An integrated approach of MAS-CommonKADS, Model-View-Controller and web application optimization strategies for web-based expert system development

S S Hasan¹ and R K Isaac²

1. Introduction

Expert system evolved as first commercial product of Artificial Intelligence and is now available in large number of areas [Durkin, 1993] such as control, design, diagnosis, instruction, interpretation, monitoring, planning, prediction, prescription, selection & simulation. [Shu-Hsiien Liao, 2005] upon survey of expert system applications and methodologies during 1995 to 2004, has classified methodologies in eleven categories viz. Rule-based systems, Knowledge-based systems, Neural networks, Fuzzy expert systems, Object-oriented methodology, Case-based reasoning, Modeling, System architecture, Intelligent agents, Ontology, and Database methodology. Expert system methodologies and applications developments are diversified due to their authors’ backgrounds, expertise, and problem domains. Trend of methodology development is also diversified due to author’s research interests and abilities in the methodology and problem domain. This indicates that the development of expert system methodologies is directed toward expertise orientation.

Use of multimedia technologies [Shalaan et al, 2004] [Ahmed Rafea et al, 1995] enhances the capabilities of an expert system. For example; clips, videos in case of pest diagnosis and treatment system better explains the symptoms and pest identification. On the other hand Web based expert system provides online accessibility with updated knowledge. Remote knowledge acquisition from domain experts and maintenance of knowledge base, inference engine, user interface, and explanation module is another advantage of web based expert system.

Although expertise incorporation in expert system makes its importance in problem domain, yet, issues such as the quality and usability of the user interface, loading time, ease of navigation, and personalization have significant importance in Web based expert systems. [Web Usability Report, 2002] by Tuscaloosa City Schools Online Technology Learning Center indicates factors to determine web site usability as; ease of learning in using web site; efficiency of use; speed of loading web sites; memorability; frequency of errors and severity; significance of errors and time for recover from errors; and satisfaction of users. General Web User Surveys [GVU’s WWW Survey Team, 2002] have found that most dissatisfying web experiences are, (a) not being able to find specific information, (b) using websites
that are confusing, and (c) websites with slow download time, respectively. Maximum complaints reported by Internet users are of slow download. Moreover, it has been found that slower web pages were significantly judged as being less interesting than their faster counterparts [Ramsay et al, 1998], and were thought to have lower quality products, as well as having compromised security. [Nielson, 1995] reported that users are most likely to lose interest in a website if the download time exceeds 10 seconds. Efficiency of web application depend on many factors as; web server technology, server processing efficiency and load, communication channel speed in between client computer and web server; efficiency of client computer itself; and also the complexity of web application. Efficiency of first three factors cannot be guaranteed. Complexity of web application may be minimized using an efficient software development methodology. Keeping in view the capabilities and features of multimedia and Internet technologies and while considering general web problems faced by users, it is imperative to develop an efficient strategy for development of web-based expert system.

The development of a Web based expert system on other hand is a multidisciplinary and difficult task and there is lack of research and of general methodology for developing Web based expert systems. Web based expert systems can be considered as Web engineering projects that can be developed by merging an expert system and a Web site with application subprojects [Duan et al, 2005]. This paper presents a framework for development of web based expert system with integrated approach of MAS-CommonKADS agent-oriented methodology, Model View Controller (MVC) architecture and Web application optimization strategies in order to achieve high usability. Study will also work out the plan of various modules, their content type and content structure for high usability and efficiency. It will also guide in software engineering of developing such system with respect to Internet Technologies.

MAS-CommonKADS is an agent-oriented methodology for development of knowledge based system. Model-View-Controller architecture is a design approach of dividing the software into Model, View and Controller component to better control the software quality with respect to processing, and interface design. Crucial part of developing successful web applications is optimizing the files that comprise them. Speed optimization affects how users perceive the efficiency and fluidity of an application, while value optimization (enhancing the readability and reusability) of the code reduces development time and provides building blocks for new versions or other web projects. Framework for developing web based expert system has been proposed and explained in section 2.

Sugarcane is a major cash crop and raw input for second largest agro-based sugar industry in India and faces many abiotic and biotic stresses which effect the crop growth and thus production and productivity. These stress cause 10-20% loss in cane production and 10-30% loss in cane productivity [Dwivedi, 2000] and may be managed with efficient strategies of crop protection. It requires having good
knowledge of various crop disorders (at various stages) along with their symptoms to identify the disorder. Expert system for disorder diagnosis can do this job in a very effective manner by providing appropriate diagnosis of disorder found in sugarcane crop. Application of above framework for developing web-based expert system in sugarcane disorder diagnosis has been developed using above framework and demonstrated in section 4 along with evaluation results.

2. Framework for development of web-based expert system

Web services are self-contained and modular business process applications based on open standards and offer a new paradigm for distributed computing. Typical agent architectures have many same features as web services, and extend web services in several ways [Minhong Wang, 2004]. Intelligent agents are used to denote a software-based system that is situated in some environment, and that is capable of autonomous action in order to meet its design objectives [Wooldridge, 2002, Wooldridge & Jennings, 1995]. To reduce the complexity of developing web based expert system, we propose MAS-CommonKADS, an agent-oriented methodology. Further, web applications work in Client-Server architecture, having server-side and client-side components for every agent. Model-View-Controller (MVC) is a design approach of dividing the software into Model, View and Controller component to better control the software quality with respect to processing, and interface design. After identification of agents, adoption of MVC paradigm will identify client and server side component in web based expert system. A crucial part of developing successful web applications is optimizing the files that comprise them. Thus strategies for optimizing identified client side and server side components of web-based expert system at implementation phase will ultimately achieve the usability of the system. Software development life cycle in our approach follows five phases of conceptualization, analysis, design, implementation and testing.

2.1 User Environment-Responsibility (UER) Technique

Conceptualization deals with elicitation of tasks to obtain a first description of the problem and to understand informal requirements [Potts et al., 1994]. It is the first step towards the identification of the functional requirements of a system. The User Environment-Responsibility (UER) technique in this phase [Iglesias et al, 1998] combines user-centered, environment-centered and responsibility-driven analysis to identify use, reaction and goal cases of an agent for multi-agent system. This technique can be used for conceptualizing a particular autonomous agent or the general requirements of a multi-agent system. In user-centered analysis potential users (called actors) of the system are identified, together with their possible tasks or functions. The result of this analysis is the set of use cases. This analysis answers the
question: What are the possible uses of the multi-agent system? The goal of environment-centered analysis is to identify the relevant objects of the environment and the possible actions and reactions of the agent. The result of this analysis is the set of reaction cases. Further, Responsibility-driven analysis explains the goals of the system.

2.2 MAS-CommonKADS

MAS-CommonKADS [Iglesias et al, 1996] methodology is an extension of CommonKADS [Schreiber et al, 1994], a knowledge engineering methodology, for multiagent systems (MAS) modeling. It defines set of seven models that together provide a model of the problem to be solved. Agent Model, that describes the characteristics of each agent; Task Model, that describes the tasks that the agents carry out; Expertise Model, that describes the knowledge needed by the agents to achieve their goals; Organization Model, that describes the structural relationships between agents (software agents and/or human agents); Coordination Model, that describes the dynamic relationships between software agents; Communication Model, that describes the dynamic relationships between human agents and their respective personal assistant software agents; and Design Model, that refines the previous models and determines the most suitable agent architecture for each agent, and the requirements of the agent network. Models are developed in analysis and design phases.

2.2.1 ANALYSIS

Analysis phase of MAS-CommonKADS determines the requirements of system starting from the problem identified in conceptualization phase. During this phase first six models are developed in a risk-driven way.

Agent Model : An agent is an executor of a task. It can be human, computer software on any other entity capable of executing a task. Agents are identified with one or more strategies as described by [Iglesias et al, 1998] viz., Analysis of the actors of the use cases defined in the conceptualization phase, Analysis of the statement of the problem, Usage of heuristics, goals of the initial task and expertise models developed, Application of the internal use cases technique, and Application of the enhanced CRC cards.

Task Model : The task model describes all the activities (so-called tasks) that should be performed in order to achieve a goal. User Environment Responsibility technique is useful for identification of tasks in the system. The task is further defined by input and output ingredients, the goal of the task, control, features, environmental constraints and required capabilities of performers.
**Expertise Model**: Expertise model (EM) describes the knowledge needed by the agents to carry out the tasks. Knowledge engineering methodology may be adopted for expertise modeling, which includes knowledge acquisition, analysis and modeling of acquired knowledge and verification of the modeled knowledge. Knowledge can be acquired using knowledge acquisition approaches such as interview, questionnaire and literature review. Knowledge Analysis & Modeling phase of knowledge engineering includes domain analysis, inference analysis and task analysis [Ahmed Rafea et al, 1994] to extract domain, inference and task knowledge. Documented knowledge acquired may be represented in the form of domain ontology and domain models in domain analysis phase. Result of inference analysis is inference knowledge, which shows all inference steps used by the system in solving the problem. Task analysis specifies tasks and algorithm of the expert system. Knowledge verification deals with quality assurance of the acquired knowledge [Adrion et al, 1982]. Knowledge is reviewed at the end of knowledge acquisition, knowledge analysis & modeling and implementation phases.

**Organization Model**: Organization model is a tool for analysis and describing the organization into which the multi-agent system is going to be introduced and the social organization of the agent society. This model shows the static or structural relationship between agents.

**Coordination and Communication Model**: Coordination model describes the conversations between agents, that is, their interactions, protocols and required capabilities. The conversations are identified, taking as input the results of the techniques used for identifying agents. Communication model details the human-software agent interactions and human factors for developing these user interfaces.

### 2.2.2 DESIGN

The design phase takes as an input the analysis models, which are transformed into specifications (the design model) ready to be implemented. It determines the architecture of both the global multiagent network and each agent. *Design model (DM)* collects the previous models and consists of three sub models [Iglesias et al, 1998] {Banff, 1996} as follows:

**Agent design or application design**: Architecture is determined for agents identified in analysis phase. Agents are subdivided in modules for user-communication (from communication model), agent communication (from coordination model), deliberation and reaction (from expertise, agent and organization models), external skills and services (from agent, expertise and task models). The agent design maps the functions defined in these modules onto the selected agent architecture. Our aim at this phase is to identify server-side and client-side components of web-based expert system under consideration. Model-
view-controller design pattern may be integrated in agent design that isolates services / business logic from user-interface resulting in an application where it is easier to modify either visual appearance of the application or underlying business rules without affecting the other.

**Agent network design or architecture design**: Infrastructure of the MAS-system is determined and consists of network, knowledge and coordination facilities.

**Platform design**: Software (multiagent development environment) and hardware needed (or available) for development of agent network and agent architecture is selected.

### 2.3 Web code optimization strategies

A crucial part of developing successful web applications is optimizing the files that comprise them. Speed optimization affects how users perceive the efficiency and fluidity of an application, while value optimization (enhancing the readability and reusability) of the code reduces development time and provides building blocks for new versions or other web projects. Many efforts have been made for increasing usability of static web pages on one hand and optimizing processing time on software execution end. Such available techniques may address the problem of web applications at implementation stage. Web applications work in client-server environment. Thus speed optimization and value optimization techniques could minimize server-side and client-side processing as well as communication between server and client.

### 2.4 Testing

Expert system testing is the procedure by which one can check that the developed system is consistent, complete, correct and satisfies the original requirements and needs of the user. This procedure evolves through a cycle of three main steps namely Verification, Validation and Evaluation [Adrion, 1982]. Verification is the demonstration of consistency, completeness and correctness of software. Validation is the determination of the correctness of the final program or software from a development project for the user needs and requirements. The goal of evaluation process is to access the quality, usability, and utility of the expert system from the point of view of human expert other than those domain experts who have participated in knowledge acquisition phase.
3. Expert system for disorder diagnosis

3.1 Conceptualization

Sugarcane is a major cash crop and raw input for second largest agro-based sugar industry in India and faces many abiotic and biotic stresses which effect the crop growth and thus production and productivity. Sugarcane crop in India faces with a number of diseases viz. Red Rot, Smut, Wilt, Grassy Shoot, Mosaic, Ratoon Stunting, Leaf Scald, Pineapple disease, Rust, Eye Spot, Yellow Spot, Brown Spot, Red Stripe. Termite, White Grub, Pyrilla, Black Bug, Scale Insect, Shoot Borer, Root Borer, Top Borer, Stalk Borer, Internode Borer, Gurdaspur Borer, Leaf Hopper, Mealy Bug, Army worm, Plassey Borer, White Fly are important insect-pest damages sugarcane crop. Further, damage is caused by regular occurrence of abiotic stresses like Drought, Water Logging, Salt Stress, Low Temperature, etc. These stress cause 10-20% loss in cane production and 10-30% loss in cane productivity [Dwivedi, 2000] and may be managed with efficient strategies of crop protection. Crop protection requires to have good knowledge of various crop disorder (at various stages) along with their symptoms to identify the disorder occurs, which in turn causes low production or yield. Expert system for disorder diagnosis can do this job in a very effective manner by providing appropriate diagnosis of disorder found in sugarcane crop.

Expert System uses human knowledge captured in a computer to solve problems. Thus there are two behavior of an expert system viz. problem solving and knowledge management. Problem solving behavior uses available knowledge to solve a given problem, while knowledge management is the updation of knowledge. Development of Expert System for Disorder Diagnosis therefore has two phases; Disorder Diagnosis and Knowledge Management.

Conceptualization reveals that Farmer, Extension Worker, Diagnostician and Domain Expert are four actors in the system. Farmer and Domain Expert are responsible for getting diagnosis of disorder and diagnostic knowledge updation respectively. Diagnostician has responsibility to diagnose the disorder. Further, Extension Worker acts as secondary actor, responsible to find disorder categories under which diagnosis has to be made. Description of actors identified along with the use cases is given in table 1.

Table 1: Actors and corresponding Use Cases for Disorder Diagnosis in Sugarcane domain

<table>
<thead>
<tr>
<th>Actor</th>
<th>Goals</th>
<th>Actor Category</th>
<th>Use Cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Farmer</td>
<td>Seek diagnosis of disorder; Ask justification of diagnosis; Ask details of disorder;</td>
<td>Primary</td>
<td>Crop Status Registration, Diagnose Disorder, Explain Diagnosis,</td>
</tr>
<tr>
<td>Extension Worker</td>
<td>Register crop status to provide relevant disorder categories, disorders under which diagnosis has to be performed</td>
<td>Secondary</td>
<td>Crop Status Registration</td>
</tr>
</tbody>
</table>
From this table it is clear that all the actors are associated with the system with certain goal. Goals of actors are represented in our study using Use Cases. Description of Use cases in Unified Modeling Language (UML) notation for disorder diagnosis is given in figure 1.

Fig 1: Use Cases for Disorder Diagnosis

Use case above shows, obtaining diagnosis of disorder by the primary actor ‘Farmer’, where the secondary actors ‘Extension Worker’ and ‘Diagnostician’ provide the knowledge for diagnosis. Extension Worker acts as secondary actor in finding the disorder category under which diagnosis can be made. Later Diagnostician predicts disorder based on the symptoms given by the ‘Farmer’ in the use cases ‘Diagnose Disorder’.

3.2 ANALYSIS

3.2.1 Agent Model

Agent modeling has generated initial instances of agent model for identifying and describing agents. Analysis of actors (at conceptualization phase), analysis of problem statement and heuristic approach has been adopted in agent identification and to specify their characteristics.

Farmer Agent: This agent works as an interface for farmers who interact with the sugarcane crop disorder diagnosis system by providing basic information about crop status and symptoms to diagnose disorders.
**Registration Agent**: Farmer needs to register his crop first to ascertain diagnosis. Registration process takes as input the properties of crop at farmer field to identify whether diagnosis for the same can be handled by our system. Once confirmed, it informs farmer about disorder categories and relevant disorders under which diagnosis has to be performed.

**Diagnostic Agent**: Once, Farmer has successfully registered his crop environment, actual diagnosis carried out by diagnostician. Diagnostician should have knowledge capabilities for diagnosis. This agent takes as input symptoms given by the Farmer, checks knowledge resources for disorder associated with given symptoms and predict disorder.

**Explanation Agent**: It explains the diagnosis using the knowledge rules available for diagnosis in knowledge resources and crop status, symptoms and confidence factor provided by the farmer for diagnosis. Based on same, it also provides certain confidence level with which diagnostician has diagnosed the disorder.

**Domain Expert Agent**: Alike Farmer agent, this agent acts as interface for domain expert to view and update knowledge.

**Knowledge Update Agent**: This agent is responsible for updation of knowledge resources on request of domain expert agent. It also provides the details of knowledge residing in knowledge resources.

**KnowledgeBase Agent**: KnowledgeBase agent provides mechanism to store and retrieve the knowledge used by all other agents.

3.2.2 **Expertise Model**

The expertise model described the reasoning capabilities of the agents to carry out their tasks and achieve their goals. We have identified two types of ontologies in sugarcane crop disorder diagnosis corresponding to knowledge used by Extension Worker and Diagnostician. Ontology for Crop Status defines the term used in domain for exploring the possibility of disorder diagnosis in sugarcane crop. Ontology for disorder diagnosis defines disorder and symptom concepts. Domain model defines the relationship of domain ontology used for solving / interpreting problems. A sample domain model used for disorder diagnosis is as follows:

(Region: State = ‘Uttar Pradesh’ & Crop: Type = ‘Plant’ & crop: Stage = ‘Maturity’)  
SUSPECT  
(Nitrogen Deficiency & Shoot Borer & Root Borer & Red Rot)
Inference knowledge in our case shows inference steps used by the system in solving a problem, while task knowledge specifies algorithm of the task.

3.2.3 **Task Model**

Task model describes the tasks carried out by the agents. Tasks should be kept minimum for optimized processing need at implementation stage. Table 2 below shows the sample textual template for a tasks ‘Determine Crop Status Relevance’ performed by Registration agent in sugarcane disorder diagnosis system.

**Table 2: Sample textual template in task model**

<table>
<thead>
<tr>
<th>Task</th>
<th>Determine Field Crop Status Relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objective</td>
<td>To find out possibility of disorder diagnosis for given Field Crop Status</td>
</tr>
<tr>
<td>Description</td>
<td>This task compares the crop status received from the ‘Farmer’ agent with available crop status (in which diagnosis can be made) in knowledge resources. Depending on availability, pass on successful message to next task, or otherwise inform Farmer about irrelevance</td>
</tr>
<tr>
<td>Input</td>
<td>(User) Field Crop Status</td>
</tr>
<tr>
<td>Output</td>
<td>An indicator of possibility of diagnosis (Diagnosis Possibility Status)</td>
</tr>
<tr>
<td>Exceptions</td>
<td>Absence of Farmer Crop Status or Crop Status in knowledge resources</td>
</tr>
<tr>
<td>Precondition</td>
<td>Availability of Field Crop Status</td>
</tr>
</tbody>
</table>

3.2.4 **Coordination and Communication Model**

Interaction among agents is of two types, one for using available knowledge and other for updating the existing knowledge. Agent-to-agent interaction in our system follows coordination model, while communication model describe user-to-agent interaction.

3.3 **Design Model**

Two types of modules are identified for agents namely, Functional and Interface modules. Task models, expertise models and heuristic approaches have been applied for the same. In most of the cases task identified in task model has been converted into modules. Further, task merger and decomposition has been done in some cases to optimize number of modules as well as modular level processing.

3.3.1 **Functional Modules**

Functional modules are associated with actual functional performance of the agent, which generate output with some processing on input provided. Tabular template (table 3) explains such modules for various agents identified earlier along with input, output and knowledge level attributes. Knowledge required to perform functions in each module were collected from Expertise model of
analysis phase. Various modules of an agent along with Knowledgebase constitute ‘Model’ component of MVC architecture which get triggers by some user input.

‘Control’ component of MVC has been mapped in the template using the Input column. Input in our case is ascertained from communication and coordination model of analysis phase. Input to agent could be from user or earlier agent / modules and work as trigger to the agent / module. ‘Output Produced’ column shows the processed result of each module.

Table 3: Functional modules in Sugarcane Disorder Diagnosis

<table>
<thead>
<tr>
<th>Agent</th>
<th>Functional Module (Task &amp; Expertise Model)</th>
<th>Input (Communication &amp; Coordination Model)</th>
<th>Knowledge Required (Expertise Model)</th>
<th>Output Produced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Registration</td>
<td>Determine Field Crop Status Relevance</td>
<td>Field Crop Status (Region.State, Crop.Type, Crop.Maturity, Crop.Stage)</td>
<td>Crop Status – Diagnosis Possibility Status</td>
<td>Diagnosis Possibility Status</td>
</tr>
<tr>
<td></td>
<td>Determine Diagnosis Relevance</td>
<td>Diagnosis Possibility Status</td>
<td>Crop Status – Disorder rules</td>
<td>Possible Disorders, Possible Disorder Categories, Diagnosis Relevance Status</td>
</tr>
<tr>
<td>Diagnostic</td>
<td>Formulate Symptom Category</td>
<td>Disorder Category</td>
<td>Disorder Category – Disorder rules</td>
<td>Symptom Categories, Relevant Disorders</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Possible Disorders</td>
<td>Disorders – Symptoms rules</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Symptoms – Symptom Category rules</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Generate Symptoms</td>
<td>Symptom Category</td>
<td>Disorders, Crop Status, Symptom Category -- Symptoms rules</td>
<td>Symptoms</td>
</tr>
<tr>
<td></td>
<td>Predict Disorder</td>
<td>Selected Symptoms</td>
<td>Symptom Category, Symptom – Disorder rule</td>
<td>Predicted Disorder</td>
</tr>
<tr>
<td></td>
<td>Explanation</td>
<td>Disorder, Explanation Category</td>
<td>Disorder, Explanation Category – Explanation rule</td>
<td>Explanation</td>
</tr>
</tbody>
</table>

3.3.2 Interface Module
Farmer and Domain Expert interact with expert system for getting diagnosis and knowledge updation respectively using ‘Farmer’ and ‘Domain Expert’ agent. Table 4 shows interface modules of ‘Farmer’ agent, which provide interface for Farmer to interact with Expert System. Since objectives of interface modules is to provide an interface for user interaction and thus correspond to ‘View’ component of MVC architecture.

Table 4: Interface module in disorder diagnosis for ‘Farmer’ agent

<table>
<thead>
<tr>
<th>Interface Module</th>
<th>Function</th>
<th>Input Required (Communication Model)</th>
<th>Output Produced (Communication Model)</th>
<th>Control Element</th>
</tr>
</thead>
<tbody>
<tr>
<td>Field Crop Status</td>
<td>Selection of crop status of Farmer’s field</td>
<td>Crop Concept Properties (Single)</td>
<td>Crop Concept Properties (Single)</td>
<td>List Box</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Crop Concept Properties Values (Many)</td>
<td>Crop Concept Properties Values (Single)</td>
<td></td>
</tr>
<tr>
<td>Disorder, Symptom, Multimedia Category</td>
<td>Selection of disorder, symptom and multimedia category under which diagnosis has to be made</td>
<td>Disorder Categories (Many), Symptom Categories (Many), Multimedia Categories (Many)</td>
<td>Disorder Categories (Many), Symptom Categories (Many), Multimedia Categories (One)</td>
<td>List Box</td>
</tr>
<tr>
<td>Symptoms</td>
<td>Selection of Symptoms found in Farmer’s field</td>
<td>Symptoms (Many)</td>
<td>Symptoms (Many)</td>
<td>Option</td>
</tr>
<tr>
<td>Predicted Disorder</td>
<td>Display predicted disorder and control for further treatment / justification</td>
<td></td>
<td>Predicted Disorder, Link to other agents (Explanation, Treatment)</td>
<td>Hyper Link</td>
</tr>
<tr>
<td>Explanation Category</td>
<td>Selection of Explanation Category for Explanation</td>
<td>Explanation Categories (Many)</td>
<td>Explanation Category (Single)</td>
<td>List Box</td>
</tr>
<tr>
<td>Explanation</td>
<td>Display Explanation of Diagnosis</td>
<td></td>
<td>Explanation</td>
<td></td>
</tr>
</tbody>
</table>

3.3.3 Knowledge Base

Optimized data storage and retrieval capabilities requires, an efficient knowledge base design. Since download speed of web pages depend on the volume of information being downloaded. Certain knowledge division policies have been adopted so that only relevant information should be downloaded at user. Adopted policies are Division of Field Crop Status under different Concept / Properties; Categorization of symptoms; Grouping of disorders; Separation of multimedia contents; Grouping of Explanation knowledge; Coding mechanism to identify properties and using codes instead of properties in user communication as far as possible; and loading knowledge base with relevant and concise data.
Entity-Relation modeling scheme has been used for knowledge representation with two type of entities ‘Data Table’ and ‘Rule Table’ as shown in figure 2.

Fig 2: Disorder Diagnosis Knowledge Base Structure

Data Tables (shown by rectangular boxes) are repository of domain ontology defined in Expertise Model and acquired during knowledge acquisition. Knowledge categories covered by these tables are crop disorders, textual & multimedia symptoms and other concepts concerned with sugarcane crop and disorders. ‘Rule Table’ (with rounded box) along with line joining two entities, maintain the relation among domain ontology defined earlier by domain models.

Three types of components are used by agents in developed system viz. Interface module, Functional module and Knowledge base. In MVC architecture, interface module is represented in one or more View component. Functionality of all agents is defined in ‘Model’ component in the form of behavior and data. Controller provides triggering events for changing the state of Model and also the View. Sample architecture of agent’s modules for Disorder Diagnosis is shown in figure 3 below.
In MVC architecture implemented in our work for web based expert system, all functional modules of agents reside in Web Server and Knowledge Base agent in Database Server (depending on volume of knowledge) for execution at server end only. Output of the functional module along with display / control element is communicated to Client End using View component. On getting user input through View component, Controller checks the same to decide the change of model state (possibly call to new agent) and also View (new user interface). All the communication between Server and Client machine in web application are made using http protocol.

Task and Expertise model defines the behavior and knowledge of these agents in Model component. Coordination and Communication model defines the Control component of MVC. Further Organization model has mapping to collection of all possible Views, Model State and Control.

### 3.4 Implementation

After efficient design of agents, coding of software has been performed at this stage. Microsoft ASP .NET technology has been adopted in developing web based expert system. Although MVC architecture was used in designing phase, in implementation we have not separated Model and Controller rather put them integrated fashion. Logic behind this is separation of controller from model also increases the software code although it increases the code reusability. View component of various modules are shown in following figures 4.
Windows 2003 operating system on Intel Xeon based server has been adopted to provide basic operating platform at server end. Internet Information Services (IIS) server software has been used to host the Web-based Expert System for Disorder Diagnosis. Web server stores all the modules of agents. For testing of expert system, we have adopted most general form of database services provided by Microsoft Access. Knowledge Base is the repository of all the expert knowledge to be used by the experts for providing solutions to problems in their domain. Relational database is best-suited design approach for an efficient management of knowledge and its retrieval. In our approach domain ontologies (concept, properties, legal values, antecedent statements and multimedia) has been stored in data tables. Domain models, (that is relationship among concept properties, antecedent statements and multimedia elements) has been indicated by relations among data tables.

Since web application work in Client-Server environment, thus efficiency could be achieved by optimizing the processing both at client and server end. In MVC this was achieved by optimizing both model-controller and view components. For high usability of web application, optimization at implementation phase has been performed on five aspects viz., server-side processing (dynamic code), database design, client-side static code, client-side downloadable contents, and structure & placement of client-side web content.
Fig 4: Screenshots of (a) Field Crop Status collection (b) Selection of various categories and (c) Selection of symptoms.
3.5 Expert System Testing

Verification of expert system for disorder diagnosis has been carried out from knowledge engineering and software engineering aspects. Knowledge acquired for disorder diagnosis was verified from domain experts. In case of conflict, recommendation of majority was considered or otherwise recommendations of most experienced one was considered. Verification of developed software was conducted for semantic and syntactic errors. Further programming logic helped us in optimization of software code for speed and value aspects.

Validation of expert system has been conducted to check that software works as per user expectation and also usability aspects of web applications. Registration and Diagnostic agents perform diagnosis of sugarcane disorder using four modules. All the modules / tasks were validated with various combination of following user input.

**Field Crop Status**
- State: Uttar Pradesh
- Season: All
- Variety: All
- Planting Time: All
- Present Stage: Maturity, All

**Categorical Choice**
- Disorder Category: Disease, Insect, Nutrition, All
- Symptom Category: Crop, Disorder, Foliage, Shoot, Root, All

**Number of Symptoms selected:** 1, 2-4, 5-8, All

Test was conducted with such input scheme in which processing time at server end will be high to show the performance of task in extreme high processing situations only. Web pages downloaded at client-end under various input scheme were found to be well below 2000 characters, as shown in figure 5 & 6, which is well below the optimum web page size limit of 34 kb [Web usability report, 2002], except few extreme cases. Such extreme cases occur when user did not select categorized input.
Further, processing time at server-end and complete download time at client-end of web pages was measured for three modules of Registration and Diagnostic agents under various input scheme defined earlier. Results shows (Fig 7, 8, 9) that complete download time of web page is almost same as processing time at web server-end and well below the 10 seconds advised by [Nielson, 1995] to achieve usability of application. Result shows that server processing time is critical for web-based expert system for working / executing server-side script.
Fig 7: Comparison of complete download time, ASP execution time and web page download time for ‘Determine Diagnosis Relevance’ module

![Graph showing comparison of complete download time, ASP execution time and web page download time for 'Determine Diagnosis Relevance' module.]

Fig 8: Comparison of ASP Execution Time and Complete Download Time for ‘Generate Symptoms’ module

![Graph showing comparison of ASP execution time and complete download time for 'Generate Symptoms' module.]
Evaluation is the process to assess quality, usability and utility of expert system from point of view of human expert other than those domain experts who have participated in knowledge acquisition phase. We have evaluated the same from 30 stakeholders dealing with extension of sugarcane production and protection technologies to farmers in Uttar Pradesh.

Stakeholders were asked to use the expert system for disorder diagnosis independently as well as by providing set of test cases. After use of the system by stakeholders, they were asked to provide evaluation results using a questionnaire. Opinion was sought from them on following points and measured in five point scale (see figure 10):

1. User screen are informative enough in understanding the queries being asked
2. Multimedia was sufficient in explaining Expert System reasoning and query
3. Readability and visibility was maintained with appropriate size of screen text
4. Layout of contents was perfectly designed to find relevant information
5. Screen color and formatting are good enough to work with expert system
6. Navigation was easy to find information
7. Searching of information was easy
8. Operation of Expert System seems to be easy
9. Web pages download (opening) was well in time
10. Need more learning and training on Expert System
78.4% respondents says that user screen of software were informative enough in understanding the queries being asked. 43% was of opinion that multimedia was sufficient in explaining expert system diagnosis. 64.6% were agreed with readability and visibility of web contents. Layout of content was perfect as per 64.1% of respondents. 57.6% says that screen color and formatting were good enough to work with expert system. On asking ease of operation with expert system with respect to navigation, searching and overall operation agreement shows by them is 92.7%, 71.14% and 78.7% respectively. 18.56% stakeholders did not responded for evaluation probably because they were not well versed with Information and Communication Technologies (ICT). It is also clear with opinion of 92.4% responded for more learning and training on expert system. 88.9% of respondents says that web page download time was quite well in time.

Fig 10: Opinion of stakeholders for expert system usability
4. Conclusion

Expert system development has been considered as complex and knowledge driven process. Keeping in view the limitation of web technologies, web-based expert system development further increases its complexity. Yet, the features available in web applications specially remote accessibility, centralized updation, minimized system dependence at client-end, cost of service and multi-platform availability cannot be ignored. Efficiency of web application depend on many factors as; web server technology, server processing efficiency and load, communication channel speed in between client computer and web server; efficiency of client computer itself; and also the complexity of web application. Efficiency of first three factors cannot be guaranteed. But, complexity of web application may be managed by an efficient software development methodology.

There are many limiting factors that influence the complexity of the software under development such as business / process requirements, methodology adopted, software engineer expertise, implementation tools used and also end user requirements. Methodology adopted in our work provides a powerful mechanism to optimize processing at server end and download time at client end for web based expert system. Optimization could be achieved in following steps of a software development life cycle.

1. **Analysis:** Work distribution among agent (Agent model); Minimum tasks performed by agents (Task model); Optimized algorithm and task knowledge (Expertise model); Optimized communication and coordination among agents and between agent and user (Communication & Coordination model)

2. **Design:** Selection of knowledgebase structure, implementation tool, agent architecture and agent network.

3. **Implementation:** Model-View-Controller (MVC) model best support optimization for web based applications, as it divide the software in Model, View and Controller components. In this architecture server side processing optimization could be achieved in Model component with optimized algorithm and data access method. User end optimization could be achieved with Client-side code and contents optimization strategies in View component.

The methodology adopted and the results obtain indicates that while developing web based expert system one should adopt following guidelines for highly usable system.

- Use agent-based approach as explained in methodology implemented.
- Use concept and antecedent combination to define knowledge. In this, domain knowledge will consist of domain ontology and domain model.
• Entity-Relationship modeling approach should be used in database design. In this two type of entities are used viz., data entity and rule entity. Data entity defines and stores domain ontology while Rule entity stores the relationship among ontology as defined by domain models.

• Use Model-View-Controller architecture in software design as it easily separates the View component of software from Model (processing) component. Using MVC approach, optimization at both client and server end may be carried out separately.

• Implementation should follow both speed and value optimization techniques. Speed optimization may be implemented with server side processing optimization (dynamic code optimization) as well as optimized downloadable code (client-side processing optimization). An efficient strategy of structure and placement of web contents as well as multi-media elements should be adopted for value optimization.

5. References


