, 0.70, w1.00 C7, T, 0.80, w1.00 C8, T, 1.00, w1.00	C5, T, 1.00, w0.90 C6, T, 0.70, w1.00 C7, T, 0.80, w0.90 C8, T, 1.00, w1.00	C5, T, 1.00, w1.00 C6, F, 1.00, w1.00 C7, T, 0.80, w1.00 C8, T, 1.00, w1.00	W5, T, 0.35, w1.00 W6, T, 0.49, w1.00 W7, T, 0.82, w1.00 W8, F, 1.00, w1.00	L1, F, 1.00	L5, F, 1.00 L6, T, 1.00 L8, F, 1.00 L9, F, 1.00 L10, F, 1.00	C5, T, 1.00 C6, T, 0.46 C7, T, 0.80 C8, T, 1.00	W5, T, 0.35 W6, T, 0.49 W7, T, 0.82
014	C5, T, 0.30, w1.00 C6, T, 0.70, w1.00 C7, F, 1.00, w1.00 C8, T, 0.80, w1.00	C5, T, 0.30, w0.90 C6, T, 0.70, w0.95 C8, T, 0.80, w1.00	C5, T, 0.30, w1.00 C6, T, 0.70, w1.00 C8, T, 1.00, w1.00	W5, T, 0.35, w1.00 W6, T, 0.35, w1.00 W7, T, 0.43, w1.00 W8, F, 1.00, w1.00	L1, F, 1.00	L5, F, 1.00 L6, F, 1.00 L8, F, 1.00 L9, F, 1.00 L10, F, 1.00	C5, T, 0.30 C6, T, 0.70 C8, T, 0.87	W5, T, 0.35 W6, T, 0.35 W7, T, 0.43
015	C5, T, 0.30, w1.00 C6, F, 1.00, w1.00 C7, T, 0.80, w1.00 C8, F, 1.00, w1.00	C7, T, 0.80, w0.90	C7, T, 0.80, w1.00	W5, T, 0.50, w1.00 W6, T, 0.49, w1.00 W7, T, 0.25, w1.00 W8, F, 1.00, w1.00	L1, F, 1.00	L5, F, 1.00 L6, F, 1.00 L8, F, 1.00 L9, F, 1.00 L10, F, 1.00	C5, T, 0.28 C7, T, 0.80	W5, T, 0.50 W6, T, 0.49 W7, T, 0.25
016	C5, T, 0.30, w1.00 C6, F, 1.00, w1.00 C7, F, 1.00, w1.00 C8, F, 1.00, w1.00	C5, T, 0.30, w1.00	C5, T, 0.30, w1.00	W5, T, 0.36, w1.00 W6, T, 0.67, w1.00 W7, T, 0.25, w1.00 W6, F, 1.00, w1.00	L1, F, 1.00	L5, F, 1.00 L6, T, 1.00 L8, F, 1.00 L9, F, 1.00 L10, F, 1.00	C5, T, 0.30	W5, T, 0.50 W6, T, 0.67 W7, T, 0.25

5.3.2 Analysis of the Results

The results of the testing are analyzed as follows:

1) **MADHS can diagnose.**

Because the anamneses used in the tests are taken from published medical manuals and textbooks, the most certain diagnoses produced by MADHS can be compared with the results provided by human experts:

Table 11 Diagnosing results by human experts vs. diagnosing results by MADHS

Patient No.	Diagnosed by Human Experts		Diagnosed by MADHS	
	Chinese	Western	Chinese	Western
001	C1	N/A	C3	W1
002	C1	W1	C1	W1
003	C2	W1	C2	W1
004	C2	N/A	C2	W1
005	C4	W2	C4	W2
006	C4	W4	C4	W4
007	C3	W3	C3	W3
008	C3	W2	C3	W3
009	C5	N/A	C5	N/A
010	C5	W5	C5	W5
011	C6	W5	N/A	W5
012	C6	W6	C6	W6
013	C8	W7	C5/C8	W7
014	C8	W8	C8	W7
015	C7	N/A	C7	W5
016	C7	W6	C5	W6

The results marked by color grey in the table and explained as follows:

i. Diagnosed by human expert, patient 001 has disease C1 (Deficiency of Spleen-Qi). Diagnosed by MADHS, disease C3 (Deficiency of Stomach-Qi) is with the highest certainty factor. According to the available information in patient 001's anamnesis, the calculated CF of C3 is really higher than the calculated CF of C1.

ii. Patient 011 is known to have disease C6 (Deficiency of Heart-Yang), which can not be diagnosed out by MADHS. Because the corresponding traditional Chinese medical manual [4] has not given enough information of this patient for us to diagnose, this situation is considered normal here.

iii. Patient 013 is diagnosed to have C8 (Deficiency_of _Heart_Qi). But MADHS gives C5 too. That's because Patient 013 just doesn't have one of the minor symptoms of C5 (white tongue fur). His medical record satisfies all other diagnostic criteria of C5.

iv. Patient 016 is known to have disease C7 (Insufficiency of both Heart-Qi and Heart-Blood), but the disease with the highest calculated CF is C5 (Deficiency of Heart-Qi). This is because patient 016's anamnesis can not satisfy main rule 2 in the rule base of disease C7. He does not have two or more minor symptoms of this disease according to the given information.

v. Patient 008 is known to have disease W2 (Gastritis). But MADHS gives W3 (Duodenal Ulcer) the highest CF. Patient 008 doesn't have one of the major symptoms of W2. If a patient has W2, he/she is supposed to have worse stomach-ache after eating. On the contrary, patient 008 feels better after his meals. So the diagnosing result of MADHS makes more sense in this case.

vi. Diagnosed by human expert, patient 014 has W8 (Rheumatic Heart Disease), but diagnosed by MADHS she has W7 (Tachycardic) with the highest CF. Actually patient 014 is known to have W7 at the same time. The information given in her anamnesis is enough to diagnose W7, but not enough to diagnose W8. So this diagnosing result can be considered normal here.

2) **MADHS can differentiate between confusing diseases.**

As mentioned previously, the eight patients of stomach diseases and another eight patients of heart diseases being diagnosed in the tests are chosen on purpose. The stomach diseases involved have very similar symptoms, so do the heart diseases. In

Table 12, the CF differences between potential diseases are listed out to show that MADHS can differentiate between similar diseases.

Table 12 CF differences between potential diseases calculated by MADHS

Patient No.	**Diagnosed by MADHS**		**CF Differences between Potential Diseases ***	
	Chinese	**Western**	**Chinese**	**Western**
001	C3,C1,C2	W1,W3,W2,W4	0.23, 0.24	0.10, 0.30, 0.00
002	C1,C2	W1,W3,W4,W2	0.62	0.39, 0.00, 0.07
003	C2	W1,W3,W4,W2	N/A	0.19, 0.00, 0.05
004	C2,C1	W1,W3,W4,W2	0.10	0.23, 0.00, 0.01
005	C4	W2,W3,W4,W1	N/A	0.10, 0.00, 0.11
006	C4	W4,W2,W3,W1	N/A	0.15, 0.15, 0.20
007	C3	W3,W4,W2,W1	N/A	0.31, 0.26, 0.03
008	C3	W3,W1,W2,W4	N/A	0.10,0.11,0.05
009	C5,C6	None	0.54	N/A
010	C5,C7,C6	W5, W6, W7	0.36, 0.15	0.32, 0.22
011	N/A	W5, W7, W6	N/A	0.21, 0.27
012	C6,C5	W6, W5, W7	0.40	0.02, 0.40
013	C8,C5,,C6,C7	W7, W6, W5	0.00, 0.20, 0.34	0.33, 0.14
014	C8,C6,C5	W7, W6, W5	0.17, 0.40	0.00, 0.08

015	C7, C5	W5, W6, W7	0.52	0.01, 0.24
016	C5	W6, W5, W7	N/A	0.17, 0.25

The "CF differences between potential diseases" columns in Table 12 are calculated based on the CFs given in Table 10. For example, in Table 10, patient 001 is diagnosed to have C3 (CF 1.00), C1 (CF 0.77) and C2 (CF 0.53). The CF difference between C3 and C1is 1.00 - 0.77 = 0.23. The CF difference between C1 and C2 is 0.77 - 0.53 = 0.24. Thus, 0.23 and 0.24 are listed as the "CF differences for Chinese diseases" in Table 12 for patient 001.

As can be seen from Table 12, MADHS can easily differentiate between confusing diseases in Traditional Chinese Medicine. In Table 12, for Traditional Chinese Medicine, 6 out of 16 patients (37.5%) have only one potential disease in their diagnoses. 5 out of 16 patients (31.25%) are diagnosed to have two potential diseases with obviously different certainty factors. Another 3 out of 16 patients (18.75%) have three potential diseases with different certainty factors. MADHS only can not differentiate between C8 and C5 on one of those 16 patients (6.25%).

For eight confusing diseases in western medicine, MADHS can easily find out which disease has the highest CF, as shown in Table 11. However, it can not differentiate between stomach disease W3 and W4 in most cases. In Table 12, the CF differences between W3 and W4 are close to zero. That's because the criteria of these two diseases are very similar; only several minor symptoms are not the same. For example, a patient with W3 (Duodenal Ulcer) usually has abdominal pain before the meal or at night in right abdomen; while a patient with W4 (Gastric Ulcer) usually has abdominal pain after the meal in left abdomen. The patient medical records from the manuals are not detailed enough to provide all those information needed.

The rule bases of W3 and W4 have been modified to emphasize their special features. Higher weights are given to the unique symptoms. The testing results are still not very satisfactory. Only two ways can be used to solve this problem in the future:

1. Use far more detailed patient records from other medical manuals; or ask the patients to fill out the questionnaires/GUI dialogs by themselves.

2. Create another Specialist. Its responsibility is to differentiate very confusing diseases right before or after the Joint Decision Maker gives out the final diagnosis. But an appropriate and detailed symptom-based medical manual or an eligible human expert must be found first.

6 Evaluation and Conclusion

6.1 Overview

The last chapter of this thesis will evaluate our research work in general from many aspects.

6.2 Revisiting the Research Objectives

The objectives of this research listed in Section 1.4 are revisited as follows:

1) **To survey the current state of multi-agent medical diagnostic systems and to find their advantages and limitations;**

   In Section 2.2, the existing models of e-medicine, on-line medical diagnostic systems like EasyDiagnosis, multi-agent diagnostic systems like ALIAS and simulations of hospitals have been discussed. Other related current research works were also surveyed in Chapter 2, such as multi-agent coordination, negotiation, joint decision-making and uncertainty management.

2) **To develop a novel model of coordination and negotiation among agents;**

   This objective was met in Section 3.2, where the novel model of coordination and negotiation in MADHS has been described in details.

3) **To develop the methodologies and algorithms of our multi-agent medical diagnostic system;**

   Chapter 3 has discussed many aspects considering the designing of the prototype system MADHS, such as the appropriate communication protocols, the knowledge engineering methods, the algorithms of CF calculation and the inference mechanism used in the knowledge bases.

4) **To develop, test and evaluate the prototype system MADHS in a range of practical domains.**

   Chapter 4 has described the implementation of the novel model presented earlier. The resulting prototype system was tested by text-book diagnostic examples in Chapter 5, where the results were also analyzed.

6.3 Research Contributions

6.3.1 A Novel Model for Multi-agent Diagnosis

In this thesis, a novel coordination and negotiation model for multi-agent diagnosis has been presented.

In order to facilitate the cooperation among multiple agents, *organizational structuring* is applied in MADHS. The roles (types) of the agents, such as Coordinator, Specialist, Examiner or Joint Decision Maker are pre-defined before the registration begins. When an agent and its services are being registered into a running platform, its role in future coordination and joint decision-making is decided at the same time.

However, the definition of the Coordinator in MADHS is comparatively more flexible than those fixed agent roles in *organizational structuring.* The Coordinator itself is an ordinary Specialist with two extra structures: a finite state machine to control the coordination and negotiation among agents, and a scheduler to control the reads from and writes to the blackboard. Any registered Specialists can be appointed as the Coordinator of a platform by the user, if only it has a finite state machine and a scheduler attached. Just remember that one platform should only have one Coordinator at the run time.

The Joint Decision Maker is another newly defined agent/structure in the model of MADHS. Existing decision-making methods like *range voting* are combined with social decision-making concepts like authority and weights to suit the special needs of multi-agent diagnoses.

6.3.2 Fuzzy Inference Structures

A classical decision tree for medical diagnosis usually starts from one of the main symptoms of a certain disease. This kind of tree structure is more suitable for western medical diagnosis, and is used frequently in western medical manuals. But in those traditional Chinese medical manuals, the main and minor criteria, must-have and may-have symptoms are clearly classified. Using these symptom-based features, the programmer of MADHS defines various types of Nodes and Rules. Fuzziness is also combined into inference techniques to facilitate the reasoning process. The Nodes and Rules in our knowledge bases can be either crisp or fuzzy.

6.3.3 The Calculation of Certainty Factors for Medical Diagnoses

A new CF calculation method for MADHS was introduced in Section 3.4.3 and 3.4.4. Different from those equations used by MYCIN and FuzzyCLIPS (introduced in Chapter

2), the CF calculation in MADHS emphasizes the weight of a symptom in the decision-making. Generally, Major criteria are considered more important than Minor criteria; Must-Have symptoms are considered more important than May-Have symptoms. Thus the CFs of the symptoms are associated with their importance in the decision-making.

6.3.4 The Implementation of MADHS Prototype

A prototype system of MADHS has been implemented according to the presented novel model. Compared with existing medical diagnostic systems discussed in Chapter 2, this prototype system is designed specifically for joint diagnoses performed by multiple agents.

First of all, the GUI of the prototype system is carefully designed. It allows the users of MADHS to mange the patient database and expert database with ease. It also provides functions like agent registration, service registration and rule base designing to facilitate the building of a multi-agent platform.

Secondly, the prototype system has been tested by 16 text-book cases in both traditional Chinese and western medicine. The results of the testing have been analyzed in Chapter 5.

6.4 Future Research

An advanced medical diagnostic system may need 5 years or more to be fully developed. The novel model and its prototype system presented in this thesis need further development in the following aspects:

1) database management

As has been mentioned earlier, JDBC technology is planned to be applied in MADHS. An appropriate DB driver and corresponding DBMS must be carefully chosen to create complete databases. After that, the anamneses of patients and information of experts can be managed more efficiently.

2) data mining

Currently in MADHS, the certainty factors assigned to the rules and facts are provided by the human experts (medical manuals) and the knowledge engineer. Users can change those default CF settings through the GUI. All CF thresholds in the rule bases are also set to default values by the knowledge engineer, and can also be changed by the users at the run time.

In future research, using data mining techniques, the system can be trained and learn to provide the most proper CF thresholds for every decision-making steps.

3) planning

The global and partial planning among cooperative and distributed agents can be a very good future research topic.

4) further tests on distributed machines

Tests on distributed machines of the same platform are included in the future plan, as well as tests with medical records of more patients.

6.5 Conclusion

In this thesis the current research works on medical diagnostic systems have been studied.

A novel coordination and negotiation model for helping multi-agent medical diagnosis has been proposed. This model was then implemented to build a prototype system called MADHS, which was tested by text-book diagnostic examples.

It has also proposed a new method to calculate the certainty factors in medical diagnoses. Fuzziness is combined into the rule bases at the same time.

Lastly, this thesis analyzed the results of the testing, evaluated the prototype system and proposed future research objectives.

References

[1] Berkow R., etal. (1997), the Merck Manual of Medical Information, ISBN 0-911910-87-5

[2] Caire G. (2003), JADE Tutorial (JADE Prgramming for Beginners), Telecom Italia.

[3] Charles L. Forgy (1982), Rete: A Fast Algorithm for the Many Pattern/Many Object Pattern Match Problem, Artificial Intelligence 19, pp.17-37

[4] Cheng S. E. and Xia H. S. (1993), Symptom-based Diagnosis and Treatment in Traditional Chinese Medicine（《中医证候诊断治疗学》）, Beijing: Beijing Science &Technology Press. ISBN: 753041201

[5] Ciampolini A., Mello P. and Storari S. (2002) Distributed Medical Diagnosis with Abductive Logic Agents. AI*IA Notizie - Anno XV (3), Settembre 2002, pp. 12-21.

[6] Fischbach F. T. (1996), A Manual of Laboratory & Diagnostic Tests, ISBN:039755186X

[7] Flores-Méndez R. A. (2002) Modeling Agent Conversations for Action. Ph.D. Thesis, Department of Computer Science, University of Calgary, Canada

[8] Guidelines for the Diagnosis of Rheumatic Fever. Jones Criteria, 1992 update. Special Writing Group of the Committee on Rheumatic Fever, Endocarditis, and Kawasaki Disease of the Council on Cardiovascular Disease in the Young of the American Heart Association. JAMA 268 (15): 2069-73. PMID 1404745.

[9] Hogg L. and Jennings N. R. (1999) Variable Sociability in Agent-Based Decision Making. Lecture Notes in Computer Science, 6th International Workshop on Intelligent Agents VI, Agent Theories, Architectures, and Languages (ATAL), pp. 305 – 318.

[10] Klügl F., Oechslein C., Puppe F., and Kirn S. (1999) Multi-agent Simulation of Diagnostic and Logistic Processes in Hospitals. Tagungsband zum Workshop, Intelligente Softwareagenten und betriebswirtschaftliche Anwendungsszenarien, Ilmenau.

[11] Nwana H. S., Lee L. and Jennings N. R. (1996), Coordination in Software Agent Systems, British Telecommunications Technology Journal, Vol. 14, No. 4.

[12] Orchard B. (2004), FuzzyCLIPS Version 6.10d User's Guide. Integrated Reasoning Group, Institute for Information Technology. Ottawa, ON: National Research Council Canada.

[13] OuYang Q., Wen Z. and Hu R. (2000), Journal of Clinical Internal Medicine（《临床内科杂志》）, Vol. 05, 2000.

[14] Pagana K. D. and Pagana T. J.(2002), Mosby's Manual of Diagnostic and Laboratory Tests, ISBN: 032301609X

[15] Pearl J. and Russel S. (2000) Bayesian Networks,UCLA Cognitive Systems Laboratory, Technical Report (R-277), In M.A.Arbib (Ed.), Handbook of Brain Theory and Neural Networks, Cambridge, MA: MIT Press, 157-160

[16] Shakshuki E. and Jun Y. (2004) Multi-agent Development Toolkits: An Evaluation IEA/AIE 2004, LNAI 3029, pp. 209-218

[17] Shortliffe E. H., Davis R., Axline S. G., Buchanan B. G., Green C. C. and Cohen S. N. (1975) Computer-based consultations in clinical therapeutics: explanation and rule acquisition capabilities of the MYCIN system, Comput Biomed Res., 8(4):303-20

[18] Sibbel R. and Urban C. (2001) Agent-Based Modeling and Simulation for Hospital Management. Cooperative Agents, Kluwer academic publishers, Boston.

[19] Sobrado F. J., Pikatza J. M., Larburu I. U., Garcia J. J. and D. L. de Ipiña (2004) Towards a Clinical Practice Guideline Implementation for Asthma Treatment. Current Topics in Artificial Intelligence, Lecture Notes in Computer Science, Volume 3040/2004, pp.587 – 596

[20] Sturman H. R., etal. (2007, March 18, Last ed.), Easy Diagnosis, http://easydiagnosis.com, [Online].

[21] Tian J. and Tianfield H. (2003). A Multi-agent Approach to the Design of an E-medicine System. MATES 2003, LNAI 2831, pp. 85-94.

[22] Wemmenhove B., Mooij J. M., Wiegerinck W., Leisink M., Kappen H. J.and Neijt J. P. (2007) Inference in the Promedas Medical Expert System, Artificial Intelligence in Medicine, Lecture Notes in Computer Science, ISBN: 0302-9743

[23] Werkman K. J. (1990), Knowledge-based model of negotiation using shareable perspectives, Proc. of the 10th International Workshop on DAI, Texas.

[24] Wu D. and Ke X. (1991) Traditional Chinese Medical Therapies of Obstinate Diseases（《难治病的良方妙法》）, China Medico-Pharmaceutical Sciences & Technology Publishing House, ISBN: 7-5067-0304-1/ R·0259

[25] Zadeh L. A. (1965), Fuzzy Sets, Information and Control, 8:338-353.

[26] Zhang E. (1990) Clinical of Traditional Chinese Medicine (I), Publishing House of Shanghai College of Traditional Chinese Medicine, ISBN: 7-81010-129-2/R·128

[27] Zhao J. D., Zhang J. R. and Zhang Z. (1996), Symptom-based Diagnosis in Traditional Chinese Medicine （《中医征候鉴别诊断学》）, China Academy of Chinese Medical Sciences, ISBN: 7-117-01100-9/R·1101

Appendix A

User Interfaces and Their Functions

In this appendix, the user interfaces of MADHS and their corresponding functions are introduced. Figure 20 shows the general menu of MADHS:

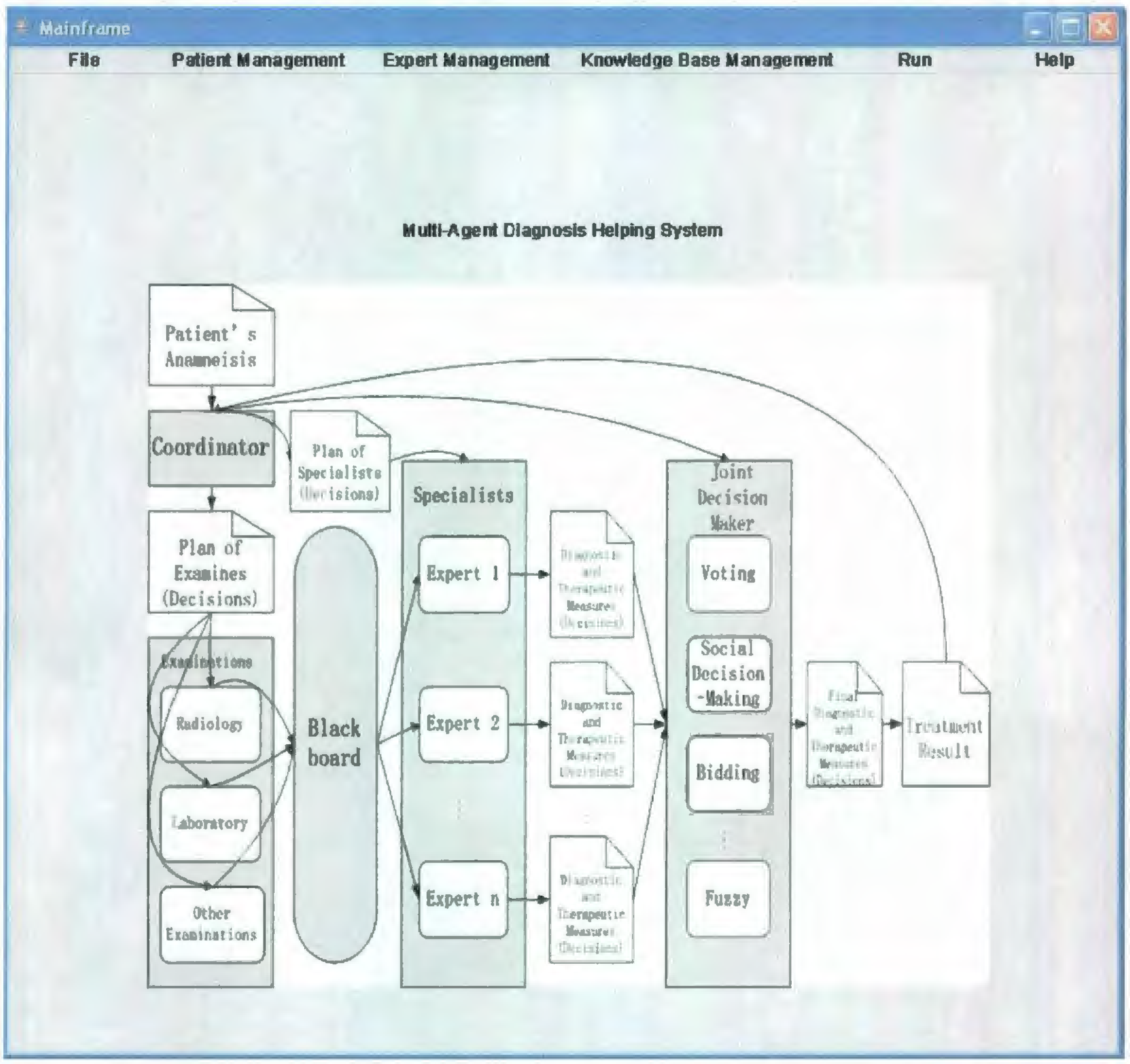


Figure 20 The general view of MADHS at the run time

"File" Menu and File Management

The "File" Menu is shown in Figure 21:

Figure 21 "File" menu

If the user clicks on the menu item "Open", a dialog will appear, as shown in Figure 22. He or she can open, edit or save any java program, patient record or expert record through this dialog.

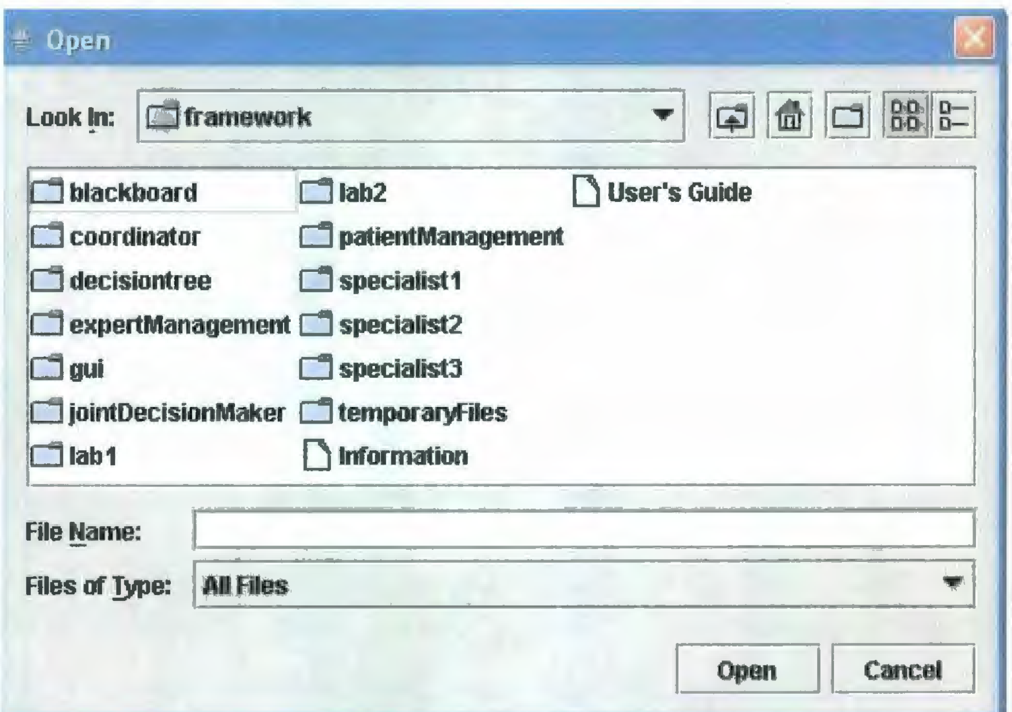


Figure 22 "Open" dialog

If the user clicks on the menu item "Exit" on Figure 21, the current running Java Virtual Machine will terminate.

"Patient Management" Menu

New versions of MADHS with JDBC (Java Database Connectivity) and SQL databases will be distributed as soon they are available. Currently, the patient information, agent information (of Chinese Specialists, western Specialists and Lab Agents) and results of diagnosing are still stored using a file system.

The "Patient Management" Menu is shown in Figure 23:

Figure 23 "Patient Management" menu

If the user clicks on the menu item "Add a New Patient", a dialog will appear as shown in Figure 24 . By filling in this form he or she can create a new patient record and save it.

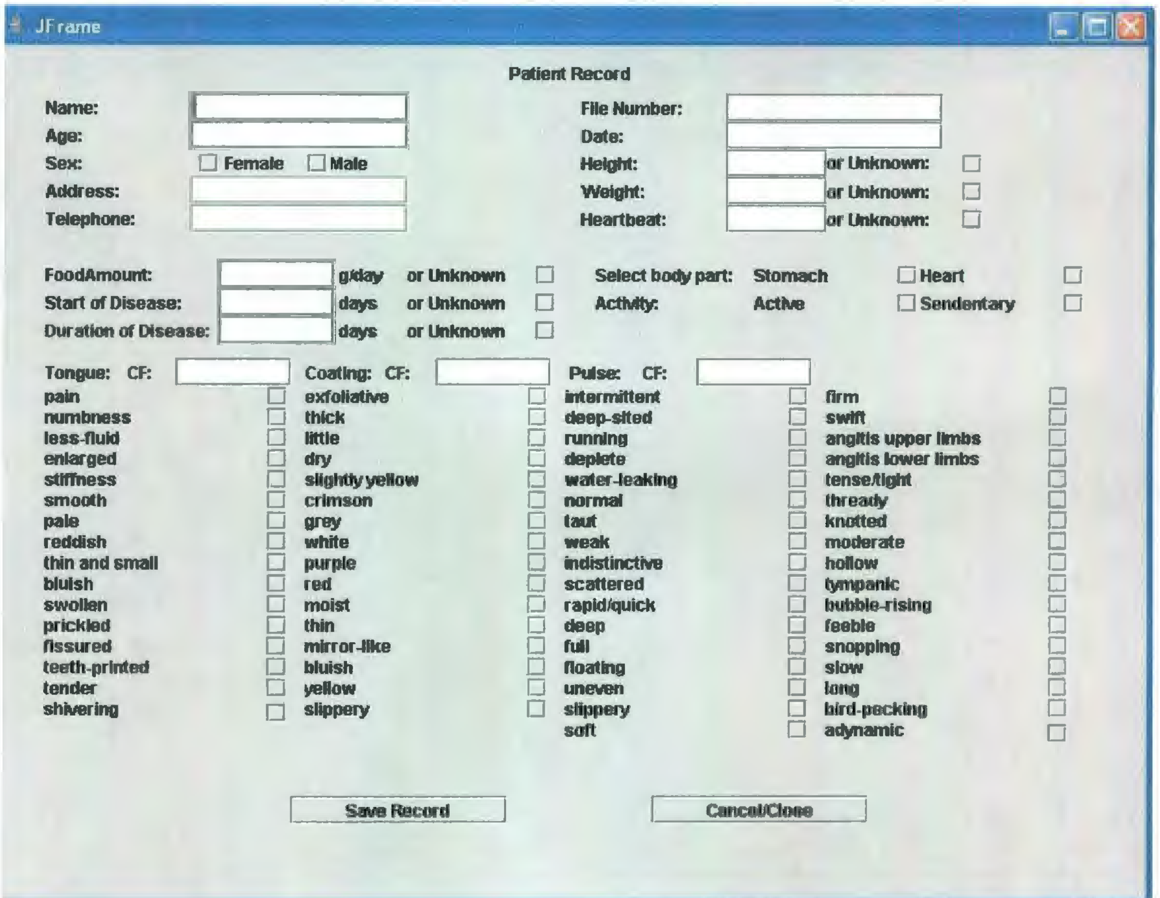


Figure 24 Add a new patient record

If the user clicks on the menu item "Delete Existing Patient", a dialog will appear as shown in Figure 25. From the list of patients the user can select the records he or she wants to delete.

A similar dialog will appear if you click on menu item "Choose Current Patient", as shown in Figure 26 on the next page. The selected patient is the one to be diagnosed by the experts in the system.

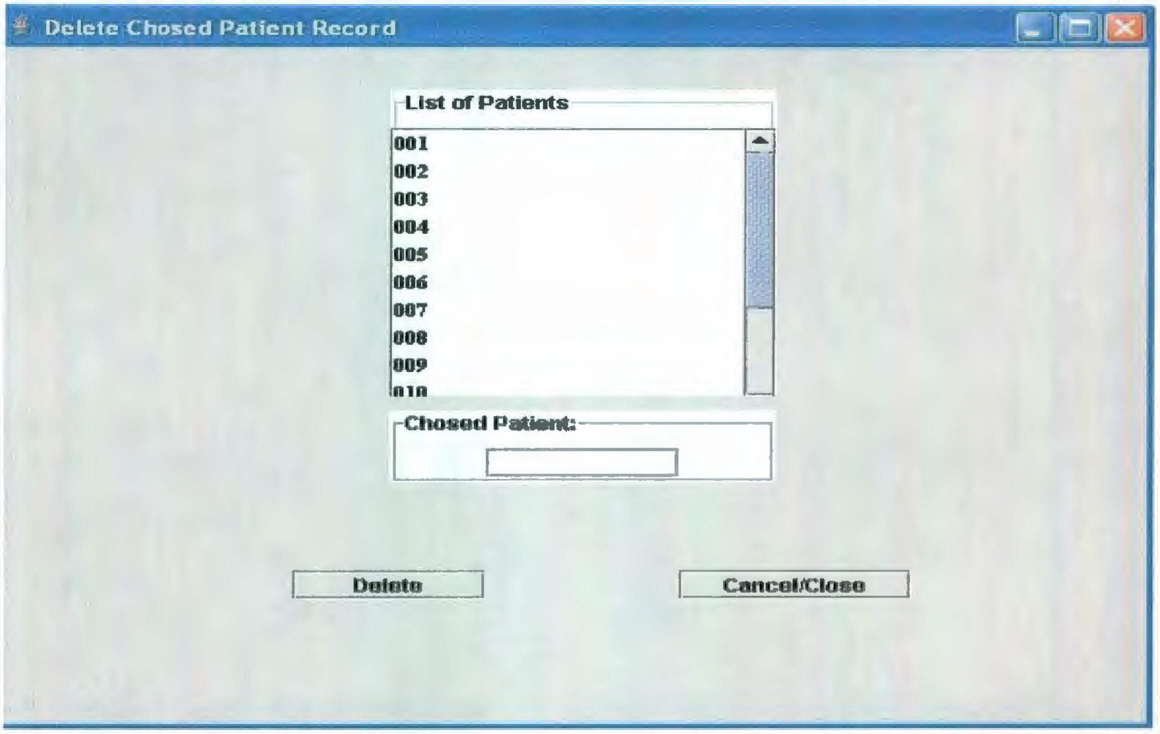


Figure 25 Delete chosen patient record

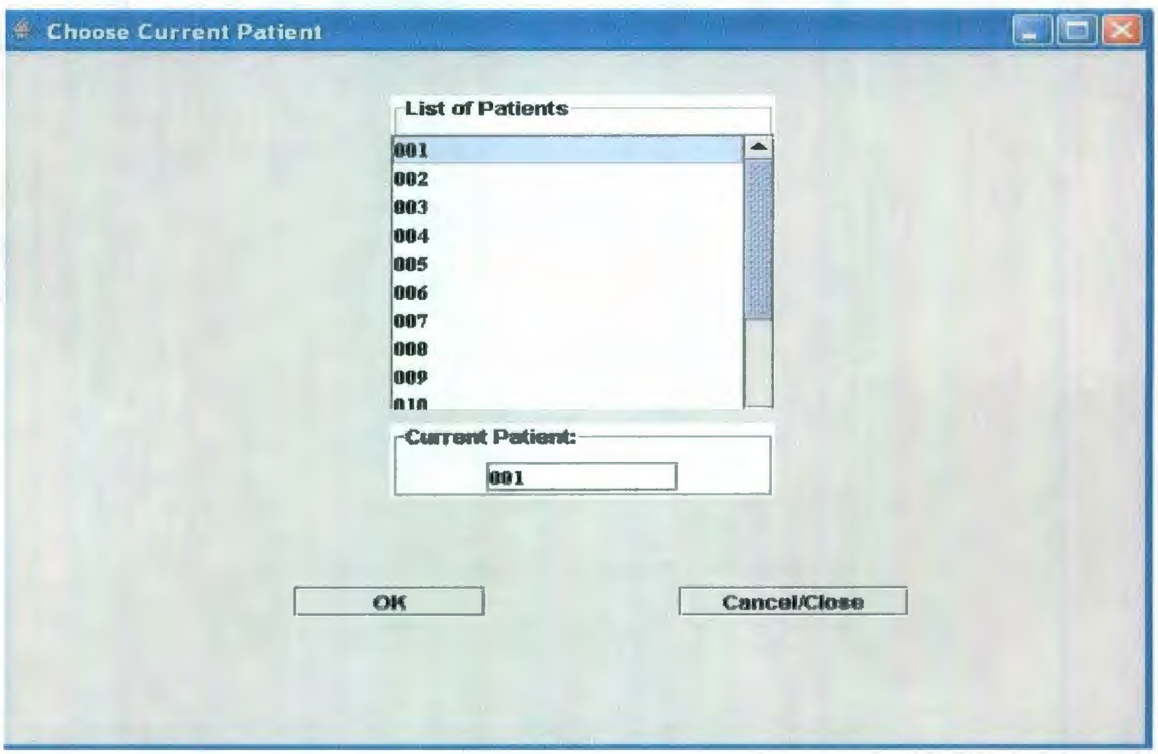


Figure 26 Choose the "Current Patient" from the list

“Expert Management” Menu

The “Expert Management” Menu is shown in Figure 27:

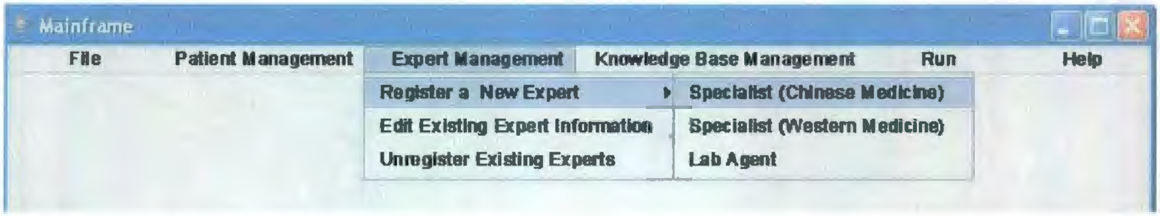


Figure 27 “Expert Management” menu

If the menu items “Specialist (Chinese Medicine)”, “Specialist (Western Medicine)” or “Lab Agent” are clicked, similar dialogs will be prompted as shown in Figure 28.

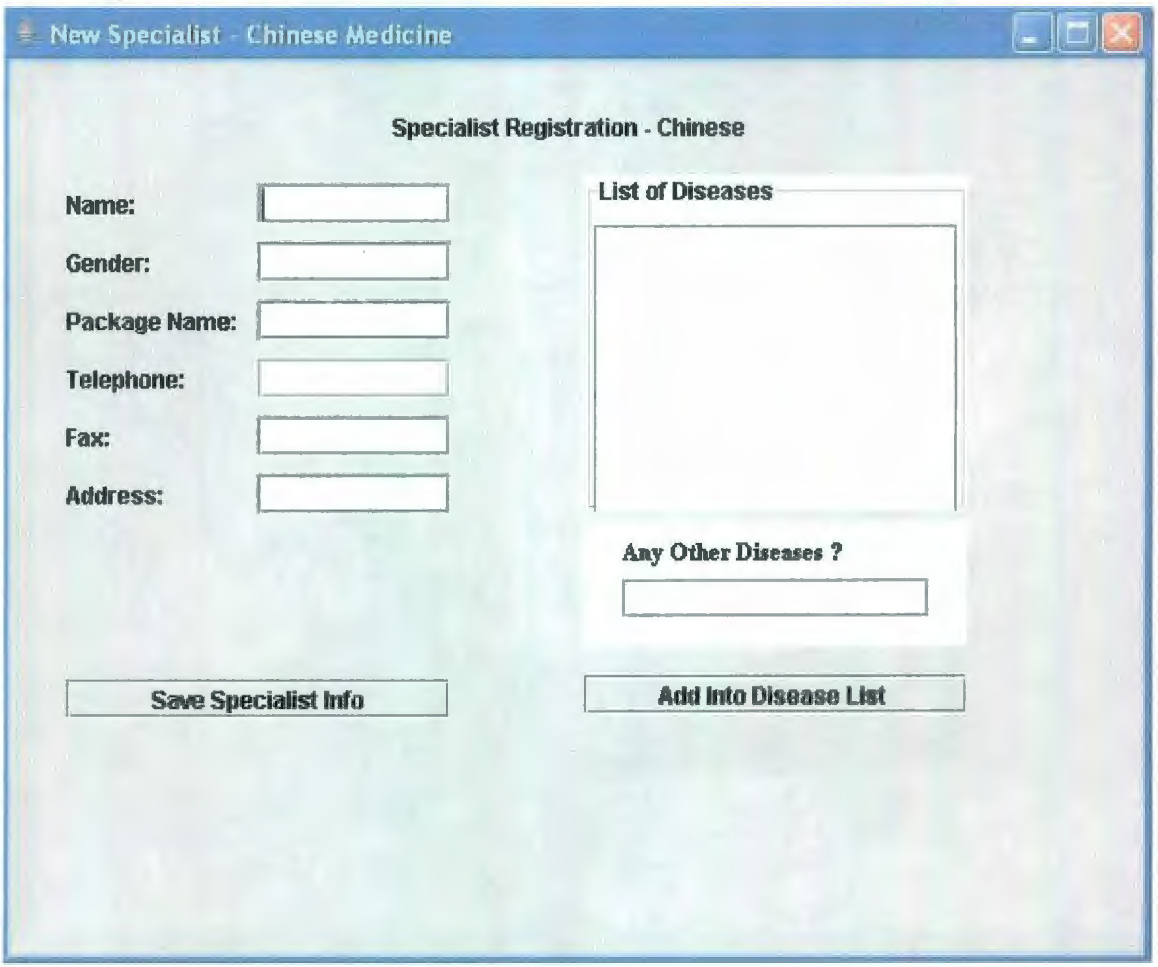


Figure 28 Registration of a Specialist

If the user clicks on the menu item "Unregister Existing Experts", a dialog will appear as shown in Figure 29. The expert selected will be deleted from the system database/file system.

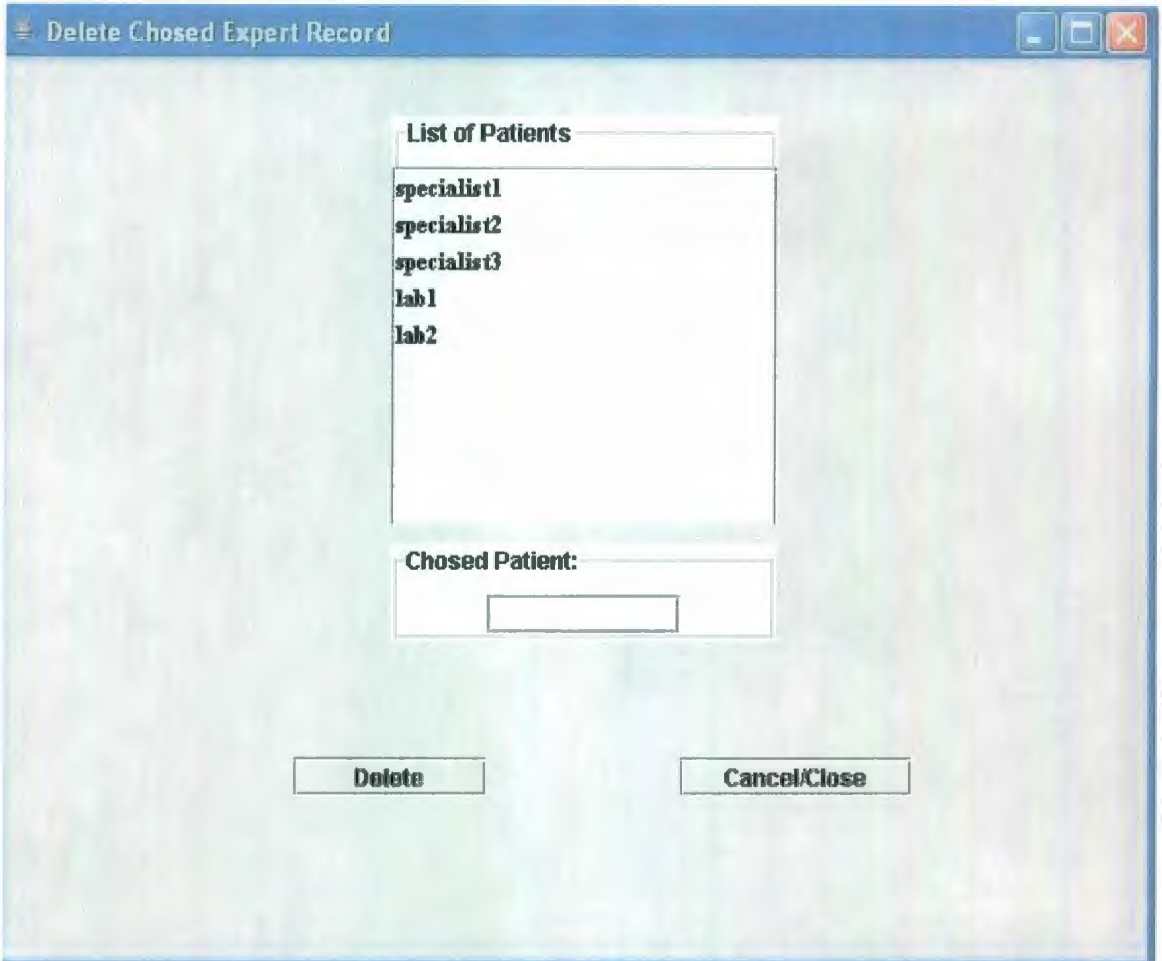


Figure 29 Unregistering existing experts

JADE also provides the function of adding new services to the agents, as shown in Figure 30 on the next page:

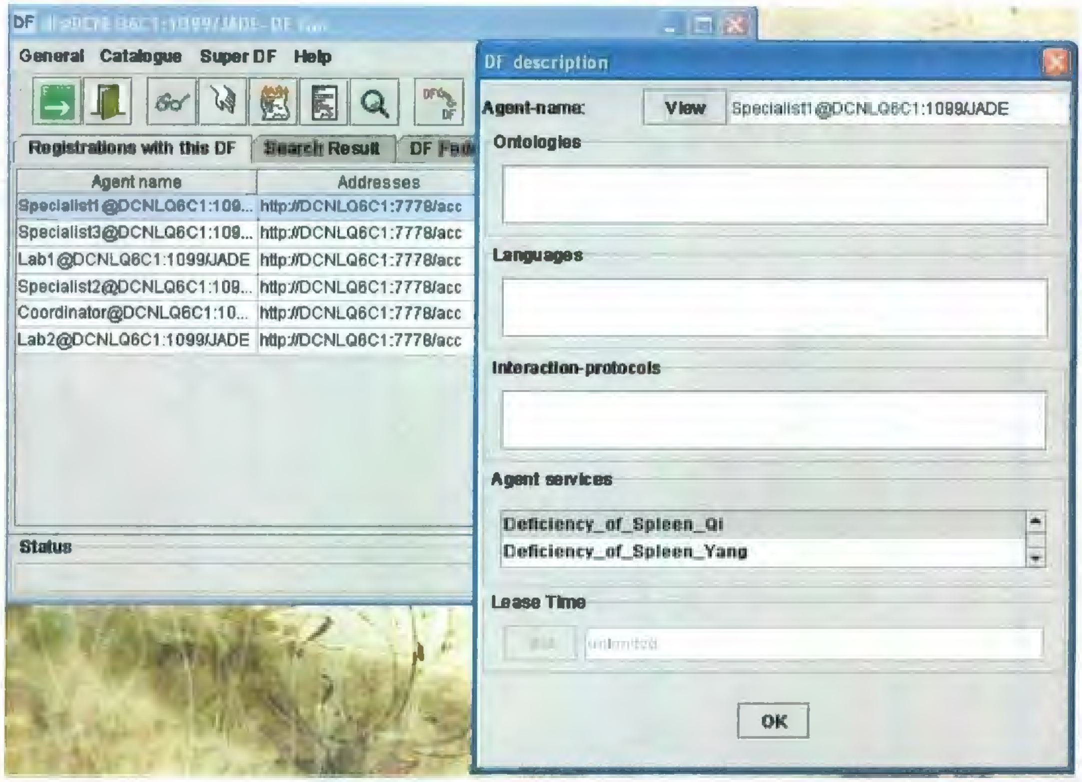


Figure 30 A DF agent with the registered services

"Knowledge Base Management" Menu

The menu for "Knowledge Base Management" is shown in

Figure 31:

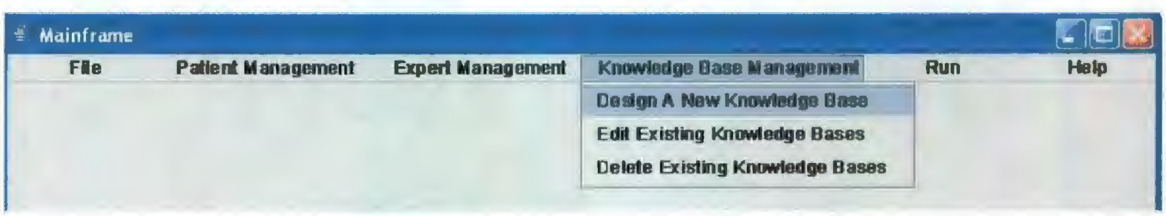


Figure 31 "Knowledge Base Management" Menu

If the user clicks on the menu item "Design a New Knowledge Base", a dialog will appear as shown in Figure 32 on the next page.

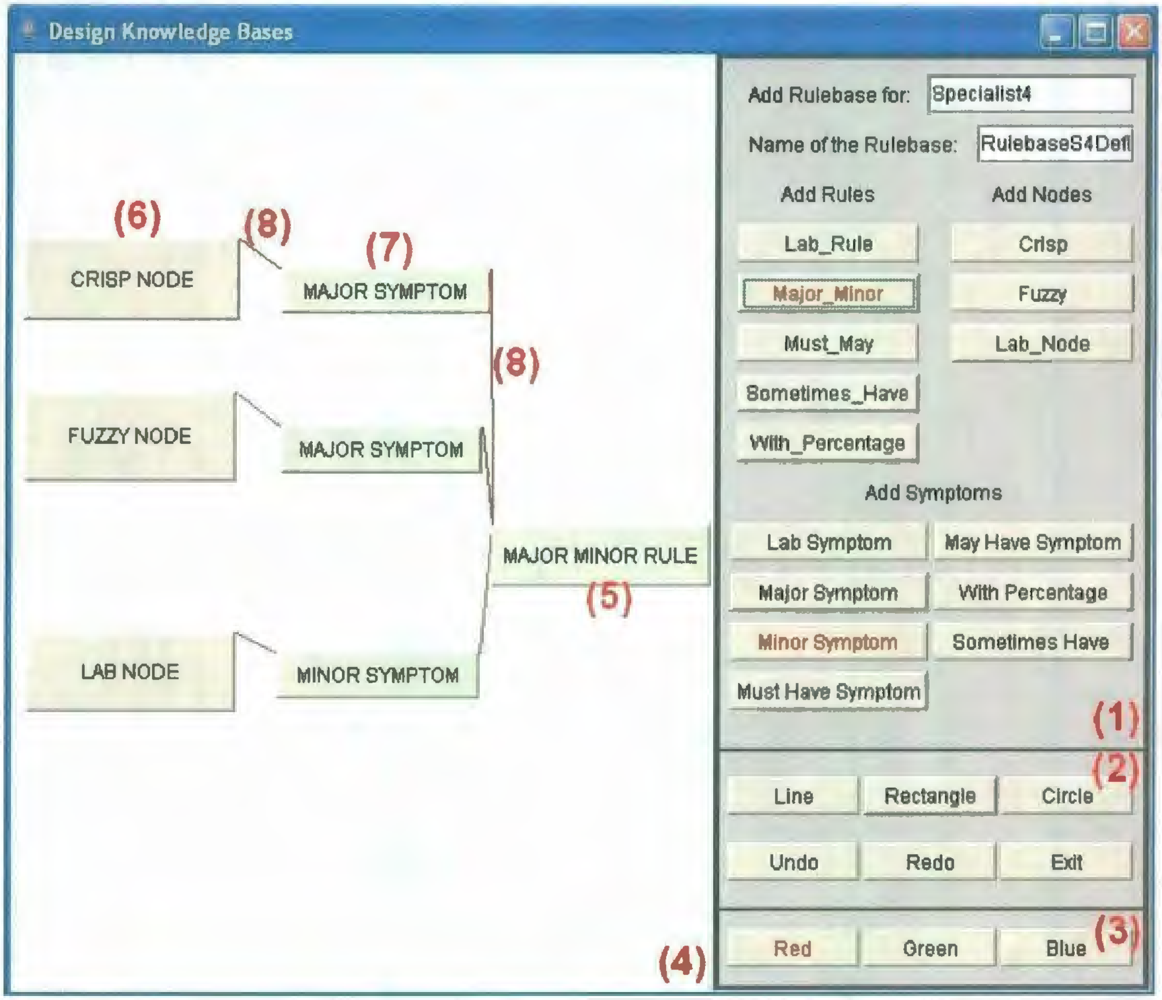


Figure 32 Dialog for the designing of a new rule base

The picture above shows the dialog provided by MADHS for the designing of a new rule base. The components of the dialog are introduced as follows:

(1) Component Panel

The Component Panel holds the buttons that are used to create the structure of a new rule base.

The two TextFields on the panel allow the user to input the name of the new rule base and which Specialist's package he or she wants to put the new rule base in.

The user can choose which component he or she wants to add to the paint board on the left side of the dialog by clicking a corresponding button. For example, if the user wants to add a Major_Minor_Rule to the rulebase, he or she can click the button "Major Minor Rule" in the column "Add Rules". After that click, the user is able to draw a button "Major Minor Rule" on the paint board using the mouse, as shown by (5) below.

(2) Shape Panel

The user can choose to draw other shapes, such as lines, rectangles and circles on the paint board by clicking the shape buttons on this panel.

The "Undo" and "Redo" buttons allow the user to undo and redo his/her drawing actions.

When the button "Exit" is pressed, the system will ask the user whether or not to save the current design of the rule base and the corresponding java code. How to generate the java code for the components of the rule base automatically will be explained in detail later.

(3) Color Panel

Three color buttons are provided on the Color Panel of the Dialog. When the user clicks on one of the three buttons, the current color setting of the paint board will change to the corresponding color.

(4) Paint Board

As mentioned previously in (1), after the current paint command is set, the user can draw the chosen component on this white paint board. He or she can decide the size of the component by dragging the mouse.

(5) Rule Buttons

A small rule base is illustrated in Figure 32 to show the designing process. Rule buttons are usually the first to be drawn. In Figure 32, "MAJOR MINOR RULE" is a typical rule button. By right clicking on this rule button, the user is able to set the main properties of this Major_Minor_Rule, such as the name and the CF thresholds. After that, the system will generate the code for this rule automatically.

(6) Node Buttons

Node buttons are usually drawn after the rule buttons. In Figure 32, "CRISP NODE" is a typical node button. By right clicking on this node button, the user is able to set the properties of this Crisp Node, such as the name and the CF threshold. After that, the system will generate the code for this Crisp Node automatically.

(7) Relation Buttons

Relation buttons are usually drawn after the rule buttons and the node buttons. These buttons represent the relations between rules and nodes. For instance, in Figure 32, "MAJOR_MINOR_SYPTOM" is a relation button.

(8) Links

The links are added to the design after the relation buttons are drawn. The red lines in Figure 32 link the rule, node and relation buttons together. For example, the CRISP NODE (6) is a MAJOR SYMPTOM (7) of the MAJOR MINOR RULE (5). First, click the node button (6), then click the relation button (7). One red line will appear between (6) and (7). After that, click the relation button (7) again, and then click the rule button (5). Another red line will be drawn between (7) and (5), as shown in Figure 32. After the links are drawn, the system will generate the code for adding this Crisp Node to the Major_Minor_Rule as one of the major symptoms.

The menu item "Edit Existing Knowledge Bases" facilitates the modification of existing designs of the rule bases. After it is clicked, a dialog will appear to let the user choose one existing design from the file system, as shown in Figure 33. The chosen design will be displayed on the white Paint Board area, as shown in Figure 32.

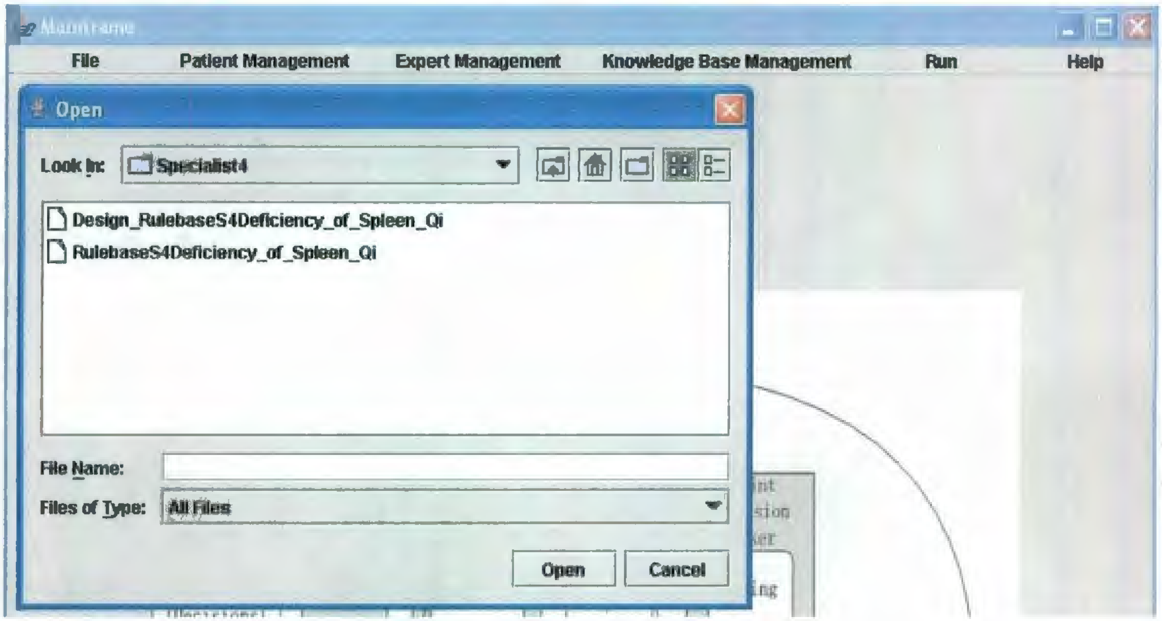


Figure 33 Edit existing knowledge bases

The menu item "Delete Existing Knowledge Bases" is attached to a similar dialog. The user can choose an existing design from the file system and send it to the recycle bin.

"Run" Menu

The "Run" Menu is shown in Figure 34:

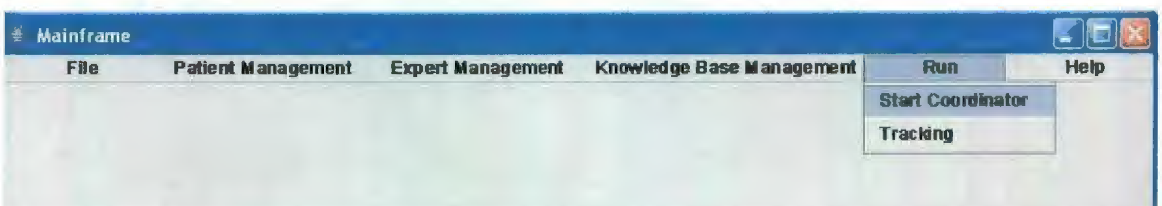


Figure 34 "Run" menu

If the user clicks on the "Start Coordinator" menu item, a "Start" request message will be sent to the Coordinator of MADHS to start a joint diagnosis of the current patient.

If the user clicks on the "Tracking" menu item, a tracking file of the current joint diagnosis will appear.

"Help" Menu

The "Help" menu is shown in Figure 35:

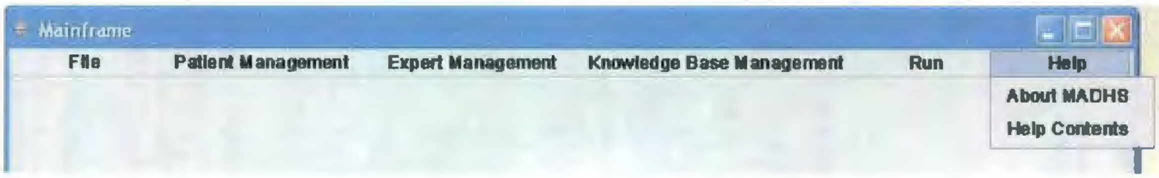


Figure 35 "Help" menu

An Example of Using MADHS GUI

A complete example of the diagnosing process using MADHS can be found in the Appendix B of this thesis.

Appendix B

Example: How to Diagnose a Patient of Stomach Disease

The complete diagnosing process of patient Li is described as an example of using MADHS GUI.

Step 1: Patient Management

If patient Li does not have a record in the MADHS file system, the user who wants to diagnose him must create a new patient record for him first. In the main menu of the GUI, choose "Patient Management" → "Add a New Patient" to create a new patient record. Fill in the name, age, gender, patient number, address and other basic information about patient Li, as shown in Figure 36 on the next page.

If the record for patient Li already exists, skip Step 1 and go to Step 2. If the user wants to edit an existing patient record, choose "File"→ "Open" to open and edit the patient record under **jade\multi\framework\patientManagement\\.**

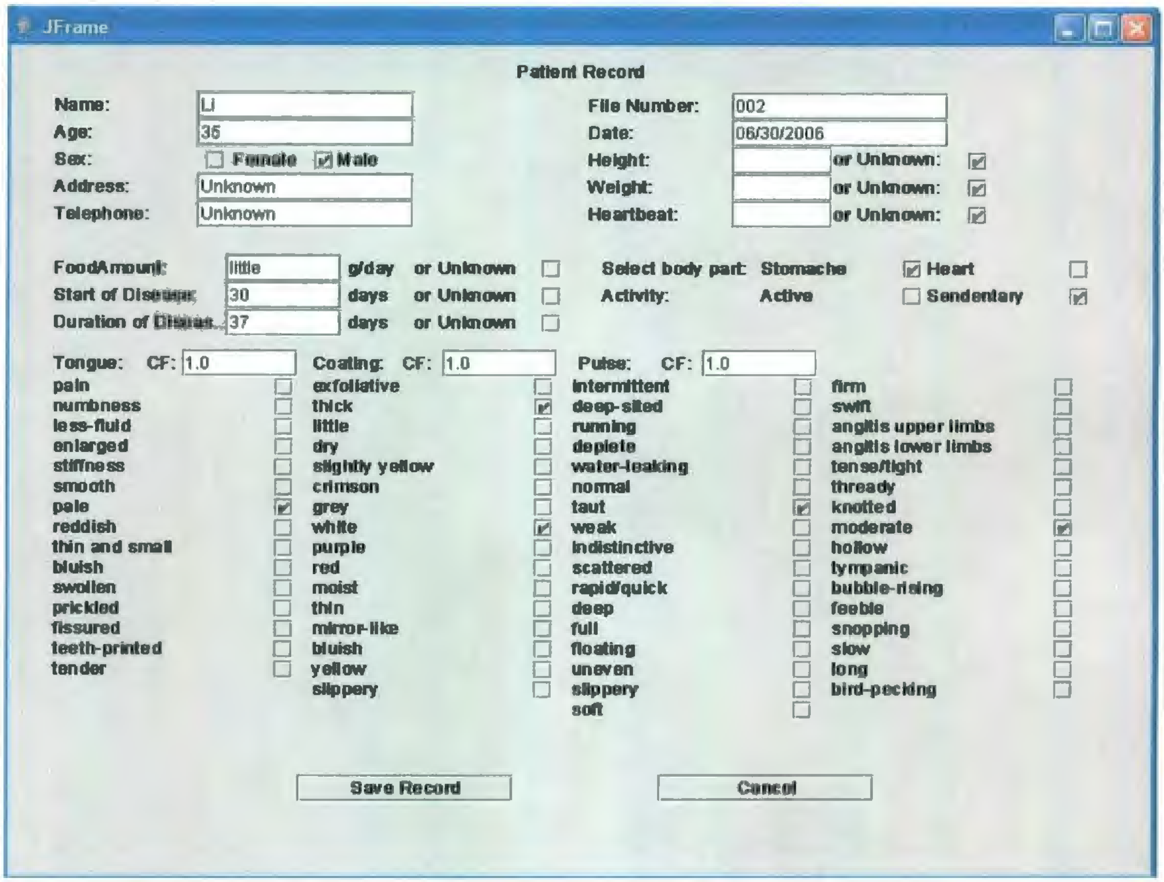


Figure 36 Step 1: Create a new patient record

Step 2: Choose the Current Patient to Diagnose

Click "Patient Management"→ "Choose Current Patient". The dialogue "Choose Current Patient" will appear. Select "002" (Patient Li) from the list of patients. Click "OK" button and close the dialogue. Step 2 is shown in Figure 37 on the next page.

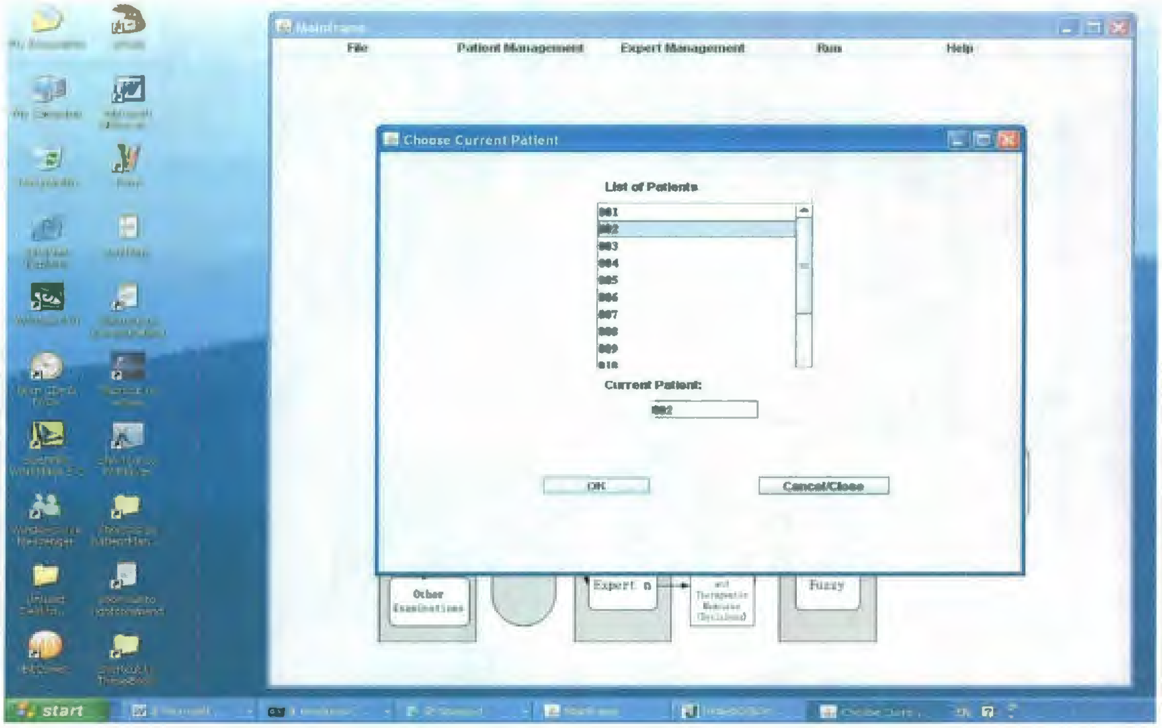


Figure 37 Step 2: Choose the current patient to diagnose: 002

Step 3: Start the Coordinator

If a user clicks on the "Run" menu item, a "Start" request will be sent to the Coordinator of the system automatically.

The user can also choose to start the joint diagnosing manually. In the GUI of JADE, right click on any agent name in the system. Use the "Send Message" function provided by JADE to send a start request to the Coordinator. In the "ACL Message" dialogue, set "Coordinator" as the sender and receiver of the message, set the "Communicative act" to be "request", set the "Content" of the message to be "Start", then send out this message. Step 3 is shown in Figure 38 on the next page.

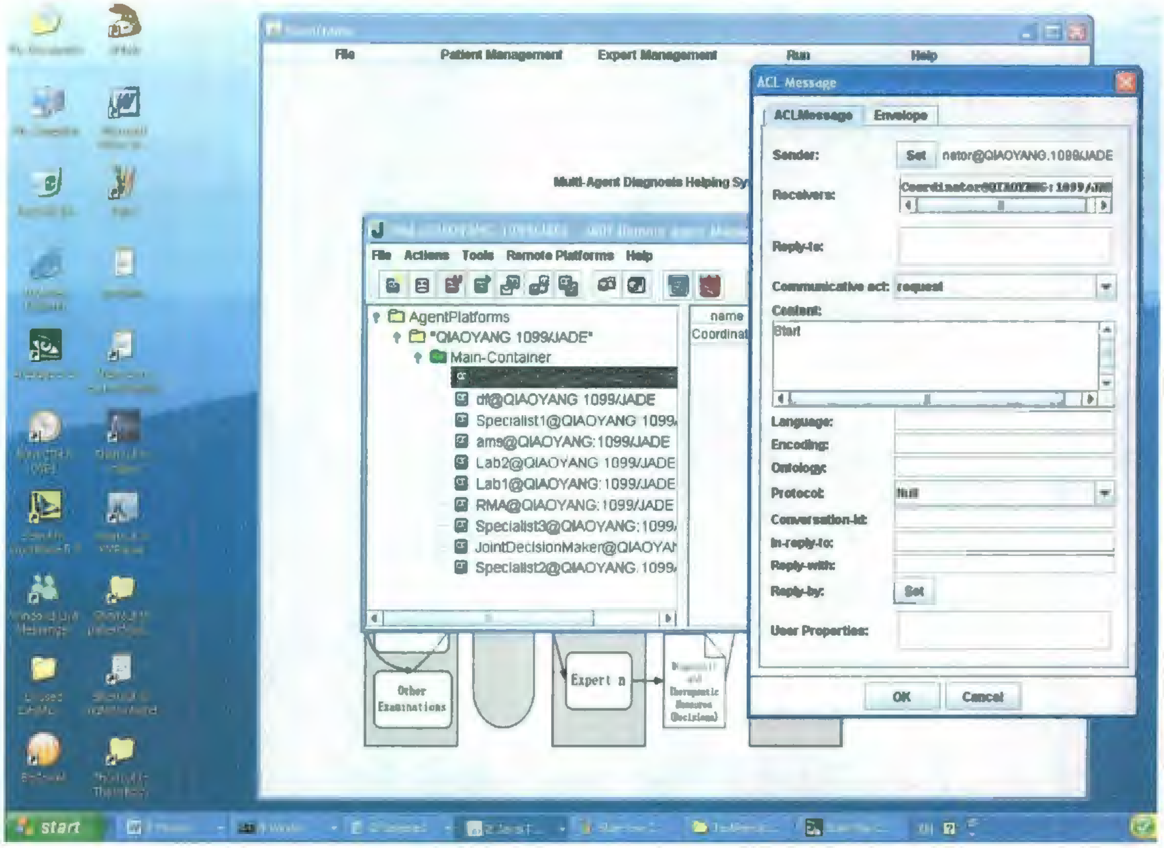


Figure 38 Step 3: Send the "Start" request to the Coordinator

Step 4: Coordinator Proposes an Initial Plan to the Joint Decision Maker

First, the Coordinator will diagnose patient Li itself. Fill the questionnaires provided by the Coordinator and Lab Agents (Lab1, Lab2). Because earlier in Figure 36 the user has specified that patient Li has stomach problems, all the questionnaires and lab forms here will be about stomach diseases, such as Deficiency of Spleen-Qi, Deficiency of Spleen-Yang, etc. This part is shown in Figure 39 and Figure 40.

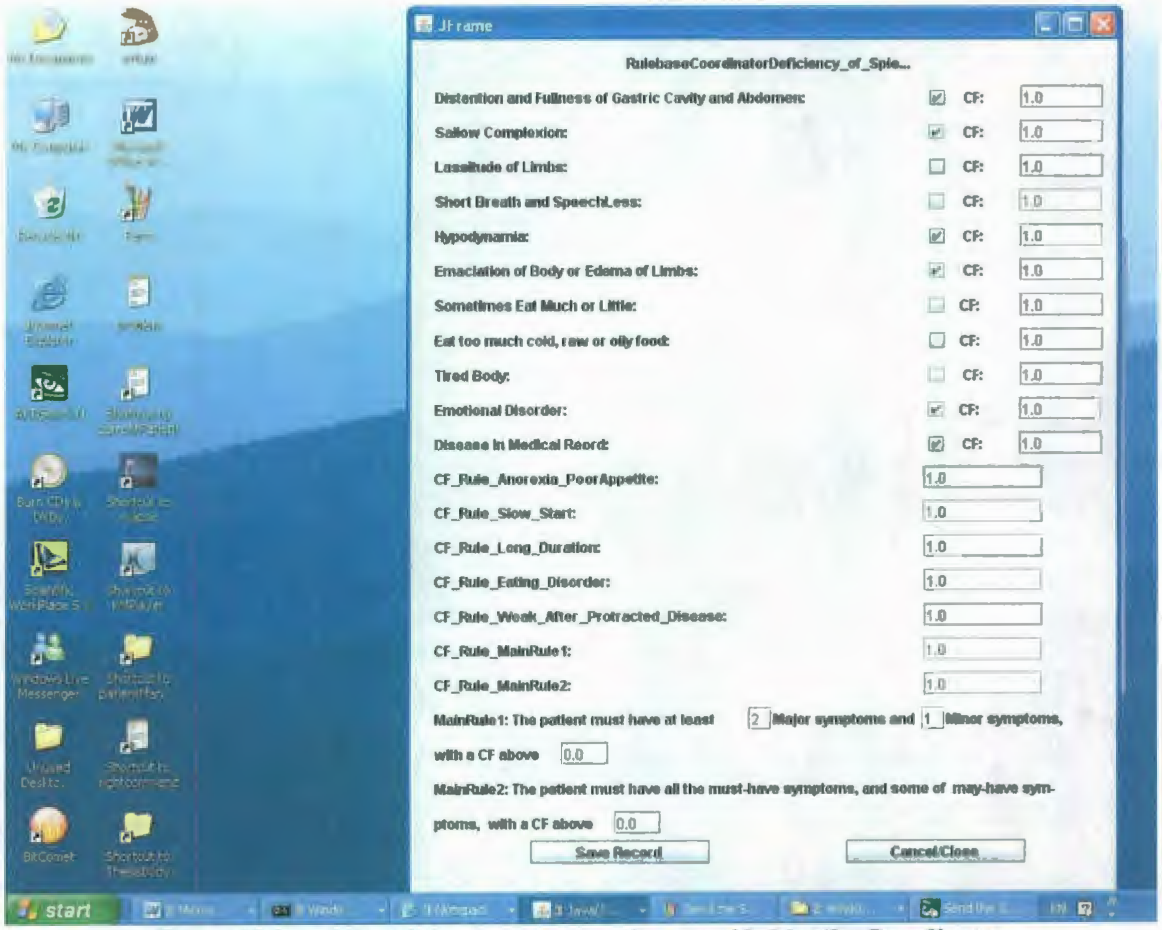


Figure 39 Step 4(1): Fill in the questionnaires provided by the Coordinator

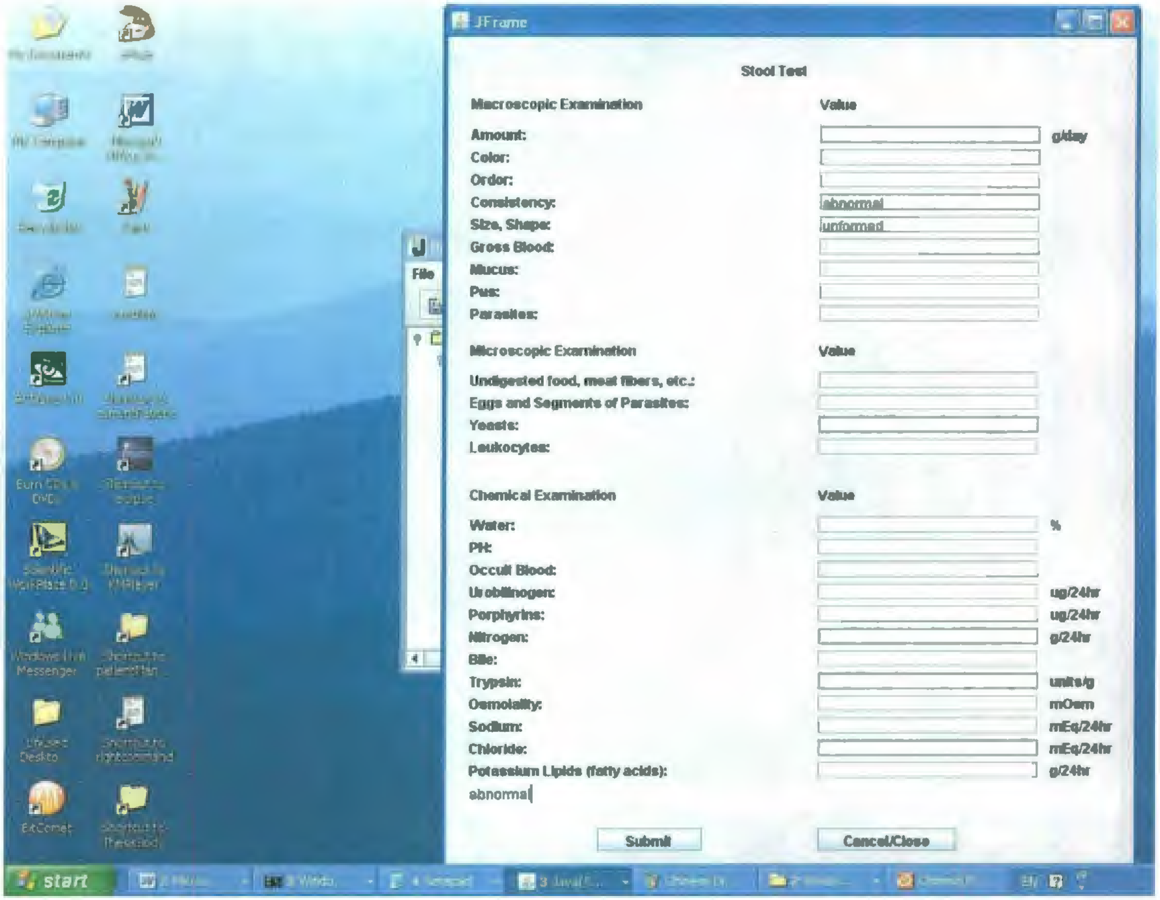


Figure 40 Step 4(2): Lab Agent (Lab1) asks information about the Stool Test

After processing all the questionnaires, the Coordinator will select the top choices from all the possible diseases, and form a team of Specialists to diagnose each of them.

Take patient Li (No. 002) for example. After processing the questionnaires handed in by the user, the Coordinator chooses two stomach diseases "Deficiency of Spleen-Qi" and "Deficiency of Spleen-Yang" from all the possible diseases, because their calculated CFs are higher than those of the other diseases. The Coordinator is also responsible for selecting a team of Specialists for each of these two diseases.

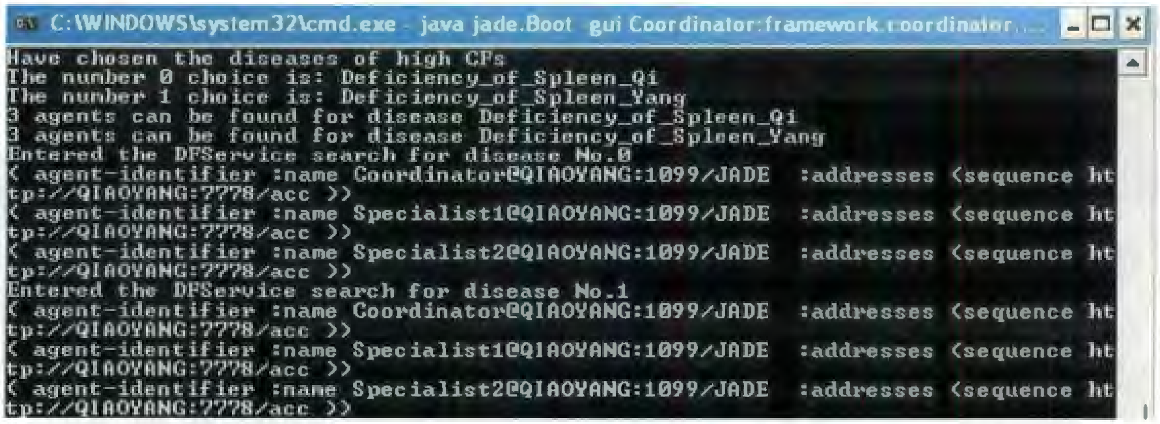


Figure 41 Step 4(3): the Initial Plan formed by the Coordinator for patient Li

Then the Coordinator sends this initial plan to the Joint Decision Maker, who will ask the user to provide the weight of each team member in the joint decision-making. This part is shown in Figure 42.

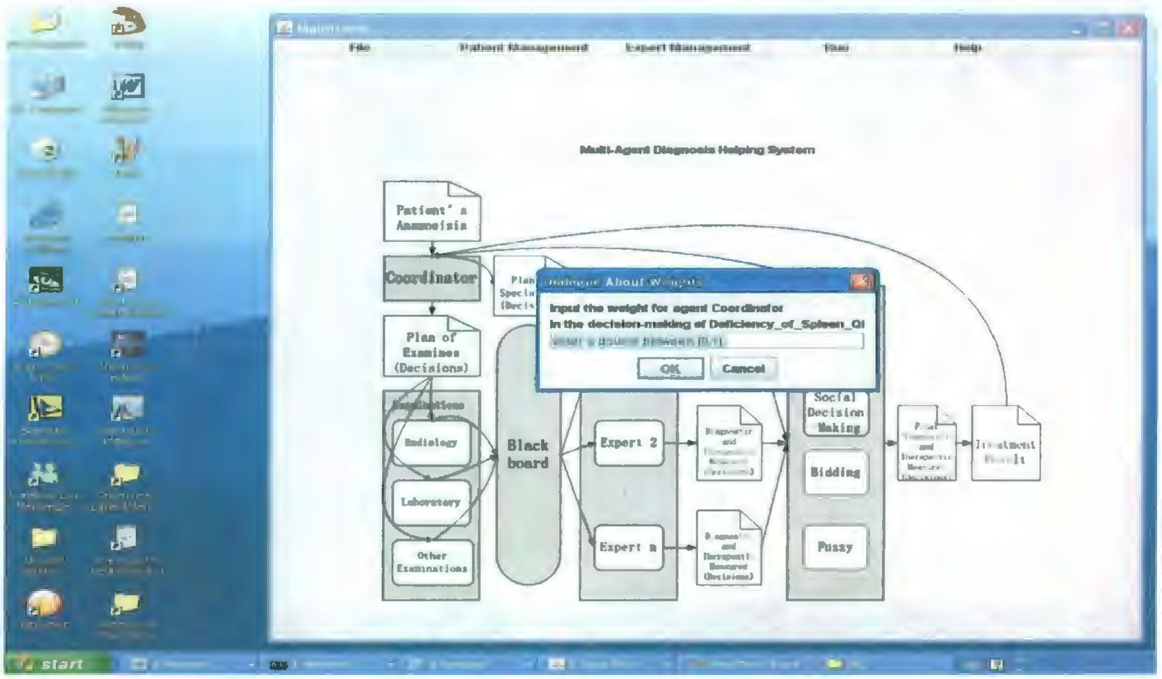


Figure 42 Step 4(4): the Joint Decision Maker asks the user to provide weights for the Specialists in the joint decision-making

The matrix of weights in the joint decision-making of this example assigned by the user is shown in Table 13:

Table 13 Weights assigned to the Specialists in the complete example

	Coordinator	*Specialist1*	*Specialist2*
Deficiency of Spleen-Qi	1.00	0.80	0.70
Deficiency of Spleen-Yang	1.00	0.90	0.80

Now the weights of Specialists in the joint decision-making are all set.

The Joint Decision Maker sends the modified initial plan (including the appointed weights) back to the Coordinator. Now the Coordinator can request the team members to do the joint decision-making together. The Specialists and Lab Agents in the team will require the user to fill in more questionnaires about the patient.

Step 5: Diagnose the Patient Using Western Medicine

In Step 4 above, the Coordinator organized a joint diagnosis of the current patient, using Chinese medical knowledge base only. After that, the Coordinator will automatically request Specialist3 (which is the only Specialist of western medical knowledge base in the system) to diagnose the patient. Specialist3 will also ask various questions about the patient and require further lab tests, as shown in Figure 43 and Figure 44:

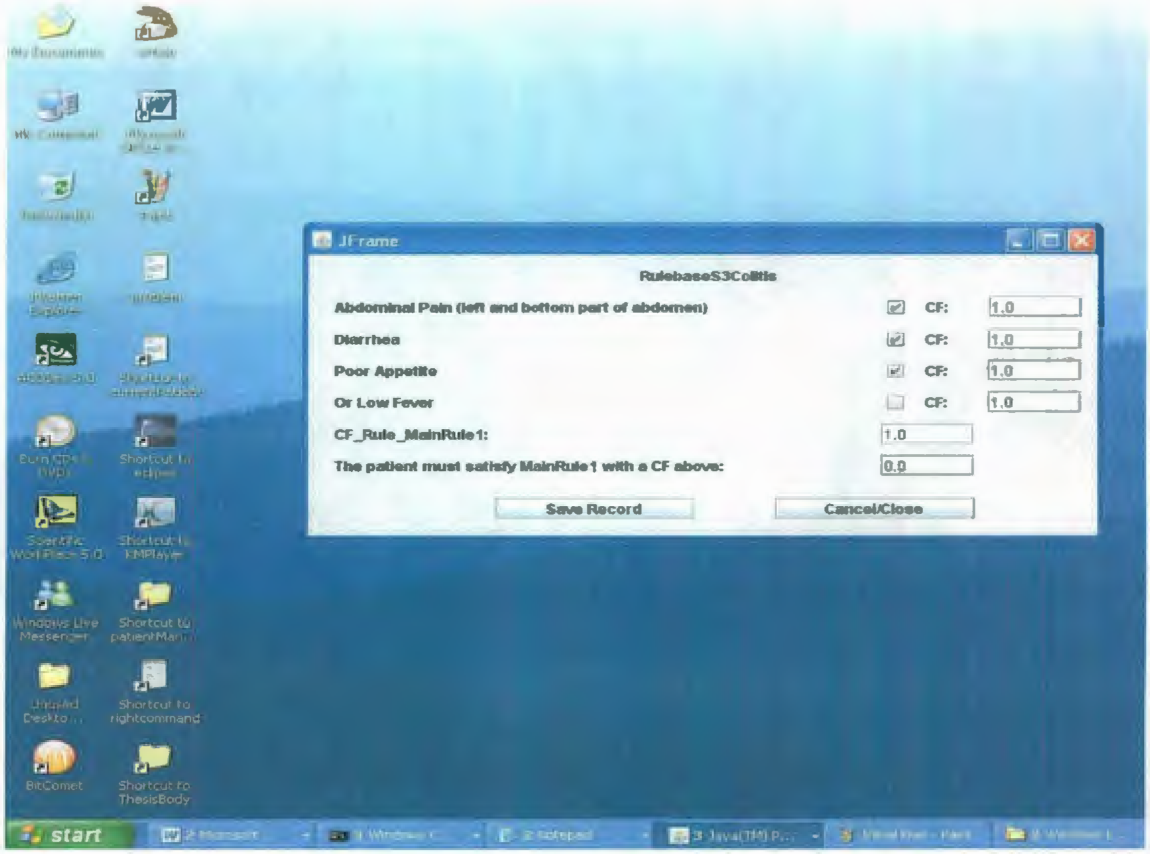


Figure 43 Step 5(1): Diagnose the current patient using western medical methods

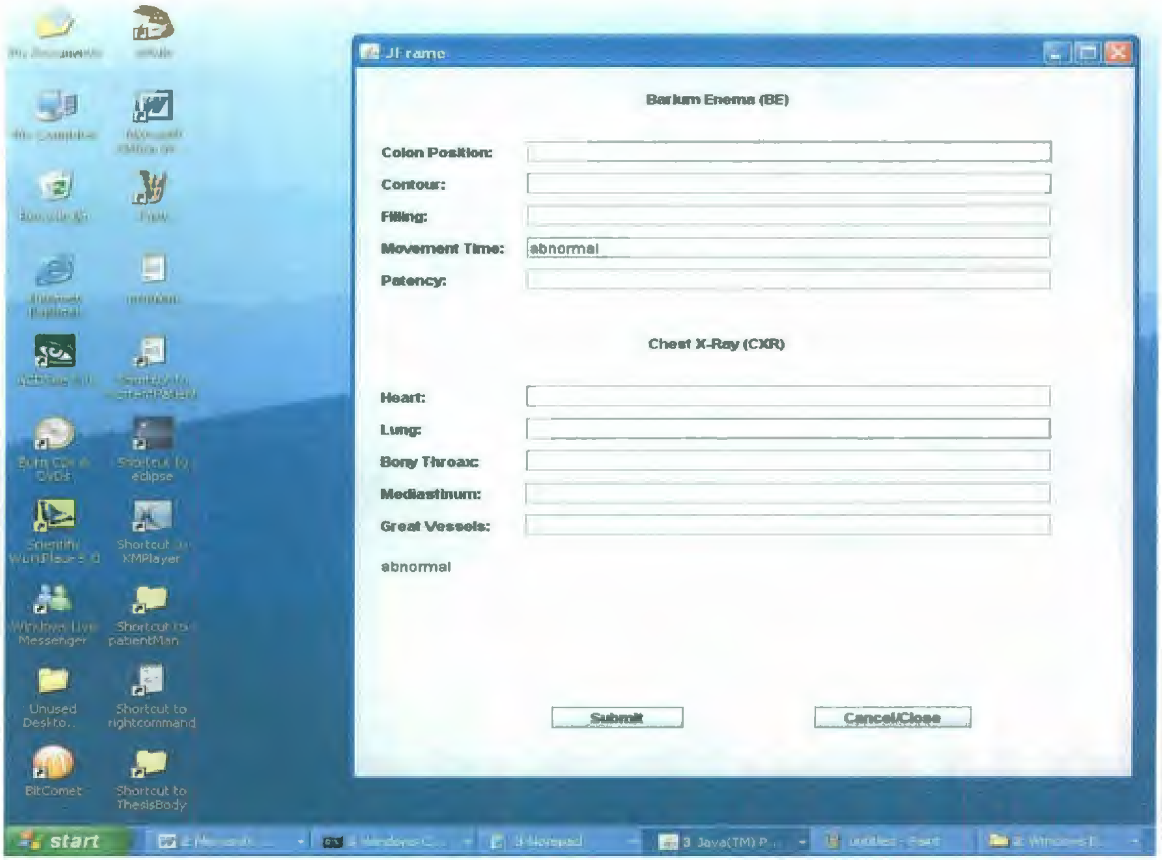


Figure 44 Step 5(2): X-Ray Test required by Specilist3 and performed by Lab2

The results of joint diagnosis of patient Li are shown in Figure 45 and Figure 46:

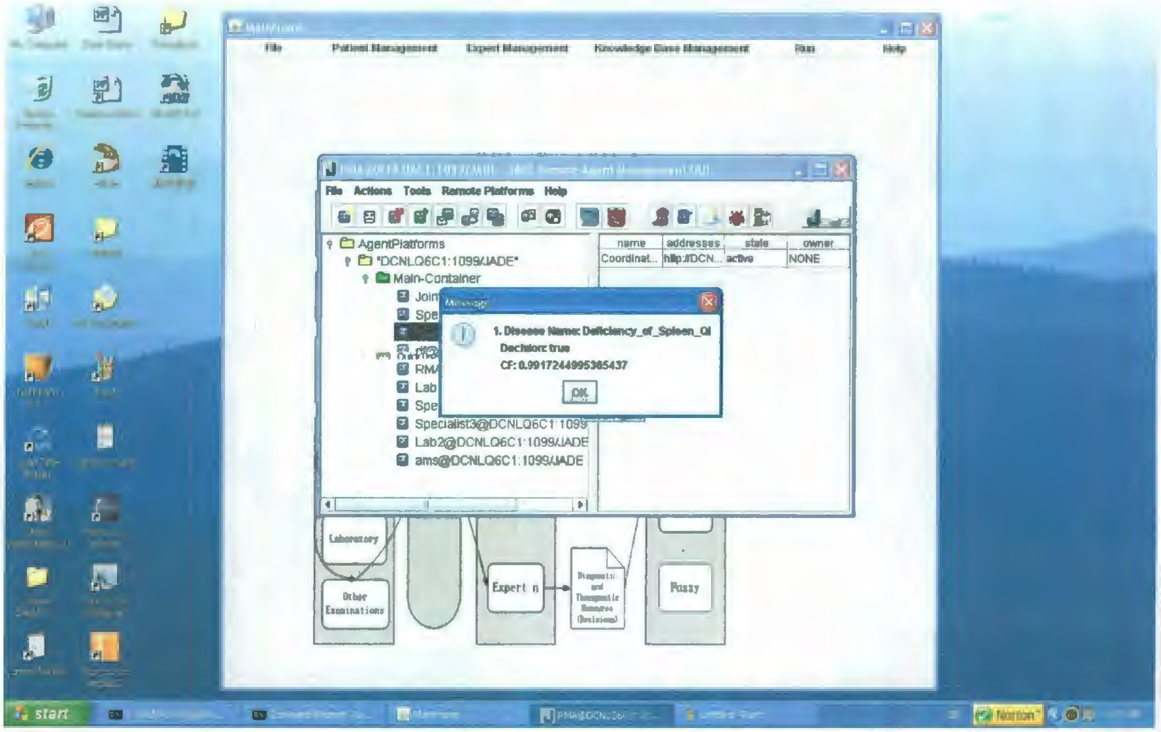


Figure 45 Step 5(3): Results of joint diagnosis for "Deficiency of Spleen-Qi"

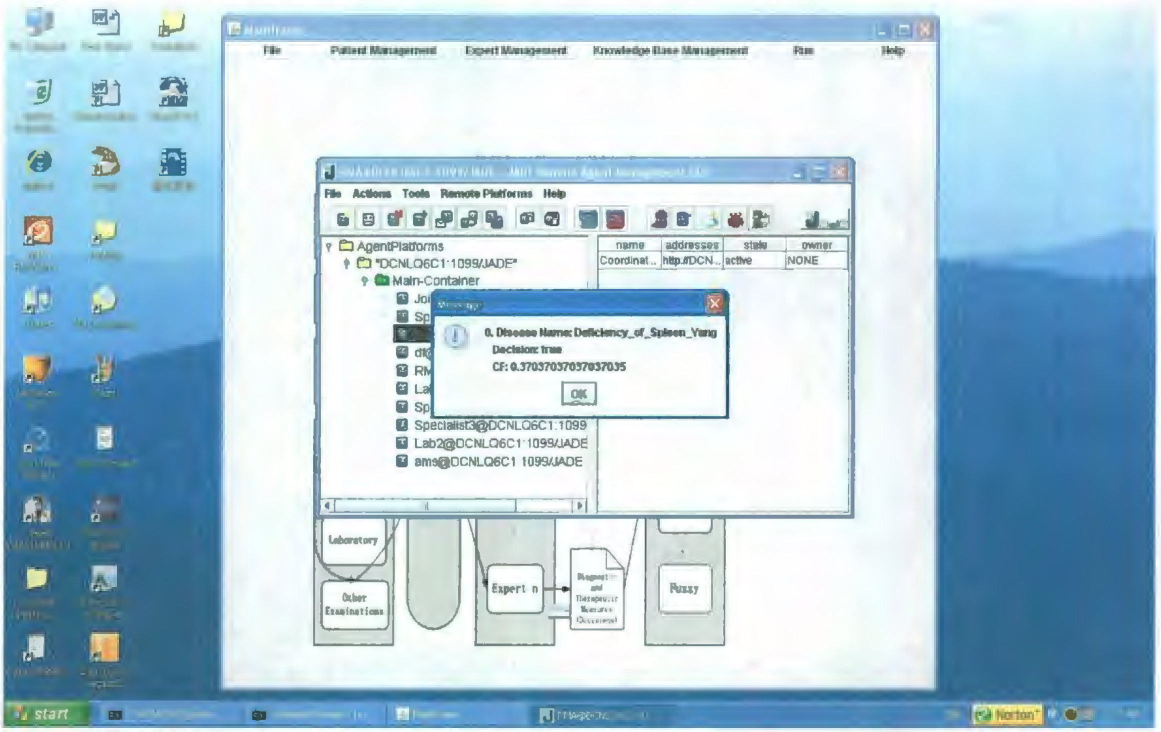


Figure 46 Step 5(4): Results of joint diagnosis for "Deficiency of Spleen-Yang"

Appendix C

List of Patient Records Used in the Tests

1. Patient 001, Wang, male, 30 years old. In the past two years, he ate less and less. He suffers short breath, hypodynamia, sallow complexion, abdominal pain (relieved when pressed). No feeling cold, no fullness of abdomen, no acid regurgitation. He has weak and deep pulse, a pale and tender tongue with white and thin tongue fur.
2. Patient 002, Li, male, 35 years old, government employee. He has been having diarrhoea for a month. Last week, he had loose stool 4 to 5 times a day. His body weight loss is 3kg. He has hypodynamia, anorexia, fullness of gastric cavity and abdomen, insomnia, retention and dribbling of urine, sallow complexion, pale tongue, white and thick tongue fur, taut and moderate pulse. Diagnosed through barium meal, he has fast enterocinesia, enteritis and irritable bowel syndrome.
3. Patient 003, Guo, female, 42 years old. She has been having abdominal pain and loose stool (2 to 3 times a day) for two months. She was diagnosed with chronic colitis before and had a history of improper treatment. She also has pale tongue, white tongue fur and taut pulse.
4. Patient 004, Zha, male, 39, Tibetan official. His first visit on March 10, 1978 was recorded as: "In the past 15 days, the situation became worse again. He caught a cold, together with oedema, hypodynamia, dizziness, heavy limbs, distension and fullness of abdomen, slightly vomiting, poor appetite, aversion to cold and cold of limbs, loose

stool, dysuria; teeth-printed, pale and enlarged tongue; white and slippery tongue fur; deep, slow and adynamic pulse. He has abnormal amount of proteinuria (++++), red blood cells and white blood cells in his urine test."

5. Patient 005, Xu, male, 50 years old, government employee. His illness was diagnosed as atrophic gastritis before. He has recurrent abdominal pain (relieved with sweet and acid food), dryness of mouth and throat, dry feces, stagnation of liver-qi; less-fluid red tongue; thready, soft and rapid pulse.
6. Patient 006, Zhang, male, 29, soldier. He has been having stomach-ache for 3 year. He was sent to hospital twice. On his visit on April 18, 1980, his symptoms were recorded as follows: "epigastralgia, fullness and pain of abdomen, hunger without appetite, retching and hiccup, dryness of mouth, dry feces, less-fluid tongue, rapid and thready pulse."
7. Patient 007, Tan, male, 36, government employee. His symptoms on September 17, 1973 were recorded as follows: "chronic stomach disease (recurrent), X-Ray showed duodenal ulcer. In the past one month, the patient had epigastralgia, pain and distension in the abdomen (especially before the meals and at night), relieved with pressure, dryness of mouth, vomiting fluid, poor appetite, acratia (hypodynamia), normal stool, frequent urination, weak and slow pulse, pale tongue, white and thin tongue fur."
8. Patient 008, He, male, 41. He has been having chronic gastritis for 5 years. Besides, he has got epigastralgia in the afternoons (relieved with eating and pressure), aversion to cold, poor appetite, pale tongue, weak pulse, white and thin tongue fur.

9. Patient 009, Wang, female, 20 years old. In the past one year, she appeared to be having some mental problems. She has a crying tendency, feverish skin, retention and dribbling of urine, normal stool, pale and reddish tongue, rapid pulse.
10. Patient 010, Xu, male, 5 years old. His visit on November 15, 1975 was recorded as follows: "The children has had whitish complexion for a week. He is exhausted because he could not sleep well at night. He has chest distress and short breath, poor appetite, hypodynamia. His heartbeat rate is 60/min, pale and tender tongue, thready and adynamic pulse. His ECG proves that he has myocarditis."
11. Patient 011, Cao, female, 37 years old. Her symptoms in February, 1975 were recorded as follows: "chest distress and pain after a cold, palpitation, heartbeat rate 110/min with premature beats, diagnosed as myocarditis in western medicine."
12. Patient 012, Hong, male, 40 years old. His visit on October 20, 1975 was recorded as follows: "The patient has been having Meniere's disease for a long time. In last April, he once had a very slow pulse (50/min). Currently, he has chest distress and short breath (especially after activities), coldness of extremities, white and thin tongue fur, and slow pulse. ECG proves that he has arrhythmia and bradycardia.
13. Patient 013, Zhang, male, 40 years old, official. His visit was on October 14, 1964. He had been having palpitation for a year, suffered short breath, insomnia, amnesia, listlessness, vertigo, acratia, white face, flushed cheeks, feverish skin, emaciate physique and body, smooth and pale tongue, rapid and adynamic pulse (130/min). His disease was diagnosed as "tachycardic" in western medicine.
14. Patient 014, Yang, female, 37. Her visit on April 18, 1975 was recorded as: "The patient has rheumatic heart disease for over ten years. After an accident in last

October, her heartbeat rate became 152/min. Her X-ray showed apparent symptoms of rheumatic heart disease. Currently, she has palpitation, chest distress and short breath (especially after activities), coldness of body extremities, spontaneous perspiration, sputum in the throat, flushed cheeks, feverish skin, insomnia, pale tongue, white and thin tongue fur.

15. Patient 015, Wen, female, 44 years old. Her first visit was on October 4, 1963. She had pulmonary tuberculosis long time ago. Her symptoms include vertigo, insomnia, palpitation, distension and fullness of abdomen, shivering tongue; white and thin tongue fur; weak, thready and slow pulse.
16. Patient 016, Kong, male, clerk. His first visit on February 6, 1975 was recorded as follows: "palpitation for two years, chest distress, short breath, dry feces; pale and reddish tongue; white and thin tongue fur; intermittent, taut and knotted tongue." Her ECG in 1972 showed premature beats.

All the patient medical records listed above are elicited and translated from a traditional Chinese medical manual [4].

Appendix D

An Example of Fuzzy Rule Bases

A Jess rule base for the disease "Deficiency of Heart-Qi" in Specialist2 is demonstrated below, which is generated automatically during run time, according to an expert's design of the rule base, and a questionnaire filled out by Patient 016. The facts are listed after the rule base.

RulebaseS2Deficiency_of_Heart_Qi.clp:

```
(import nrc.fuzzy.*)
(import nrc.fuzzy.jess.*)
(import framework.decisiontree.*)
(defglobal ?*MainRule1* = (new framework.decisiontree.Must_May_Rule))
(bind ?*MainRule1* (new framework.decisiontree.Must_May_Rule))
(call ?*MainRule1* setName "mainrule1")
(call ?*MainRule1* setThresholdCF 0.0)
(call ?*MainRule1* setCF_Rule_Must_May_Rule 1.0)

(defglobal ?*MainRule2* = (new framework.decisiontree.Must_May_Rule))
(bind ?*MainRule2* (new framework.decisiontree.Must_May_Rule))
(call ?*MainRule2* setName "mainrule2")
(call ?*MainRule2* setThresholdCF 0.0)
(call ?*MainRule2* setCF_Rule_Must_May_Rule 1.0)

(defglobal ?*Subtree1* = (new framework.decisiontree.Major_Minor_Rule))
(bind ?*Subtree1* (new framework.decisiontree.Major_Minor_Rule))
(call ?*Subtree1* setName "Subtree1")
(call ?*Subtree1* setThresholdMajor 1)
(call ?*Subtree1* setThresholdMinor 2)
(call ?*Subtree1* setThresholdCF 0.0)
(call ?*Subtree1* setCF_Rule_Major_Minor_Rule 1.0)

(defglobal ?*Subtree2* = (new
framework.decisiontree.Sometimes_Have_Rule))
(bind ?*Subtree2* (new framework.decisiontree.Sometimes_Have_Rule))
(call ?*Subtree2* setName "Subtree2")
(call ?*Subtree2* setThresholdCF 0.0)
(call ?*Subtree2* setCF_Rule_Sometimes_Have_Rule 1.0)
```

```
(defglobal ?*Subtree3* = (new
framework.decisiontree.Sometimes_Have_Rule))
(bind ?*Subtree3* (new framework.decisiontree.Sometimes_Have_Rule))
(call ?*Subtree3* setName "Subtree3")
(call ?*Subtree3* setThresholdCF 0.0)
(call ?*Subtree3* setCF_Rule_Sometimes_Have_Rule 1.0)

(defrule Rule_Palpitation "Rule_Palpitation"
(declare (salience 350))
(Record (name "Palpitation") (decision ?y) (CF ?x))
=>
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "Palpitation" crlf)
(printout outRouter ?y crlf)
(printout outRouter ?x crlf)
(bind ?P (new framework.decisiontree.CrispNode))
(call ?P setName "Palpitation")
(call ?P setResultCF ?x)
(call ?P setResultDecision ?y)
(call ?*Subtree1* addMajorSymptom ?P)
(assert (Palpitation Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
)

(defrule Rule_Chest_Distress_and_Short_Breath
"Rule_Chest_Distress_and_Short_Breath"
(declare (salience 350))
(Record (name "Chest_Distress_and_Short_Breath") (decision ?y) (CF ?x))
=>
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "Chest_Distress_and_Short_Breath" crlf)
(printout outRouter ?y crlf)
(printout outRouter ?x crlf)
(bind ?CDSB (new framework.decisiontree.CrispNode))
(call ?CDSB setName "Chest_Distress_and_Short_Breath")
(call ?CDSB setResultCF ?x)
(call ?CDSB setResultDecision ?y)
(call ?*Subtree1* addMajorSymptom ?CDSB)
(assert (Chest_Distress_and_Short_Breath Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
)

(defrule Rule_Worse_After_Activities "Rule_Worse_After_Activities"
(declare (salience 350))
(Record (name "Worse_After_Activities") (decision ?y) (CF ?x))
=>
```

```
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "Worse_After_Activities" crlf)
(printout outRouter ?y crlf)
(printout outRouter ?x crlf)
(bind ?WAA (new framework.decisiontree.CrispNode))
(call ?WAA setName "Worse_After_Activities")
(call ?WAA setResultCF ?x)
(call ?WAA setResultDecision ?y)
(call ?*Subtree1* addMajorSymptom ?WAA)
(assert (Worse_After_Activities Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
)

(defrule Pale_and_Tender_Tongue "Pale_and_Tender_Tongue"
(declare (salience 300))
(Record (name "Pale_and_Tender_Tongue") (decision ?y) (CF ?x))
=>
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "Pale_and_Tender_Tongue" crlf)
(printout outRouter ?y crlf)
(printout outRouter ?x crlf)
(bind ?PTT (new framework.decisiontree.CrispNode))
(call ?PTT setName "Pale_and_Tender_Tongue")
(call ?PTT setResultCF ?x)
(call ?PTT setResultDecision ?y)
(call ?*MainRule1* addMayHaveSymptom ?PTT)
(assert (Pale_and_Tender_Tongue Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
)

(defrule White_Fur "White_Fur"
(declare (salience 300))
(Record (name "White_Fur_Tongue") (decision ?y) (CF ?x))
=>
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "White_Fur" crlf)
(printout outRouter ?y crlf)
(printout outRouter ?x crlf)
(bind ?WF (new framework.decisiontree.CrispNode))
(call ?WF setName "White_Fur")
(call ?WF setResultCF ?x)
(call ?WF setResultDecision ?y)
(call ?*MainRule1* addMayHaveSymptom ?WF)
(assert (White_Fur_Tongue Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
```

```
)

(defrule Rule_Weak_or_Adynamic_or_RKI_pulse
"Rule_Weak_or_Adynamic_or_RKI_pulse"
(declare (salience 300))
(Record (name "Weak_or_Adynamic_or_RKI_Pulse") (decision ?y) (CF ?x))
=>
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "Weak_or_Adynamic_or_RKI_Pulse" crlf)
(printout outRouter ?y crlf)
(printout outRouter ?x crlf)
(bind ?WAR (new framework.decisiontree.CrispNode))
(call ?WAR setName "Weak_or_Adynamic_or_RKI_Pulse")
(call ?WAR setResultCF ?x)
(call ?WAR setResultDecision ?y)
(call ?*MainRule1* addMustHaveSymptom ?WAR)
(assert (Weak_or_Adynamic_or_RKI_Pulse Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
)

(defrule Rule_Vitality_Exhausted "Rule_Vitality_Exhausted"
(declare (salience 350))
(Record (name "Vitality_Exhausted") (decision ?y) (CF ?x))
=>
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "Vitality_Exhausted" crlf)
(printout outRouter ?y crlf)
(printout outRouter ?x crlf)
(bind ?VE (new framework.decisiontree.CrispNode))
(call ?VE setName "Vitality_Exhausted")
(call ?VE setResultCF ?x)
(call ?VE setResultDecision ?y)
(call ?*Subtree1* addMinorSymptom ?VE)
(assert (Vitality_Exhausted Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
)

(defrule Rule_Speechless "Rule_Speechless"
(declare (salience 350))
(Record (name "Speechless") (decision ?y) (CF ?x))
=>
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "Speechless" crlf)
(printout outRouter ?y crlf)
(printout outRouter ?x crlf)
(bind ?SBSL (new framework.decisiontree.CrispNode))
```

```
(call ?SBSL setName "Speechless")
(call ?SBSL setResultCF ?x)
(call ?SBSL setResultDecision ?y)
(call ?*Subtree1* addMinorSymptom ?SBSL)
(assert (Speechless Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency of_Heart_Qi.clp)
)

(defrule Rule_Whitish_Complexion "Rule_Whitish_Complexion"
(declare (salience 350))
(Record (name "Whitish_Complexion") (decision ?y) (CF ?x))
=>
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "Whitish_Complexion" crlf)
(printout outRouter ?y crlf)
(printout outRouter ?x crlf)
(bind ?WC (new framework.decisiontree.CrispNode))
(call ?WC setName "Whitish_Complexion")
(call ?WC setResultCF ?x)
(call ?WC setResultDecision ?y)
(call ?*Subtree1* addMinorSymptom ?WC)
(assert (Whitish_Complexion Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
)

(defrule Rule_Spontaneous_Perspiration "Rule_Spontaneous_Perspiration"
(declare (salience 350))
(Record (name "Spontaneous_Perspiration") (decision ?y) (CF ?x))
=>
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "Spontaneous_Perspiration" crlf)
(printout outRouter ?y crlf)
(printout outRouter ?x crlf)
(bind ?SP (new framework.decisiontree.CrispNode))
(call ?SP setName "Spontaneous_Perspiration")
(call ?SP setResultCF ?x)
(call ?SP setResultDecision ?y)
(call ?*Subtree1* addMinorSymptom ?SP)
(assert (Spontaneous_Perspiration Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
)

(defrule Rule_Insomnia_or_Amnesia "Rule Insomnia or Amnesia"
(declare (salience 350))
(Record (name "Insomnia_or_Amnesia") (decision ?y) (CF ?x))
=>
```

```
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency of Heart_Qi.t
xt outRouter "a")
(printout outRouter "Insomnia or Amnesia" crlf)
(printout outRouter ?y crlf)
(printout outRouter ?x crlf)
(bind ?IA (new framework.decisiontree.CrispNode))
(call ?IA setName "Insomnia_or_Amnesia")
(call ?IA setResultCF ?x)
(call ?IA setResultDecision ?y)
(call ?IA setAnd Or Flag TRUE)
(call ?*Subtree2* addSometimesHaveSymptom ?IA)
(assert (Insomnia_or_Amnesia Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart Qi.clp)
)

(defrule Rule_Somnolence "Rule_Somnolence"
(declare (salience 350))
(Record (name "Somnolence") (decision ?y) (CF ?x))
=>
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of Heart Qi.t
xt outRouter "a")
(printout outRouter "Somnolence" crlf)
(printout outRouter ?y crlf)
(printout outRouter ?x crlf)
(bind ?S (new framework.decisiontree.CrispNode))
(call ?S setName "Somnolence")
(call ?S setResultCF ?x)
(call ?S setResultDecision ?y)
(call ?S setAnd_Or_Flag TRUE)
(call ?*Subtree2* addSometimesHaveSymptom ?S)
(assert (Somnolence Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
)

(defrule Rule_Chest_Pain "Rule_Chest_Pain"
(declare (salience 350))
(Record (name "Chest_Pain") (decision ?y) (CF ?x))
=>
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency of Heart_Qi.t
xt outRouter "a")
(printout outRouter "Chest_Pain" crlf)
(printout outRouter ?y crlf)
(printout outRouter ?x crlf)
(bind ?CP (new framework.decisiontree.CrispNode))
(call ?CP setName "Chest_Pain")
(call ?CP setResultCF ?x)
(call ?CP setResultDecision ?y)
(call ?CP setAnd Or Flag TRUE)
(call ?*Subtree1* addMinorSymptom ?CP)
```

```
(assert (Chest_Pain Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
)

(defrule Rule_Sorrowfulness_Crying_Tendency_Absentminded
"Rule_Sorrowfulness_Crying_Tendency_Absentminded"
(declare (salience 350))
(Record (name "Sorrowfulness_Crying_Tendency_Absentminded") (decision ?y)
(CF ?x))
=>
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "Sorrowfulness_Crying_Tendency_Absentminded" crlf)
(printout outRouter ?y crlf)
(printout outRouter ?x crlf)
(bind ?SCTA (new framework.decisiontree.CrispNode))
(call ?SCTA setName "Sorrowfulness_Crying_Tendency_Absentminded")
(call ?SCTA setResultCF ?x)
(call ?SCTA setResultDecision ?y)
(call ?SCTA setAnd_Or_Flag TRUE)
(call ?*Subtree1* addMinorSymptom ?SCTA)
(assert (Sorrowfulness_Crying_Tendency_Absentminded Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
)

(defrule Rule_Convulsion "Rule_Convulsion"
(declare (salience 350))
(Record (name "Convulsion") (decision ?y) (CF ?x))
=>
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "Convulsion" crlf)
(printout outRouter ?y crlf)
(printout outRouter ?x crlf)
(bind ?C (new framework.decisiontree.CrispNode))
(call ?C setName "Convulsion")
(call ?C setResultCF ?x)
(call ?C setResultDecision ?y)
(call ?C setAnd_Or_Flag TRUE)
(call ?*Subtree1* addMinorSymptom ?C)
(assert (Convulsion Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
)

(defglobal ?*start* = (new nrc.fuzzy.FuzzyVariable "start" 0.0 31.0
"day"))
(load-package nrc.fuzzy.jess.FuzzyFunctions)
(?*start* addTerm "fast" (new nrc.fuzzy.ZFuzzySet 0.0 3.0 ))
(?*start* addTerm "normal" (new nrc.fuzzy.PIFuzzySet 6.5 3.5))
```

```
(?*start* addTerm "slow" (new nrc.fuzzy.SFuzzySet 10.0 31.0))

(assert (patient_startingPeriod (name "Kong") (startingPeriod (new
nrc.fuzzy.FuzzyValue ?*start* "slow"))))
(printout t "the disease_duration fact has been asserted to the Rete!"
crlf)
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)

(load-package nrc.fuzzy.jess.FuzzyFunctions)
(call nrc.fuzzy.FuzzyValue setMatchThreshold 0.0)

(defrule Rule_Slow_Start_TRUE "Rule_Slow_Start"
(declare (salience 300))
(patient_startingPeriod (name ?n) (startingPeriod ?s&:(fuzzy-match ?s
(new nrc.fuzzy.FuzzyValue ?*start* (new nrc.fuzzy.SFuzzySet 10.0
31.0)))))
=>
(printout t ?n " is Slow_Start with degree (similarity)" (* 1.0 (fuzzy-
rule-similarity)) crlf)
(assert (Record (name "Slow_Start")(decision TRUE)(CF (* 1.0 (fuzzy-
rule-similarity)))))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "Slow_Start" crlf)
(printout outRouter "TRUE" crlf)
(printout outRouter (* 1.0 (fuzzy-rule-similarity)) crlf)
(bind ?ss (new framework.decisiontree.FuzzyNode))
(call ?ss setName "CF_Rule_Slow_Start")
(call ?ss setResultCF (* 1.0 (fuzzy-rule-similarity)))
(call ?ss setResultDecision TRUE)
(call ?*MainRule2* addMayHaveSymptom ?ss)
(assert (Slow_Start Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
)

(defrule Rule_Slow_Start_FALSE "Rule_Slow_Start"
(declare (salience 300))
(not (patient_startingPeriod (name ?n) (startingPeriod ?s&:(fuzzy-
match ?s (new nrc.fuzzy.FuzzyValue ?*start* (new nrc.fuzzy.SFuzzySet
10.0 31.0))))))
=>
(printout t " This patient has not got Slow_Start of the disease." crlf)
(assert (Record (name "Slow_Start")(decision FALSE)(CF (- 1 (* 1.0
(fuzzy-rule-similarity))))))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
```

```
(printout outRouter "Slow_Start" crlf)
(printout outRouter "FALSE" crlf)
(printout outRouter (- 1 (* 1.0 (fuzzy-rule-similarity))) crlf)
(bind ?ss (new framework.decisiontree.FuzzyNode))
(call ?ss setName "CF_Rule_Slow_Start")
(call ?ss setResultCF (- 1 (* 1.0 (fuzzy-rule-similarity))))
(call ?ss setResultDecision FALSE)
(call ?*MainRule2* addMayHaveSymptom ?ss)
(assert (Slow_Start Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
)

(defglobal ?*duration* = (new nrc.fuzzy.FuzzyVariable "duration" 0.0
18250.0 "day"))
(load-package nrc.fuzzy.jess.FuzzyFunctions)
(?*duration* addTerm "short" (new nrc.fuzzy.ZFuzzySet 0.0 7.0 ))
(?*duration* addTerm "normal" (new nrc.fuzzy.PIFuzzySet 14.0 7.0))
(?*duration* addTerm "long" (new nrc.fuzzy.SFuzzySet 21.0 18250.0))

(assert (patient_diseaseDuration (name "Kong") (diseaseDuration (new
nrc.fuzzy.FuzzyValue ?*duration* (new nrc.fuzzy.PIFuzzySet 1095.0
1.0)))))
(printout t "the disease_duration fact has been asserted to the Rete!"
crlf)
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)

(load-package nrc.fuzzy.jess.FuzzyFunctions)
(call nrc.fuzzy.FuzzyValue setMatchThreshold 0.0)

(defrule Rule_Long_Durtion_TRUE "Rule_Long_Durtion"
(declare (salience 300))
(patient_diseaseDuration (name ?n) (diseaseDuration ?d&:(fuzzy-match ?d
(new nrc.fuzzy.FuzzyValue ?*duration* "above normal"))))
=>
(printout t ?n " is Long_Duration with degree (similarity)" (* 1.0
(fuzzy-rule-similarity)) crlf)
(assert (Record (name "Long_Duration")(decision TRUE)(CF (* 1.0 (fuzzy-
rule-similarity)))))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "Long_Duration" crlf)
(printout outRouter "TRUE" crlf)
(printout outRouter (* 1.0 (fuzzy-rule-similarity)) crlf)
(bind ?ld (new framework.decisiontree.FuzzyNode))
(call ?ld setName "CF_Rule_Long_Duration")
(call ?ld setResultCF (* 1.0 (fuzzy-rule-similarity)))
(call ?ld setResultDecision TRUE)
(call ?*MainRule2* addMayHaveSymptom ?ld)
(assert (Long_Duration Done))
```

```
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
)

(defrule Rule_Long_Durtion_FALSE "Rule_Long_Durtion"
(declare (salience 300))
(not (patient_diseaseDuration (name ?n) (diseaseDuration ?d&:(fuzzy-
match ?d (new nrc.fuzzy.FuzzyValue ?*duration* (new nrc.fuzzy.SFuzzySet
21.0 18250.0))))))
=>
(printout t " This patient has not got not Long_Duration of disease."
crlf)
(assert (Record (name "Long_Duration")(decision FALSE)(CF (- 1 (* 1.0
(fuzzy-rule-similarity))))))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "Long_Duration" crlf)
(printout outRouter "FALSE" crlf)
(printout outRouter (- 1 (* 1.0 (fuzzy-rule-similarity))) crlf)
(bind ?ld (new framework.decisiontree.FuzzyNode))
(call ?ld setName "CF_Rule_Long_Duration")
(call ?ld setResultCF (- 1 (* 1.0 (fuzzy-rule-similarity))))
(call ?ld setResultDecision FALSE)
(call ?*MainRule2* addMayHaveSymptom ?ld)
(assert (Long_Duration Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
)

(defrule Rule_Damage_of_Heart_Qi_Due_to_Exopathogen
"Rule_Damage_of_Heart_Qi_Due_to_Exopathogen"
(declare (salience 300))
(Record (name "Damage_of_Heart_Qi_Due_to_Exopathogen") (decision ?y)
(CF ?x))
=>
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "Damage_of_Heart_Qi_Due_to_Exopathogen" crlf)
(printout outRouter ?y crlf)
(printout outRouter ?x crlf)
(bind ?U1 (new framework.decisiontree.CrispNode))
(call ?U1 setName "Damage_of_Heart_Qi_Due_to_Exopathogen")
(call ?U1 setResultCF ?x)
(call ?U1 setResultDecision ?y)
(call ?*Subtree3* addSometimesHaveSymptom ?U1)
(assert (Damage_of_Heart_Qi_Due_to_Exopathogen Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
)
```

```
(defrule Rule_Deficiency_of_Gift "Rule_Deficiency_of_Gift"
(declare (salience 300))
(Record (name "Deficiency_of_Gift") (decision ?y) (CF ?x))
=>
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "Deficiency_of_Gift" crlf)
(printout outRouter ?y crlf)
(printout outRouter ?x crlf)
(bind ?U2 (new framework.decisiontree.CrispNode))
(call ?U2 setName "Deficiency_of_Gift")
(call ?U2 setResultCF ?x)
(call ?U2 setResultDecision ?y)
(call ?*Subtree3* addSometimesHaveSymptom ?U2)
(assert (Deficiency_of_Gift Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
)

(defrule Rule_Weak_Due_to_Senility "Rule_Weak_Due_to_Senility"
(declare (salience 300))
(Record (name "Weak_Due_to_Senility") (decision ?y) (CF ?x))
=>
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "Weak_Due_to_Senility" crlf)
(printout outRouter ?y crlf)
(printout outRouter ?x crlf)
(bind ?LTDMR (new framework.decisiontree.CrispNode))
(call ?LTDMR setName "Weak_Due_to_Senility")
(call ?LTDMR setResultCF ?x)
(call ?LTDMR setResultDecision ?y)
(call ?*Subtree3* addSometimesHaveSymptom ?LTDMR)
(assert (Weak_Due_to_Senility Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
)

(defrule Rule_Weak_Due_to_Protracted_Disease
"Rule_Weak_Due_to_Protracted_Disease"
(declare (salience 300))
(Record (name "Weak_Due_to_Protracted_Disease") (decision ?y) (CF ?x))
=>
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "Weak_Due_to_Protracted_Disease" crlf)
(printout outRouter ?y crlf)
(printout outRouter ?x crlf)
(bind ?U3 (new framework.decisiontree.CrispNode))
(call ?U3 setName "Weak_Due_to_Protracted_Disease")
(call ?U3 setResultCF ?x)
```

```
(call ?U3 setResultDecision ?y)
(call ?*Subtree3* addSometimesHaveSymptom ?U3)
(assert (Weak_Due_to_Protracted_Disease Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
)

(defrule Subtree2 "Subtree2"
(declare (salience 300))
(Insomnia_or_Amnesia Done)
(Somnolence Done)
=>
(call ?*Subtree2* setResultCF)
(call ?*Subtree1* addMinorSymptom ?*Subtree2*)
(assert (Record (name "Subtree2") (decision (call (call ?*Subtree2*
getResult) getDecision)) (CF (call (call ?*Subtree2* getResult) getCF))))
(assert (Subtree2 Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "Subtree2" crlf)
(printout outRouter (call (call ?*Subtree2* getResult) getDecision) crlf)
(printout outRouter (call (call ?*Subtree2* getResult) getCF) crlf)
)

(defrule Subtree1 "Subtree1"
(declare (salience 150))
(Palpitation Done)
(Chest_Distress_and_Short_Breath Done)
(Worse_After_Activities Done)
(Vitality_Exhausted Done)
(Speechless Done)
(Whitish_Complexion Done)
(Spontaneous_Perspiration Done)
(Subtree2 Done)
(Chest_Pain Done)
(Sorrowfulness_Crying_Tendency_Absentminded Done)
(Convulsion Done)
=>
(call ?*Subtree1* setResultCF)
(call ?*MainRule1* addMustHaveSymptom ?*Subtree1*)
(assert (Record (name "Subtree1") (decision (call (call ?*Subtree1*
getResult) getDecision)) (CF (call (call ?*Subtree1* getResult) getCF))))
(assert (Subtree1 Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "Subtree1" crlf)
(printout outRouter (call (call ?*Subtree1* getResult) getDecision) crlf)
(printout outRouter (call (call ?*Subtree1* getResult) getCF) crlf)
```

```
)

(defrule Subtree3 "Subtree3"
(declare (salience 300))
(Damage_of_Heart_Qi_Due_to_Exopathogen Done)
(Deficiency_of_Gift Done)
(Weak_Due_to_Senility Done)
(Weak_Due_to_Protracted_Disease Done)
=>
(call ?*Subtree3* setResultCF)
(call ?*MainRule2* addMayHaveSymptom ?*Subtree3*)
(assert (Record (name "Subtree3") (decision (call (call ?*Subtree3*
getResult) getDecision)) (CF (call (call ?*Subtree3* getResult) getCF))))
(assert (Subtree3 Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "Subtree3" crlf)
(printout outRouter (call (call ?*Subtree3* getResult) getDecision) crlf)
(printout outRouter (call (call ?*Subtree3* getResult) getCF) crlf)
)

(defrule MainRule1 "MainRule1"
(declare (salience 100))
(Subtree1 Done)
(Pale_and_Tender_Tongue Done)
(White_Fur_Tongue Done)
(Weak_or_Adynamic_or_RKI_Pulse Done)
=>
(call ?*MainRule1* setResultCF)
(assert (Record (name "MainRule1") (decision (call (call ?*MainRule1*
getResult) getDecision)) (CF (call (call ?*MainRule1* getResult)
getCF))))
(assert (MainRule1 Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "MainRule1" crlf)
(printout outRouter (call (call ?*MainRule1* getResult) getDecision)
crlf)
(printout outRouter (call (call ?*MainRule1* getResult) getCF) crlf)
)

(defrule MainRule2 "MainRule2"
(declare (salience 100))
(Slow_Start Done)
(Long_Duration Done)
(Subtree3 Done)
=>
(call ?*MainRule2* setResultCF)
```

```
(assert (Record (name "MainRule2") (decision (call (call ?*MainRule2*
getResult) getDecision)) (CF (call (call ?*MainRule2* getResult)
getCF))))
(assert (MainRule2 Done))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "MainRule2" crlf)
(printout outRouter (call (call ?*MainRule2* getResult) getDecision)
crlf)
(printout outRouter (call (call ?*MainRule2* getResult) getCF) crlf)
)

(defrule RulebaseS2DOHQ_TRUE "determine RulebaseS2DOHQ"
(declare (salience 0))
(Record (name "MainRule1") (decision TRUE) (CF ?x))
(Record (name "MainRule2") (decision TRUE) (CF ?y))
=>
(assert (Record (name "RulebaseS2DOHQ") (decision TRUE) (CF
(min ?x ?y))))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "Deficiency_of_Heart_Qi" crlf)
(printout outRouter "TRUE" crlf)
(printout outRouter (min ?x ?y) crlf)
)

(defrule RulebaseS2DOHQ_FALSE "determine RulebaseS2DOHQ"
(declare (salience 0))
(not (Record (name "MainRule1") (decision TRUE) (CF ?x))
(Record (name "MainRule2") (decision TRUE) (CF ?y)))
=>
(assert (Record (name "RulebaseS2DOHQ") (decision FALSE) (CF "1.0")))
(save-facts
multi/framework/specialist2/FactsS2Deficiency_of_Heart_Qi.clp)
(open
multi/framework/patientManagement/016/RulebaseS2Deficiency_of_Heart_Qi.t
xt outRouter "a")
(printout outRouter "Deficiency_of_Heart_Qi" crlf)
(printout outRouter "FALSE" crlf)
(printout outRouter "1.0" crlf)
)
(facts)
(watch facts)
(watch all)
```

FactsS2.clp (input facts generated at the beginning of the diagnosing according to the questionnaire):

```
(MAIN::initial-fact)
(MAIN::patient_name (name "Kong"))
(MAIN::patient_age (name "Kong") (age 0.0))
(MAIN::patient_sex (name "Kong") (sex "Male"))
(MAIN::patient_address (name "Kong") (address "Unknown"))
(MAIN::patient_telephone (name "Kong") (telephone "Unknown"))
(MAIN::patient_fax (name "Kong") (fax "null"))
(MAIN::patient_email (name "Kong") (email "null"))
(MAIN::patient_filenumber (name "Kong") (filenumber "016"))
(MAIN::patient_date (name "Kong") (date "07/01/2006"))
(MAIN::patient_weight (name "Kong") (weight Unknown))
(MAIN::patient_height (name "Kong") (height Unknown))
(MAIN::patient_heartbeats (name "Kong") (heartbeats 0))
(MAIN::patient_active (name "Kong") (active "Sendentary"))
(MAIN::Record (name "Palpitation") (decision TRUE) (CF 1.0))
(MAIN::Record (name "Chest_Distress_and_Short_Breath") (decision TRUE)
(CF 1.0))
(MAIN::Record (name "Worse_After_Activities") (decision FALSE) (CF 1.0))
(MAIN::Record (name "Vitality_Exhausted") (decision FALSE) (CF 1.0))
(MAIN::Record (name "Speechless") (decision FALSE) (CF 1.0))
(MAIN::Record (name "Whitish_Complexion") (decision FALSE) (CF 1.0))
(MAIN::Record (name "Spontaneous_Perspiration") (decision FALSE) (CF
1.0))
(MAIN::Record (name "Insomnia_or_Amnesia") (decision FALSE) (CF 1.0))
(MAIN::Record (name "Somnolence") (decision FALSE) (CF 1.0))
(MAIN::Record (name "Chest_Pain") (decision FALSE) (CF 1.0))
(MAIN::Record (name "Sorrowfulness_Crying_Tendency_Absentminded")
(decision FALSE) (CF 1.0))
(MAIN::Record (name "Convulsion") (decision FALSE) (CF 1.0))
(MAIN::Record (name "Pale_and_Tender_Tongue") (decision TRUE) (CF 1.0))
(MAIN::Record (name "White_Fur_Tongue") (decision TRUE) (CF 1.0))
(MAIN::Record (name "Weak_or_Adynamic_or_RKI_Pulse") (decision FALSE)
(CF 1.0))
(MAIN::Record (name "Damage_of_Heart_Qi_Due_to_Exopathogen") (decision
FALSE) (CF 1.0))
(MAIN::Record (name "Deficiency_of_Gift") (decision FALSE) (CF 1.0))
(MAIN::Record (name "Weak_Due_to_Protracted_Disease") (decision TRUE)
(CF 1.0))
(MAIN::Record (name "Weak_Due_to_Senility") (decision FALSE) (CF 1.0))
(MAIN::patient_startingPeriod (name "Kong") (startingPeriod <Java-
Object:nrc.fuzzy.FuzzyValue>))
(MAIN::patient_diseaseDuration (name "Kong") (diseaseDuration <Java-
Object:nrc.fuzzy.FuzzyValue>))
```

FactsS2Deficiency_of_Heart_Qi.clp (facts generated after the diagnosing):

```
(MAIN::Convulsion Done)
(MAIN::Sorrowfulness_Crying_Tendency_Absentminded Done)
(MAIN::Chest_Pain Done)
(MAIN::Somnolence Done)
(MAIN::Insomnia_or_Amnesia Done)
(MAIN::Spontaneous_Perspiration Done)
(MAIN::Whitish_Complexion Done)
(MAIN::Speechless Done)
(MAIN::Vitality_Exhausted Done)
(MAIN::Worse_After_Activities Done)
(MAIN::Chest_Distress_and_Short_Breath Done)
(MAIN::Palpitation Done)
(MAIN::Record (name "Subtree2") (decision FALSE) (CF 1.0))
(MAIN::Subtree2 Done)
(MAIN::Record (name "Long_Duration") (decision TRUE) (CF 1.0))
(MAIN::Long_Duration Done)
(MAIN::Record (name "Slow_Start") (decision TRUE) (CF 1.0))
(MAIN::Slow_Start Done)
(MAIN::Weak_Due_to_Senility Done)
(MAIN::Weak_Due_to_Protracted_Disease Done)
(MAIN::Deficiency_of_Gift Done)
(MAIN::Damage_of_Heart_Qi_Due_to_Exopathogen Done)
(MAIN::Record (name "Subtree3") (decision TRUE) (CF 1.0))
(MAIN::Subtree3 Done)
(MAIN::Weak_or_Adynamic_or_RKI_Pulse Done)
(MAIN::White_Fur_Tongue Done)
(MAIN::Pale_and_Tender_Tongue Done)
(MAIN::Record (name "Subtree1") (decision FALSE) (CF 1.0))
(MAIN::Subtree1 Done)
(MAIN::Record (name "MainRule1") (decision TRUE) (CF 0.3))
(MAIN::MainRule1 Done)
(MAIN::Record (name "MainRule2") (decision TRUE) (CF 1.0))
(MAIN::MainRule2 Done)
(MAIN::Record (name "RulebaseS2DOHQ") (decision TRUE) (CF 0.3))
```