Chapter 3
Generic Architecture of Multi-agent System for Agricultural Activities

3.1 Introduction

The inspiration for this research was taken from cognitive science. This interdisciplinary scientific study focuses on how information is represented, processed and transformed within humans and machines [28]. Today’s cognitive science provides a totally new insight to optimize the performance of knowledge systems. Three factors are important in learning newly designed knowledge systems to match the skills of a human expert: the building of experience through implicit learning; the use of examples to construct reasoning; and the development of strategies through explicit learning [82]. With appropriate social skills computer agents would perform more as humans and can be used as procurement agents, personal agents, monitor agents, etc. The architecture of cognition is able to contribute a great deal to an alternative way of reasoning for expert systems.

Figure 3.1: Cognitive Architecture
The multi-agent system as shown in Figure 3.1 that is developed as part of this research is based on this cognitive science architecture consisting of experience (implicit), example (reasoning) and strategy (explicit). Problem-solving is one important form of intelligent behavior, where the goal is to find a solution which satisfies certain criteria. Lessons learned from the past in problem solving and valuable information tracked from the previous events are very important for knowledge sharing. Research provides ample examples which can be put to good use when they are conveyed to users at the right time. All of them put together provide ample support and bring viable solutions to everyday problems faced in numerous domains. The concepts of cognition that are covered are very high level and how they are actually implemented through multi-agent systems are covered in detail in this and the next chapter.

The chapter organization is as follows. Section 3.2 presents the general architecture. Section 3.2.1 of this chapter covers the details of front end layer. The design of repository layer is covered in Section 3.2.2. In Section 3.2.3 different kinds of agents used in agent layer are presented. The technologies used in back end layer are talked about in Section 3.2.4. Section 3.3 concludes this chapter.

3.2 General Architecture

There are various components involved in the final prototype developed using the broad strategy mentioned above. Three tier or n-tier architecture is often used in client server environment. Each layer is allocated to carry out different responsibilities of the application. This type of layered architecture hides complexity and gives the flexibility needed when a component needs to be upgraded or modified. This type of modular design makes components in each layer to be more reusable. The concept of multi tier architecture defined in the chapter 1 is considered here. As stated, the framework is designed to consist of four layers. Figure 3.2 illustrates design of front end layer, repository layer, agent layer and back end layer.
Front end layer is the key gateway to users. Information is presented and collected from them through this layer. Since we live in a visual medium it has become more important to give equal importance to the way information is presented to users. While using the latest technologies available, user needs and their adaptability levels are also kept in mind.

A major portion of data is store in local databases and MySQL is used for this. MySQL not only has tables to hold local knowledge base but also stores mapping information. Ontologies are also used in this prototype. Ontology information is stored in files.
This prototype is developed using multi-agent systems so agents play a key role. There are various agents developed like utility agents, profile agents, data retrieval agents, update agents and monitoring agents. Each type of agents performs specific functions to make sure system performs smoothly. The specific workings of each type of agents are covered in the agent layer.

A lot of third party tools are readily available that are used in the prototype for retrieving/managing data. All of these tools are called from within agents developed in the prototype. The unpretentious integration platform of agents makes it possible to keep the design open ended for future enhancements. Back end layer covers the details about these tools and the how they are integrated into the design.

3.2.1 Front End Layer

Front end layer is the key gateway for users through which they receive and update data. Users are oblivious of other layers and the technologies used in accessing the content. Multiple interacting agents solve the information overload problem at the same time intelligence is added in them to personalize the content and apply the appropriate filters based on user profile and other settings.

Getting the proper content not only saves time from having users sort through hordes of unwanted stuff but also where internet service is still a luxury or where the connection speeds are low it is very crucial to get to the target data in less page clicks. User needs can be collected through user profiles and preferences and their behaviors can be tracked and logged so that these can be used for customization when data is returned in future. User profile data can be updated at any time as specifics about their wants or environment change. The same goes with their behavior data which is stored in the application database and it is continually tracked and updated.

3.2.1.1 Personalization

Every individual needs and desires are different. Most of them have specific needs why they access or use a system. Based on the traits and preferences of each individual user content and services can be optimized by applying personalization techniques. Nowadays all applications, either Web-based or not, call for personalized behavior. Personalization is
based on user profile information collected mostly during the process of creating their accounts. Some of the information captured while building user’s profile is their preferences, their background, their roles and the environment they work under. This information is stored in some form of database and used while retrieving and presenting content [66].

![Figure 3.3: Personalization Techniques](image)

Along with providing the right content very effectively and efficiently users should also be oblivious of the backend complexities. Vast amounts of information on the Web and the different modes of communication demand special attention and care to personalization techniques be it collection of user preferences or logical application of that information used during information retrieval. Figure 3.3 shows the personalization methods used in this prototype.

In the prototype application created when a new user registers their information is collected through user profile page. This information is passed on to agents and they stored it in database and are used when displaying content to users. Agents also allow users to update their profile information at any time later if needed.
Below are some of the ways in which user profile information is used to personalize content:

- Weather is a crucial element especially for farmers. Collection of location information like pin code helps give them services like weather report on all the pages in a unified and up to date manner.

- There are different core areas covered in the prototype. Since users might be interested in particular core areas having an option to specify that in user’s profile page filters content directly to that area instead of users having to select it each time.

- Past experience and knowledge level determine how a new article can be taken by users. “Research articles” are very informative but they are in a totally different genre and an uneducated person can make very little use of it. So capturing experience and knowledge level of users helps get pertinent content for them.

- Not all farmers play the same role in real life, some might like to follow traditional methods of farming and some might be constantly looking for new innovative methods available. Since there is an overload of information in all categories it is all the more important to use these kinds of personalization techniques.

3.2.1.2 Feedback and Filters

One bit of information leads to a couple of facts, which lead to multiple articles, then to some videos probably not in the same order. The point is a small of piece of information might take a few minutes to a few days to comprehend depending on the traces of information we follow. This is a good thing that we have an abundance of information but it overwhelms and burdens our brains. This is when application of filtering methods comes in handy. Filtering is the means by which unwanted data is weeded out as only pertinent information is retained.

Filtering can be done manually and automatically. Regular information retrieval systems have some kind of filters but they are not sophisticated enough so we are left to still sort through the results resulting in waste resources. The distinction between relevant and non-relevant data is always not very clear. Over the time interests and circumstances changes and users data needs also change. All of these have to be kept in consideration while
designing the systems. The design should be flexible enough to let users update their preferences when needed.

Some of the inputs for filtering methods come from user’s profile and some are collected as they navigate through system. The more they use the system and give their feedback the better it will be tailored to their needs. Figure 3.4 shows the filtering methods used in this prototype.

Figure 3.4: Filtering Techniques

Below is a brief description of the four different filters that are used along with personalization techniques.

- There is lot of content that is stored in local knowledge base and users can rate the content at any time. When a user provides ratings it might be either they like it or not so impressed with it. Capturing these interests can give clues on what they would like to see in future. All the content in local knowledge base has keywords attached to them to identify what kind of content it is like “organic farming”, “tips”, “fertilizers”, etc. Each time user likes content the associated keywords for the resource are stored in their user profile. One type of filtering is done based on these keywords. This will help bring similar content that they liked before. Past viewing history and their feedback is an important indication of the trending likes of users.

- Rural areas still lack the luxuries of having continuous high speed internet access. With the help of personalization the application can get the right content to users but if pages
take a long time to load or never load then there is no use of all the customization efforts. Information collected from profile page like internet connection speeds, farm size, etc helps to determine the type of filters to use. These filters will be applied for content with high download times.

- All the searchable content that is in local database has a title given to it. A background agent runs continuously to get all the titles from searchable content and stores them in a separate table. This agent runs all the time at specified periodic intervals. When a user starts entering keywords in a search box then auto fill feature will populate all the existing titles based on the letters entered for that category. When a user is not sure of how to word the search criteria this helps by seeing the titles of content already existing.

- The users of the system come from different background and each have their own experiences and also each one can give their own ratings for the content. Collaborative filters work based on these cumulative ratings. These ratings provide view of a greater community of users and can be very helpful to others.

3.2.1.3 Fuzzy Logic in Users Profile

Information on user profile page is captured using natural language conversations. There is always a certain degree of vagueness and uncertainty because same words mean different things to different people and most of the times it is difficult to translate the linguistics into crisp values. This can be handled by using fuzzy linguistic variables. The two main pieces of data considered in this application as inputs to fuzzy system are the “size of the farm” of user and their “internet connection speeds”. There are lot of other input variables also which are not well defined but in the farming application based on these two input variables the output can be significantly different so these two are taken to show how the proposed system can work effectively.

There are various plans available for internet services offered by different companies. The same plan when offered by different companies might offer different connection speeds. Similarly farm sizes cannot be easily categorized. A certain amount of acreage might be very small farm for some and be medium size for some others. These two combined together
brings out a new dimension on the kind of content that will be useful and also which can be delivered to them easily.

On the users profile page information in these two input fields are collected as raw inputs meaning users can enter the data as it is without having them to guess what kind of category their inputs fall under. These two inputs will be fed to a fuzzy system that has a set of rules defined. Based on the rules fuzzy system generates fuzzy output that will be used to determine the content to be presented to users. All these values (input and fuzzy output) will be stored in database in userprofile table. As shown in Figure 3.5 when a user decides to update their profile data if there is any change in the fuzzy input variables then fuzzy logic will generate new fuzzy output and these will also updated in the database.

Figure 3.5: Fuzzy Systems
3.2.2 Repository Layer

Once of the main objective of this research is faster easier access to data and one of the main source of information is local knowledge bases. Soil testing information is also part of the data that is used in the system but it is stored solely in ontology format. Keeping data in ontology format makes it easily portable. Detailed database design and ontology models are presented in this section.

3.2.2.1 MySQL

A lot of data for this application is stored in local knowledge base and for this a good database that is easy to implement and use is very much needed. After looking at the various options available in the market MySQL is chosen for this research work implementation. MySQL is open source database management software. MySQL is readily available freeware and it has the capability to execute queries from Java which is the language in which agents are developed. It is very fast and robust with a good feature set. Installation of MySQL is very easy and has excellent backup feature also.

Tables are designed after careful examination of the requirements and modified as the prototype is built. Care is taken to maintain some flexibility for future enhancements. To increase the execution speed of complex queries normalization techniques are used. This eliminates unnecessary duplication of data. All tables are built with by carefully choosing a good field as primary key. This speeds up the queries and also ensures data is unique and the problem of duplicate data is eliminated. Where ever appropriate foreign keys are set up which helps with data consistency.

Figure 3.6 illustrates the database model used in this prototype. All the tables used in the application are presented here.
Figure 3.6: Database Model
3.2.2.2 Ontologies

The quality of soil is an important factor to give better yield for any crop. The amount and type of fertilizer to use depends on the type of soil. So it is very important for farmers to get soil tested before sowing seeds. This prototype incorporates an easy Web interface to search for nearby soil testing locations. The same data can be stored in databases but this prototype shows how Ontologies can be used to make it a Semantic Web interface. Semantic Web is driving the evolution of current Web and it relies heavily on the use of ontology. Ontology is an important representation model for semantic Web.

The ontology in the application is developed using a free open source tool called Protégé [110]. Protégé is a tool based on Java, it is extensible with the help of plugins, and provides a good foundation for customized knowledge based systems. Jena is a Semantic Web framework that is developed by Brian Bride at the HP Semantic Web laboratory. Jena provides an API for reading, writing and processing RDF data along with a rule based engine for reasoning [63]. A part of the application data is stored exclusively in Ontologies and data is retrieved using Jena queries [4].

The first step for constructing soil testing interface is to look at the requirements and organize classes and properties. This is similar to database design so care should be taken to normalize structures and also keep it open for further enhancements. Table 3.1 gives the classes and the corresponding properties.

<table>
<thead>
<tr>
<th>CLASS</th>
<th>STATE</th>
<th>Property</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>CLASS</td>
<td>CITY</td>
<td>Property</td>
<td>Name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>SubOrganizationOf</td>
<td>STATE (CLASS)</td>
</tr>
<tr>
<td>CLASS</td>
<td>LOCATION</td>
<td>Property</td>
<td>Name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Address</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Phone</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>part of</td>
<td>CITY (CLASS)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TestingType (Array)</td>
<td>TESTINGTYPES (CLASS)</td>
</tr>
<tr>
<td>CLASS</td>
<td>TESTINGTYPES</td>
<td>Property</td>
<td>Name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Description</td>
<td></td>
</tr>
</tbody>
</table>

Table 3.1: Ontology Classes and Properties
Once the classes and properties are established the next step is to build the classes and their properties in protégé as shown in Figure 3.7.

![Ontology Editor](image)

**Figure 3.7: Ontology Editor**

The model for this is developed using protégé and the resulting ontology is stored as an owl document. For implementation purposes this document can be stored in any database but for this application it is left in owl format. This shows another way to access data from Ontologies without the need of databases. Leaving it in owl format makes it easily portable.

The creation of Soil Testing ontology is given here. The design can be modified easily if necessity arises. Jena API is used for running queries and getting the data from ontology and the details of these are given in the next chapter.
3.2.3 Agent Layer

Agents play a key role in the prototype developed. They carry out all the important functions like creating and updating user profiles, browsing and searching videos, links and advices, and performing monitoring actions. Below (Figure 3.8) is an illustration of the different agent types used in this prototype and how they interact with each other.

![Figure 3.8: Agents used in the Framework](chart.png)
3.2.3.1 Utility Agents

These agents provide services which will be used throughout the application. They are not dependent of the page they are called from so will return the same result always for the same search parameters. The concept of utility agents is to provide certain information when called on. They can be called by users or by other agents in the system. This separation of duty to get certain independent pieces of information relieves other agents which are tackling much bigger problems. They come in handy while retrieving data that is not tied to the search parameters.

The two utility agents that are developed in this prototype are CropAgent and CategoriesAgent (Figure 3.9). In this application categories represent a broader division of farming like fruits, vegetables, nuts, etc. Each category has related crops associated with them. Under fruits categories there are crops like mango, banana, apple, etc. These two together play a major role when users are searching for links, videos, advices, etc. The first step would be to pick a category this will bring back related crops and populates crop drop down box. Then search can be narrowed to a particular crop. When these actions are invoked corresponding agents get related data by querying database. If new categories or crops are added to database they will be reflected immediately in the drop down boxes without any delay.

Figure 3.9: Categories and crops
3.2.3.2 Profile Agents

The main goal of profile agents is to manage user profile data. There are two modes to access the application admin mode and normal mode. The features for both the modes are different and as is obvious from the names admin’s have more data update features. For a normal user the first step to accessing the application is by creating their profile in the system. When a new user creates his profile on the prototype application page their data will be stored in database by User profile agent. The usage of fuzzy linguistics on user profile page is already mentioned before. User profile agent handles the fuzzy logic and is responsible for storing fuzzy output in database so it can be used where appropriate in data retrieval process.

There are two tables that this agent talks to. One is userprofile table and another one is userpreferences table. Along with creating a new entry in userprofile table when a new user registers this agent will also be responsible for handling any future updates done by users to their profile data. Every time there are any changes to fuzzy input variables fuzzy logic is run again to get the new output and this new output is stored in database. Along with the basic profile data this agent is also responsible for updating userpreferences table. All the content in the local database are associated with keywords that identify the kind of content. Any time a user gives his rating to an article the ratings will be stored in the database and the keywords associated with that article will be stored in the userpreferences data. These keywords will be used in customizing the content on future searches.

3.2.3.3 Data Retrieval Agents

One of the main uses of the application is to retrieve information. The prototype developed is not just plain information retrieval system there are various facets that enhance the output and user experience. The concept of these agents is so unique that they can be used to retrieve information for any category like videos, links or advices. These agents handle different kinds of requests and intelligence is added into the agents to pull the right kind of content.

Users can look for information in two different ways they can either browser available content or they can do keyword searches. When users are browsing for information the first step is to select a category. When a particular category is picked the related crops are
populated with the use of utility agents. Data available for that particular crop are returned to users. Instead of just retrieving the data and presenting them to users in this prototype application it is categorically arranged in multiple silos. The silos used in this application are “Latest Videos”, “Most Popular”, “Based on Previous Likes” and “Videos from Web”. This way of organizing and presentation enhance search results further. The data for the first three tabs comes from local Knowledge base and the last tab “Videos from Web” gets data by querying Web. If they cannot find the information they are looking for they can click on this tab and from within the same application data will be retrieved from web. This eliminates the need to open a browser and do the search all over again. If they find the piece of information useful with the click of a button it can saved in local database and next time any user searches for similar data it will be retrieved from local database without the need to make Web search.

With the kind of browsing feature just mentioned users can drill down to a particular crop in a category. But if users are looking for specific data and want to do a query based on keyword they can use search feature instead. This way they can go directly to the content they are looking for instead of browsing through all the available information in local database. Users can enter keywords in the search text box. As they start entering words a dynamic drop down keeps filling content like auto suggest feature. These suggestions are not random keywords they are existing content titles from the database. This helps users to formulate the right search keywords. Knowing what to search for is not just enough, framing the right questions is also equally important. With the help of this auto suggest feature the application is making it easier for users to search with the right keywords.

3.2.3.4 Update Agents

Since the main source of information for the prototype application is from local database it has to be updated periodically. There are two modes to access the system. One is for regular users and one is for admin’s who are subject matter experts. A few of the sections like giving advices to users are given only to admin’s. Users are looking for advices and subject matter experts are the best people to answer them so this feature is off limits to regular users. Along with admin’s of the system regular users are allowed to add content through user interface. When they find any useful videos or web links with just a click of a button they
can be added to local database. Since the Web search is done from within the application this is possible. This way the possibility of them uploading faulty or non working links is reduced. As more and more users use the system and add content to local database and give their feedback the system becomes more useful. This eliminates the need for dedicated people to do system maintenance.

3.2.3.5 Monitoring Agents

Along with searching and updating for data, the system also has to constantly do things to make sure database clean up is done and notifications are sent when needed. When a user is trying to find an advice and if they cannot find any answer then Email monitoring agent will send email to concerned people who can answer the question. This way questions can be answered in a more reasonable time. Similarly there are other agents which are run constantly performing data organizing and clean up acts. These agents are started when the system comes up and can be configured to run at certain times.

AutoSuggest is another monitoring type agent. This agent constantly looks at existing content titles and adds them to a different table in database called autosuggest. This helps to gather all the content titles from like videos, links and advice tables at one location. The titles gather in autosuggest table will be used when users are trying to enter search keywords. When a user is trying to enter keywords to make a search based on the words they are typing titles from autosuggest table will be populating automatically to frame the right questions. This agent keeps running continuously at a specified interval once it is started.

3.2.4 Back End Layer

A lot of tools are readily available in the market that could be used in the application for querying/managing data. Most of these tools are straightforward to use and can be easily accessed from within agents. Back end layer talks about the various communication methods that are available and the tools used in the research which enhances user portal experience.
3.2.4.1 Information Sources

For a long time cooperative societies, village fairs and radio have been main sources of information for farmers. Most of these sources do not exist anymore or their significance has greatly diminished. The rapid explosion of internet created new mediums of information delivery. These new sources not only provide up to date information but also loads of education material in them.

In an article titled “Program Delivery Methods” [60] different educational delivery methods were classified into three groups: reinforcement, experiential, and integrative. According to the article to promote effective and efficient learning, a delivery system should comprise of methods wherever possible that provide desired experiential opportunities for the learner, augment the learning, and provide opportunities to integrate new information with existing knowledge and skills. Some of the factors that should be considered in the delivery of educational information were identified as below:

- Target audience and their educational objectives;
- Content and the type of message being provided;
- Nature of the delivery method; and
- Learning support provided.

Different stages of the adoption process require different methods of communication. To create awareness and interest in the adoption/diffusion process media types of information sources may play a more significant role while interpersonal methods may be more important in the trial, evaluation, and adoption stages. Some forms of information sources are given below along with that the integration methods used to gather and organize the data that are used in this prototype are also looked at.

Science and Technology has bought lot of revolutionary methods to agricultural development. Agricultural universities [97] [18] [119] are heavy into research and constantly strive to generate technologies for increasing farm production and income to farmers. They not only provide educational content but also agri-extension services which involve bridging the knowledge gap through farmer awareness campaigns and seminars. These initiatives provide opportunities for rural youth and women to learn and adopt
improved agro-technologies. Every agricultural university and research institutes maintain their own websites. A lot of them maintain online libraries so that not only research content but basic information on different kinds of crops is also maintained. The information has to serve not just farmers interested in new technologies but also people who are entering new into this industry. Some of them maintain content in native languages so they can better server local people.

Lot of portal sites are geared more for medium and large scale farmers. The focus is on creating web tools so that lot of book keeping can be done online. It can be used not just to store data but when information is entered for a couple of year’s analysis tools can be used to plan for future crops. Specific tools that are available on most of these portals lets farmers enter their criteria and get suggestions for future crops.

There are lots of farmers who are interested in new innovative cost effective sustainable farming methods. Villgrow [115] is an example of one such website. The goal of these portals is to make innovations easily accessible with enough support, training and expertise. This is a wonderful way to bridge the gap between research institutes and ordinary farmers.

Usage of phones has increased exponentially over the years. It is easier to transmit information to rural areas through phones than computers. Phones are lot easier to use than computers since they have fewer limitations when power and connection issues are concerned. There are a lot of SMS services [114] [54] which are developed to deliver information to farmers through a phone. Information about weather, crop prices change very rapidly and dynamically. The information about these need to reach farmers in a timely manner and SMS services deliver it in the most cost effective way. Information is available in different sources and across various mediums, in the next section how third party open source tools can be used to navigate through the data is presented.

### 3.2.4.2 Integration Tools

The explosive use of the internet and the ease of data creation helped users get closer to the information they need but also is aggravating information overload problem. What has started as a few websites has slowly grown into a few hundreds and then countless hundreds of websites [3]. The number of active websites keeps growing. As of 25th July
2008, Google software engineers Jesse Alpert and Nissan Hajaj announced that Google Search had discovered one trillion unique URLs [85].

It is not easy to navigate through all the individual web sites to find the right information. With the use of simple querying/correlating tools this new age problem of information overload can be tackled to a certain extent. The three tools that are used in this research are Yahoo! Pipes, Google API and YouTube API. They are very simple to use and can easily be integrated into agents. The working details of these are given in the next chapter.

Yahoo! Pipes is a web-based tool developed by Yahoo! in 2007. It is visual programming environment that enables users to "rewire the Web". Input for the pipes is sourced from the Web (RSS feeds, web pages, raw data). This is a dataflow system in which the input data flows through an interconnected set of modules that act upon it, ultimately producing some result. Pipes name is inspired by the concept of pipes in UNIX operating systems that enabled the composition of command line sequences of UNIX tools through which data “flow” for processing [121].

Yahoo! Pipes is a powerful composition tool to aggregate, manipulate, and mashup content from around the Web. Collection of appropriate RSS feeds can easily be integrated and manipulated in Yahoo! Pipes. This helps users to get all the content they need at a single place instead of visiting multiple Web sites. Pipes are very customizable depending on the needs of various users and can be easily modified to suit individual needs.

As can be seen from Figure 3.10 instead of visiting n number of individual websites rules can be mentioned in pipes based on the required output. Pipes provide a wide range of data aggregation and filtering methods so content from the web can be easily mashed up.
Figure 3.10: Information Gathering from Web with the use of Yahoo! Pipes
Searching multiple websites is not just time consuming but also the limitations of internet connection speeds hinder the process. No matter how resourceful a web portal is if has to go to web for all queries then with the varying connection speeds the data cannot be loaded in a timely manner. This can be overcome to a certain extent by using local knowledge bases. Data retrieval times can be drastically improved by extracting data from local knowledge bases. Data is more authentic and controlled than getting it from internet. The downside to this approach is increased overhead of data maintenance. Information is constantly changing and knowledge bases have to be updated continuously.

A mid way approach taking advantage of both Web and local knowledge bases. Agents can be designed to use knowledge bases along with Google and YouTube API. Google and YouTube have their own API’s which makes it easy to integrate and use them from within agents. When user queries first it will be looked up in knowledge bases if the information is not found then searches will be extended to retrieve content from internet using API’s like Google API and YouTube API. If the users find the links/data from the web useful then it can saved in the local databases and ratings can be given so the next time a similar search is made data retrieval will be faster. This way a separate process or procedure is not needed to maintain local knowledge bases.

3.3 Conclusion

The concepts of agents and multi-agent systems are covered in previous chapter. This chapter presents the general architecture that was proposed in this research work. Cognitive science provides new insights to improving knowledge systems. This branch of science focuses on experience, example and strategy. The inspiration for this research is taken from concepts mentioned in cognitive science and this chapter describes how they can be achieved in multi-agent systems.

The generic design consists of four layers front end layer, repository layer, agent layer and back end layer. Front end layer is the interface through which users interact with the system and how they receive the results. One of the main goals of the system is to keep design simple and data well organized. This chapter covers the various methods used to customize the data and how fuzzy logic is used in handling uncertainty and vague information. Repository layer holds the data needed for this application. Database design and the details
of the tables are given in Section 3.2. MySQL is used to hold the data. Along with data stored in MySQL soil testing information is stored in Ontologies. Information on how to build classes and properties and how to add instances to them is presented in this section. The actual methods of retrieving information from them are given in the next chapter.

Section 3.3 explains the different types of agents that are used in this prototype. Utility agents provide services which can be called from other agents. The two utility agents which are used in the application are CategoryAgent and CropAgent. Profile agent handles user related information. Users profile data and preferences are created/updated with the help of this agent. Searching for information is one of the main features of this prototype and this is handled with the help of data retrieval agents. Similarly update agents help to create/update information in database. Monitoring agents are run continuously. Two of the agents that provide monitoring services are EmailAgent and AutoSuggestAgent.

Next section covers back end layer. The various sources of information methods available right now are visited. The various ways to aggregate different information sources and present just the right kind of content are mentioned in Section 3.4. This section also demonstrates how Google and YouTube API can be used to better handle information extraction process.