Implementation of Experimental System

Content

5.1 Introduction to Grid-based Model........ 123

5.2 A Grid-based Model for Integration of Geographically Distributed & Heterogeneous Educational Resources for Knowledge Extraction & Delivery: A Case Study........................................ 124

5.3 Students’ Performance Evaluation: An Experimental System......................... 128

5.4 Conclusion........................................ 165

References............................................ 165
Chapter 5

Implementation of Experimental System

This chapter introduces implementation details of an experimental system. The experimental system is developed based on the generic framework for integration of a multi-agent knowledge-based system and distributed database grid. It also follows a methodology mentioned in the previous chapter. As an experimental system, Students’ performance evaluation has been chosen. Students’ performance evaluation system uses the data grid service agents in order to expose the heterogeneous and distributed databases in grid environment. It implements various agents to realize the data grid services. Moreover, it implements the fuzzy interface agent to realize the knowledge-based component, which assists the users in the decision making process at a certain level.

Section 5.1 & 5.2 of this chapter explain the grid-based model along with a case study of the integration of geographically distributed & heterogeneous educational resources resides in the typical university domain for knowledge extraction & delivery. Section 5.3 demonstrates the students’ performance evaluation as an experimental system, which is implemented based on the generic framework and grid-based model. The data grid service agents are demonstrated in the previous chapter. This section focuses on implementation details of domain specific agents and fuzzy interface agent. It also includes various aspects of the design and development phases of an experimental system in detail with necessary screen layouts and code snippets.

5.1 Introduction to Grid-based Model

The goal of the grid-based model is to provide a uniform query and browse mechanism for multiple distributed & geographically distributed university databases, shield their dissimilarities in data schema or location, use multi-agent knowledge-based system for retrieving data from existing databases, and provide a new solution for the research on the integration of the domain specific databases.

University domain is a very big and complex domain. In a big and complex domain such as university, students’ progress monitoring is a key task. Therefore, we have implemented an experimental expert system for students’ performance evaluation. The database grid is
developed to integrate some of the university databases which may span across geographically scattered locations. The databases are heterogeneous by nature and geographically distributed. Any database node can come and join the database grid dynamically. The number of active nodes in database grid is also identified and displayed to privileged users. The agent resides in the system access and integrates data from database grid. Several task agents are implemented to carry out the different activity like user management, data access & integration, resource management, student profile management, student progress monitoring & path advising, fuzzy interface agent, report generator agent etc. Periodic evaluation of students is performed by applying fuzzy set theory and fuzzy logic and reports and advices are generated based on that. Criteria like students’ attendance, term work marks and internal exam marks are considered as input parameters for the fuzzy control logic system. The performance of the student is the output parameter which generates after applying defuzzification process. This chapter includes the implementation details of the experimental system like entities of the system, integration of databases into database grid, classes, packages to implement agents, rule base and inference engine for fuzzy system and so forth.

5.2 A Grid-based Model for Integration of Geographically Distributed & Heterogeneous Educational Resources for Knowledge Extraction & Delivery: A Case Study

While many modern academic applications (such as digital library, e-learning, etc.) involve accessing heterogeneous databases which are located in a geographically distributed environment, there is an increasing demand on integration of different data resources across different information systems or administrative domains to provide a uniform access interface for users. Therefore, effective integration of heterogeneous university databases has been considered as one of the most challenging fields. Emerging as a new paradigm for distributed computing, grid is defined as the secure, flexible, coordinated resource sharing mechanism among dynamic collections of individuals, institutions, and resources [2]. As a prospective solution for managing and collaborating distributed resources, data grid deals with the issue of heterogeneity by developing uniform and standardized interfaces for data access and integration of dissimilar data resources [5].
Universities often use information systems for operational and strategic tasks. These information systems may have been developed using different protocols, technologies, architectures and platforms. These systems containing a large number of digital objects that are stored in many different storage systems and data formats with differences in their schema, access rights, metadata attributes and so forth. In such scenario, it is a very difficult and challenging task to access these digital objects. For that, this case study proposes a grid-based model for sharing these academic resources in the heterogeneous environment. This model will improve the accessibility, integration, controlling and management of these academic resources.

As data grid technology deals with sharing distributed heterogeneous resources without compromising local administration and provides functionality to federate heterogeneous data resources, here, a new model for integrating heterogeneous university databases has been developed. The goal of the model is to provide a uniform query and browse mechanism for multiple distributed university databases, shield their differences in data schema or location, and provide a new solution for the research on integration of university databases. There are good opportunities for exploiting synergies between grid and agents. The grid-based model provides a uniform and standardized interface which can be used to represent an abstract view of data resources, which can permit homogeneous access to heterogeneous university databases. Therefore, the said model is able to provide easy and transparent access to distributed heterogeneous resources span across different organizations and administrative domains.

Figure 5.1 shows the grid-based model for integration of geographically distributed & heterogeneous educational resources for knowledge extraction & delivery. In this architecture a university domain is seen as a decentralized collection of interacting self-interested agents where an agent represents the knowledge and interests of an individual user. The university is having a large number of digital objects stored in various distributed data repositories which may span across different organizations and administrative domains. To integrate these data objects in the data grid environment, we have used OGSA-DAI (Open Grid Services Architecture - Data Access and Integration) as a data grid middleware.
Chapter 5: Implementation of Experimental System

OGSA-DAI is a database-centric middleware solution which mainly provides the data virtualization services. We have used multiple agents which provide the user level services to the client applications and users. The users have to first interact with university portal to retrieve the information or knowledge stored in the form of digital objects. The grid-based model incorporates an agent-based framework which allows a university domain to be considered, designed and developed as an information environment being composed of

Figure 5.1: A Grid-based Model for Integration of Geographically Distributed & Heterogeneous Educational Resources for Knowledge Extraction & Delivery

OGSA-DAI is a database-centric middleware solution which mainly provides the data virtualization services. We have used multiple agents which provide the user level services to the client applications and users. The users have to first interact with university portal to retrieve the information or knowledge stored in the form of digital objects. The grid-based model incorporates an agent-based framework which allows a university domain to be considered, designed and developed as an information environment being composed of...
different co-operating agents. Apart from this, it provides access to database nodes via the OGSA-DAI server. Thus, the databases need to be opened only for the OGSA-DAI server(s). In a grid environment, where an application can run on a large number of machines, this is the only way to provide better security for database providers.

In multi-agent system environment, multiple collaborative agents are working together to accomplish the tasks. Currently, the multi-agent system environment provides four types of typical services namely presentation service, domain services, knowledge-based services and data grid services. The presentation services are provided by the interface facilitator agent. It provides an effective presentation of content and delivery mechanism. Also, it gets the request from users and sends a response back to them.

The domain services are application specific and customizable. In university environment, the typical domain agent could be user management agent, search agent, learning management agent, student profile agent, course management agent, advisory and help agent, e-communication agent, report generator agent and so forth. The user management agent is responsible to create users in the system, assigns privileges to them, authenticates and authorizes the user by using metadata management service. Search agent retrieves specific information based on queries fired by users. It may use AI (Artificial Intelligence) mechanism for query retrieval. Learning management agent provides uniform interface for accessing learning resources to the users. Learning resources may store in heterogeneous form and on disparate location. This agent hides such differences from its underlying users and provides transparent access to them. Student profile agent maintains the students’ profiles of entire university which may have different undergraduate and postgraduate departments. The various departments of the university may run different courses and the information about these courses are stored in digital form. The course management agent provides details about various courses to learners and visitors. It also provides an interface between instructor and system to manage and rearrange course content. Advisory and help agent provide support and help to instructors and learners. Also, they incorporate the decision-making processes by applying AI techniques. E-communication agent provides e-mail and chat facility among the users for communication. The report generator agent generates various reports by combining and using information stored in several distributed databases.

The knowledge-based services are offered by integrating a typical agent. A knowledge-based system is a system that uses artificial intelligence techniques in problem solving.
processes to support human decision-making, learning, and action. The knowledge-based system is used to provide intelligent decisions with justification. Knowledge is acquired and represented using various knowledge representation techniques such as rules.

The knowledge-based system consists of mainly two components: knowledge base and inference engine. A knowledge base is a repository of domain knowledge and meta knowledge. The inference engine is a software program, which infers the knowledge available in the knowledge base. The domain agents of the model provide effective and seamless information retrieval and the typical agent provides knowledge-based services uses AI mechanism to turn this information into something more i.e. knowledge. Knowledge is derived from information in a similar way as information is derived from data. The knowledge base can be used as a repository of knowledge in various forms. The Inference Engine infers the knowledge that is available in the knowledge base. Fuzzy set theory and fuzzy logic are a highly suitable and applicable basis for developing knowledge-based systems in university for various tasks.

As discussed earlier also, the data grid offered core, data and user level services. Through these services, the client applications and users may get access to digital objects of the data grid environment. The typical agents namely data access & integration agent, resource management agent and grid administrative agent provide the above said services. We have implemented these agents in our experimental system and the detailed methodology to implement these agents is already discussed in the chapter 4. Here, in this chapter, we will discuss the structure of the experimental system, implementation details of domain specific agents and fuzzy interface agent to provide knowledge-based services. This architecture can improve the accessibility, integration and management of educational resources.

5.3 Students’ Performance Evaluation: An Experimental System

We have developed the generic framework for multi-agent knowledge-based system accessing distributed database grid. Also, in the above section, we have developed a grid-based model for integration of geographically distributed & heterogeneous educational resources for knowledge extraction & delivery. The university domain is very large and
may contain several different modules. We have taken one of the modules for our experimental system i.e. students’ performance evaluation.

In a university, there are several different departments existing and they may manage their students’ data individually and in disparate form. The aim of this research is towards integrating these distributed and heterogeneous data repositories in the data grid environment. Once the data grid is formed, we have integrated it with the multi-agent system having knowledge-based component. This system performs knowledge extraction and delivery of data or information retrieved from data grid which helps students and teachers to evaluate students’ performance and assist them in the decision making process at a certain level.

In the current competitive era, every school, colleges and institutes strongly focus on improvement of the students’ performance by applying fair evaluation methods. Performance of students is difficult to assess before the final results are declared. Also, there are several factors are needed to be considered which affect the performance. Major factors that affect on performance are the attendance of students in their theory and laboratory sessions, the exam grades they obtained in internal exams and the exam grades they obtained during term work evaluation. It is possible to analyze these factors qualitatively such as poor, very good, average, excellent etc. Even though quantitative figures are available such as 55% attendance in theory session, 80% marks in internal exam, 70% marks in the term work evaluation; it cannot directly be used for performance assessment.

Suppose in the grading process of students’ performance, the borderline between a good performance and a bad performance is 50%. Suppose the student got 49% during the grading process. Can this performance considered as a bad? Or if it is 50%, can it consider as a good? So, assessing students’ performance is the area in which strict rules often do not represent the real situation. Also, can two students have the same final exam grade, one of which has been present in the majority of the sessions and the other only in half of the sessions? These are the criteria according to which the teachers can realistically assess students in order to evaluate their performance. There are two main reasons why above classical logic systems cannot deal with problems in which knowledge is approximate. They do not provide a means for representing the meaning of propositions expressed in a natural language when it is imprecise, and they do not provide a mechanism for inference in the cases where knowledge is represented symbolically along with its meaning [4].
As we can clearly see from the above discussion, developing a system for students’ performance evaluation based on strict and rigid rules would not be a truthful evaluation process. Therefore, we need systems that will deal with knowledge, which is rather imprecise or incomplete. Also, human routinely and subconsciously place things into classes whose meaning and significance are not well defined. Some of the examples are big house, cool season, rich man, beautiful woman, etc. Furthermore, we need a different kind of logic than the traditional Boolean logic. We need an approach to common sense reasoning based on fuzzy set theory and fuzzy logic. Fuzzy logic provides a way of representing the behavior of systems which are either too complex or too imprecise. Fuzzy knowledge-based decision making becomes especially applicable in the above systems.

Apart from this, the performance of the students’ should evaluate on a regular basis and regular interval so that they can get the chance to improve their performance before the final exam grades are declared. Also, faculties and teachers should evaluate the students’ performance track as a part of their continuous evaluation process. We have developed an experimental system which will assist the process of decision making of students’ performance. The system will utilize the fuzzy logic theory and develop the decision making process based on fuzzy rules to assess whether a student gets very poor, poor, good, average or excellent performance.

We have implemented multi-agent system on top of data grid mechanisms. In an experimental system, as the infrastructure of information management and processing, grid computing provides the ability of integration and sharing of enormous distributed information resources and autonomous systems. Thus, it provides a solution to meet the demand of retrieving information, sharing and analyzing data. Agents can be implemented by web services which are loosely-coupled, autonomous and distributed components that interact with each other through the exchange of messages. The implemented experimental system leverages of OGSA-DAI’s workflow execution functionality and out-of-the-box activities for data access, integration, transformation and delivery. We have implemented application-specific activities and thus provide support for application-specific data resources.
5.3.1 An Architectural View of Students’ Performance Evaluation System

Figure 5.2 presents the architectural view of an experimental system i.e. students’ performance evaluation. We can clearly see from the Figure 5.2, that there are multiple collaborative agents working together as a part of multi-agent system environment. These agents create and provide data grid services to the users.

**Entities of the System:** There are mainly three entities of the system. They are distributed & heterogeneous databases, multiple collaborative agents and users of the system. We have created a data grid environment in which these databases are integrated. These databases may part of different information systems or administrative domains. The multi-agent system provides a scalable environment to integrate this data grid to perform the knowledge extraction and delivery process and hides the heterogeneity of the database nodes from the users and handles the data access and integration in an efficient manner. Therefore, the system produces the dynamic, flexible and extensible architecture based on both agent and grid technology.

*Figure 5.2: An Architectural View of Students’ Performance Evaluation System*
Distributed & Heterogeneous Databases Reside in Database Grid

The distributed database grid of generic framework is able to integrate the heterogeneous and distributed databases. As we discussed in the chapter 3, currently it may integrate relational data resources - e.g. Oracle, MySQL, SQL Server, DB2 etc., XML data resources - e.g. Xindice and File-based data resources - e.g. files and directories. In the experimental system, we have taken three database nodes which contain relational data resources (e.g. MySQL, Oracle, SQLServer etc.) but we may also integrate XML and file-based data resources according to the application specific requirement.

These databases are located at three different sites. As we can see from the Figure 5.2, the database at site A contains all the data of MBA department. The database at site B contains all the data of MCA department. Also, there is one more database node located at site C contains students’ final exam data. Each site may contain different relational database type and schema. We have access and manage these geographically distributed and disparate databases through data grid services provided by multiple collaborative agents.

- Database Fragmentation

In the distributed database grid scenario, the database tables should be fragmented to achieve the highest benefits. Fragmentation has mainly two different types: Horizontal fragmentation and Vertical fragmentation.

**Horizontal Fragmentation:** Figure 5.3 demonstrates an example of horizontal fragmentation. Table A stores the students’ data which are having fields like student ID, student name, city, course etc. It makes sense for the university to split Table A into different partitions or fragments based on the students who study in that department.

This type of fragmentation makes the management, queries, and transactions for Table A convenient and efficient. The down side of this type of fragmentation is that, whenever a query involving all records of students who are studying in university at a given time, it has to request all partitions or fragmentation from all sites and do a union with them [6]. This is called data integration and we have implemented a typical agent named data access & integration which is responsible to send a query to multiple databases at a same time, collects the data or response from these multiple databases, makes the union of query
results and passes it to the user. It exposes multiple databases to users as a single virtual database and thus hides the heterogeneity and provides transparent and single point access.

**Vertical Fragmentation:** Figure 5.4 demonstrates an example of vertical fragmentation. In vertical fragmentation, the columns of a table are divided up among several sites located on the network. Each such partition/ or fragment must include the primary key attribute of the table. This type of fragmentation can have a sense when different sites are responsible for processing different operations involving a table entity. For example, the student personal information like identification code (ID), name and city of Table A might be stored at one site and his academic information like course and semester of study might be stored on another site. Both partitions, must include the primary key attribute i.e. student ID.
A downside of vertical fragmentation is that, a query involving the entire Table A would have to request all portions from all sites and do a join on them [6]. Here also, data access & integration agent is provided the service of standardized interface to send a query to multiple databases and get the result back.

We have used horizontal fragmentation in our experimental system where students’ databases are stored at different sites but having similar table schemas. The data tuples are different in these databases and they are relative to the department or course in which student is studying. We can later choose vertical fragmentation also as we have already implemented data access & integration agent who is able to pass the query to multiple databases and make the union of the result retrieved from the passed query.

**Multiple Collaborative Agents**

Multiple collaborative agents are the key components of the whole system as we can see in the Figure 5.2. Here, the agents may divide into two categories: User level agents, which are domain specific agents and can be customized and updated as per the requirement of an application and Data grid service agents, which are typical agents providing data grid services to users. Also, data grid service agents interact with OGSA-DAI, a data grid middleware, to access its services.

The configuration and implementation details about OGSA-DAI data grid middleware and JAVA web services are already discussed in the chapter 4 i.e. detailed methodology for development of framework. Also, the implementation scenario for data grid service agents is discussed and demonstrated in the chapter 4. Therefore, in this chapter, we only discuss about the implementation scenario for domain specific agents and integration of knowledge-based component i.e. fuzzy set theory and fuzzy logic through a fuzzy interface agent.

**Domain Specific Multiple Collaborative Agents:** In this section, we discuss the design and development phases of domain specific agents. In our experimental system, we have used following domain specific agents. User Management Agent, Student Profile Agent, Progress Monitoring Agent, Path Advising Agent, Search Agent and Report Generator Agent. The following is the detail of the services offered by each agent.
• User Management Agent

The typical services provided by user management agent are as follows. The agent is implemented as a user management web service in an experimental system.

- New user registration and user management;
- Verification of user id, password and their role;
- Direct the user to the related web page according to their type like admin, student or any other;
- In case of user type is a student, pass the resource id to the student profile agent; and
- In case of user type is admin, get the resource id list from the resource management agent so that an admin can pass the query to the multiple databases.

• Student Profile Agent

The typical services provided by the student profile agent are as follows. This agent is implemented as student profile web service in an experimental system.

- Provide access to see the student’s basic profile;
- Provide access to see the student’s advanced profile;
- Display course announcement at regular basis;
- Display backlog information;
- Show the student’s regular progress report; and
- Used data access & integration agent to accomplish the above mentioned tasks.

• Progress Monitoring Agent

The typical services provided by progress monitoring agent are as follows. This agent is implemented as a progress monitoring web service in an experimental system.

- Used to monitor the continuous progress of the student by generating reports at regular time interval;
- Three criteria’s are followed to monitor the progress: student’s attendance, student’s internal evaluation marks and student’s term work evaluation marks;
- Calculate the attendance, internal evaluation marks and term work evaluation marks of the specific student for the given time period and send this data set to the Path Advising Agent; and
• Used data access & integration agent to accomplish the mentioned tasks.

• **Path Advising Agent**

Path advising agent is a typical fuzzy interface agent which uses fuzzy set theory and fuzzy logic. This agent is implemented as a path advising web service in an experimental system. The main task of the path advising agent is to evaluate the student’s performance based on the data retrieved from distributed database grid nodes by using fuzzy set theory and fuzzy logic. The typical services provided by the path advising agent are as follows.

- Get the three parameters as discussed above from Progress Monitoring Agent i.e. attendance (%), internal evaluation marks (%) and term work evaluation marks (%);
- Perform the fuzzy logic procedure and evaluate the overall performance of the student by applying fuzzy set theory and logic. It is used jfuzzyLogic open source library to implement a fuzzy interface; and
- Pass the result to the report generator agent to show it in proper format.

• **Report Generator Agent**

The typical services provided by the report generator agent are as follows. This agent is implemented as a report generator web service in an experimental system.

- Generates report based on the data passed by progress monitoring agent and path advising agent;
- Generate chart in specified format; and
- Pass these generated charts to the student profile agent.

• **Search Agent**

The typical services provided by search agent are as follows. This agent is realized by a web service in an experimental system.

- Search for different requests according to the user’s requirements & passes the output to relevant user; and
- For that, it may access and integrates data from various distributed database grid nodes registered within the data grid environment through data access & integration agent.
5.3.2 Integration of Knowledge-based Component through Implementation of Fuzzy Logic

As we have discussed above, the generic framework provides a typical agent that provides knowledge-based services. To provide the knowledge-based assistance, the fuzzy set theory and logic is used. In our experimental system, the path advising agent is a typical knowledge-based agent that contains the methodology and algorithms to implement the fuzzy interface. The basics of fuzzy set theory and fuzzy logic is already discussed in the chapter 1 i.e. introduction and overview. In this section, we put emphasis on how to evaluate the students’ performance by using fuzzy logic.

❖ Fuzzy Logic Model

In this section, we have explained the fuzzy logic model and its important components. The fuzzy inference process is expanding to three stages and having five components. As we can see in Figure 5.5, there are mainly five components: Crisp Values, Fuzzification, Fuzzy Inference Engine, Fuzzy Knowledge base and Defuzzification. The Figure 5.5 demonstrates the fuzzy inference process. The details of the fuzzy inference process are as follows:

1. **Crisp Value**: Crisp value is passed as an input to the fuzzification stage.
2. **Fuzzification**: Fuzzification means crisp value is converted into fuzzy input value with the help of suitable membership function. Crisp inputs are fuzzified into linguistic values to be associated with the input linguistic variables.
3. **Fuzzy Inference Engine**: Once all crisp input values have been fuzzified into their respective linguistic values, the inference engine will access the fuzzy rule base of the fuzzy expert system. It derives linguistic values for the intermediate as well as the output linguistic variables.
4. **Fuzzy Knowledge Base**: A repository contains fuzzy “If Then” rules.
5. **Defuzzification**: Defuzzification means calculation of the final output with the help of suitable defuzzification method. Once the output linguistic values are available, the defuzzifier generates the final crisp values of the output linguistic values. The most common techniques for defuzzification are Center-of-Maximum (CoM) and Center-of-Area (CoA).
Chapter 5: Implementation of Experimental System

Figure 5.5: A Fuzzy Inference Process

♂ Linguistic Variables

The power and strength of fuzzy logic come from the ability to deal with vague linguistic variables. Linguistic variables are the input or output variables of the system whose values are words or sentences from a natural language, rather from numerical values. A linguistic variable is generally decomposed into a set of linguistic terms.

A linguistic variable is a variable whose values are words or sentences in a natural language. For example, let the temperature (t) is the linguistic variable which represents the temperature of a room. To qualify the temperature, we are using terms such as “hot” and “cold” in real life. These are the linguistic values of the temperature. Then, T(t) = {too-hot, hot, warm, cold, too-cold} can be the set of decompositions for the linguistic variable temperature. Each member of this decomposition is called a linguistic term and can cover a portion of the overall values of the temperature [1].

♂ Membership Functions

A membership function for a fuzzy set A on the universe of discourse X is defined as μA:X → [0,1], where each element of X is mapped to a value between 0 and 1.

This value quantifies the grade of membership of the element in X to the fuzzy set A and it is called membership value or degree of membership. Membership functions allow us to graphically represent a fuzzy set. The X axis represents the universe of discourse and the Y axis represents the degrees of membership in the [0,1] interval. Below is a list of the membership functions used frequently in fuzzy expert system.
**Triangular function:** Triangular function is defined by a lower limit \(a\), an upper limit \(b\), and a value \(m\), where \(a < m < b\).

\[
\mu_A(x) = \begin{cases} 
0, & x \leq a \\
\frac{x-a}{m-a}, & a < x \leq m \\
\frac{b-x}{b-m}, & m < x < b \\
0, & x \geq b 
\end{cases}
\]

**Trapezoidal function:** Trapezoidal function is defined by a lower limit \(a\), an upper limit \(d\), a lower support limit \(b\), and an upper support limit \(c\), where \(a < b < c < d\).

\[
\mu_A(x) = \begin{cases} 
0, & (x < a) \text{ or } (x > d) \\
\frac{x-a}{b-a}, & a \leq x < b \\
1, & b \leq x < c \\
\frac{d-x}{d-c}, & c \leq x \leq d 
\end{cases}
\]
There are two special cases of a trapezoidal function. They are called R-functions and L-functions. The description of these special functions is described as follows:

- **R-functions**: with parameters \( a = b = -\infty \)

\[
\mu_A(x) = \begin{cases} 
0, & x > d \\
\frac{d - x}{d - c}, & c \leq x \leq d \\
1, & x < c
\end{cases}
\]

- **L-functions**: with parameters \( c = d = +\infty \)

\[
\mu_A(x) = \begin{cases} 
0, & x < a \\
\frac{x - a}{b - a}, & a \leq x \leq b \\
1, & x > b
\end{cases}
\]
5.3.3 Students’ Performance Evaluation through Fuzzy Expert System

In this section, we have explained the implementation details of the fuzzy logic model that applied to develop a fuzzy expert system of students’ performance evaluation. The fuzzy expert system is realized by incorporating fuzzy set theory and fuzzy rules. To implement a fuzzy expert system by applying fuzzy control logic, jFuzzyLogic packages are used. A jFuzzyLogic is an open source fuzzy logic package written in JAVA. It implements Fuzzy Control Language (FCL) specification (IEC 61131 part 7). It is a standard that deals with fuzzy control programming.

The fuzzy expert system takes three input parameters for its calculations: Attendance, Internal Exam Marks and Teamwork Marks. Also, it generates one output parameter as a result i.e. Performance. The input parameters are supplied by the data access & integration agent. Data access & integration agent takes all these values from distributed and heterogeneous databases that reside in a data grid environment. The fuzzy inference agent of an experimental system matches the input provided by the corresponding rules of the rule base and produces a result fuzzy set. The given output defuzzifies value of the result fuzzy set. Figure 5.6 demonstrates a fuzzy logic model for students’ performance evaluation.

\[
\mu_A(x) = e^{-\frac{(x-m)^2}{2k^2}}
\]
Students’ performance evaluation with a fuzzy expert system comprised with three steps:

1. Fuzzification of input & output parameters.
3. Defuzzification of performance value.

**Fuzzification of Input & Output Parameters**

Fuzzification of input parameters is carried out using input variables and their membership functions of fuzzy sets. We have formed a set of linguistic variables with respect to their input parameters as well as output parameter.

Before applying the fuzzification, the range of possible values for the input and output variables are determined. In Fuzzy logic, these are called the membership functions (Input variable vs. the degree of membership function). Membership functions are used to map
the real world measurement values to the fuzzy values to apply fuzzy set operations on fuzzified values. The values of the input variables are considered in terms of percentage.

Fuzzification of performance is carried out by using three input parameters and their associated membership functions. For each input parameter, we have defined three linguistic values and membership functions. The fuzzy set of all three input parameters (attendance, internal exam evaluation and term work evaluation) and their associated membership functions are given in Table 5.1.

<table>
<thead>
<tr>
<th>Linguistic Values</th>
<th>Membership Function</th>
<th>Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Poor</td>
<td>Trapezoidal R</td>
<td>(3,5)</td>
</tr>
<tr>
<td>Good</td>
<td>Triangular</td>
<td>(3,5,7)</td>
</tr>
<tr>
<td>Excellent</td>
<td>Trapezoidal L</td>
<td>(5,7)</td>
</tr>
</tbody>
</table>

*Table 5.1: Fuzzy Set for Input Parameters*

For output variable, we have defined five linguistic values and membership functions. The fuzzy set for output parameter (performance) and its associated membership functions are given in Table 5.2.

<table>
<thead>
<tr>
<th>Linguistic Values</th>
<th>Membership Function</th>
<th>Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>VeryPoor</td>
<td>Trapezoidal R</td>
<td>(1,3)</td>
</tr>
<tr>
<td>Poor</td>
<td>Triangular</td>
<td>(1,3,5)</td>
</tr>
<tr>
<td>Average</td>
<td>Triangular</td>
<td>(3,5,7)</td>
</tr>
<tr>
<td>Good</td>
<td>Triangular</td>
<td>(5,7,9)</td>
</tr>
<tr>
<td>Excellent</td>
<td>Trapezoidal L</td>
<td>(7,9)</td>
</tr>
</tbody>
</table>

*Table 5.2: Fuzzy Set for Output Parameter*

The following Figures 5.7, 5.8 and 5.9 demonstrate the fuzzy membership function applied to input parameters and Figure 5.10 demonstrates the fuzzy membership function applied to an output parameter.
Chapter 5: Implementation of Experimental System

Figure 5.7: Fuzzy Membership Function for Input Parameter ‘Attendance’

Figure 5.8: Fuzzy Membership Function for Input Parameter ‘Exam Evaluation Marks’
Figure 5.9: Fuzzy Membership Function for Input Parameter ‘Termwork Evaluation Marks’

Figure 5.10: Fuzzy Membership Function for Output Parameter ‘Performance’
**Determination of Fuzzy Rules and Inference Method**

**Fuzzy Rules**: Fuzzy rules are the rules determine input and output membership functions that will be used in inference process. The decision, which the fuzzy inference system makes, is derived from the rules. These rules are a set of ‘If-Then’ statements that are intuitive and easy to understand as they are written in common English declarations. Here, ‘If’ refers to an antecedent that is compared to the inputs, and “Then” refers to a consequent, which is the result of the output. All the rules that have any truth in their antecedent will be executed. They contribute towards the fuzzy conclusion set. Rules used here are derived from interviewing experts belongs to academic field and also based on their experience in the area of students’ performance evaluation. These rules may further modify according to changes in academic strategies and policies or even based on academic experts. These rules are linguistic and are entitled as “IF-THEN” rules. The rules derived and framed for the research work are as follows:

**RULE 1**: IF Attendance IS poor AND Exam_Evaluation IS poor AND Termwork_Evaluation IS poor THEN Performance IS VeryPoor;

**RULE 2**: IF Attendance IS poor AND Exam_Evaluation IS poor AND Termwork_Evaluation IS good THEN Performance IS VeryPoor;

**RULE 3**: IF Attendance IS poor AND Exam_Evaluation IS poor AND Termwork_Evaluation IS excellent THEN Performance IS Poor;

**RULE 4**: IF Attendance IS poor AND Exam_Evaluation IS good AND Termwork_Evaluation IS poor THEN Performance IS VeryPoor;

**RULE 5**: IF Attendance IS poor AND Exam_Evaluation IS good AND Termwork_Evaluation IS good THEN Performance IS Average;

**RULE 6**: IF Attendance IS poor AND Exam_Evaluation IS good AND Termwork_Evaluation IS excellent THEN Performance IS Good;

**RULE 7**: IF Attendance IS poor AND Exam_Evaluation IS excellent AND Termwork_Evaluation IS poor THEN Performance IS Poor;

**RULE 8**: IF Attendance IS poor AND Exam_Evaluation IS excellent AND Termwork_Evaluation IS good THEN Performance IS Good;
RULE 9: IF Attendance IS poor AND Exam_Evaluation IS excellent AND Termwork_Evaluation IS excellent THEN Performance IS Good;

RULE 10: IF Attendance IS good AND Exam_Evaluation IS poor AND Termwork_Evaluation IS poor THEN Performance IS VeryPoor;

RULE 11: IF Attendance IS good AND Exam_Evaluation IS poor AND Termwork_Evaluation IS good THEN Performance IS Average;

RULE 12: IF Attendance IS good AND Exam_Evaluation IS poor AND Termwork_Evaluation IS excellent THEN Performance IS Good;

RULE 13: IF Attendance IS good AND Exam_Evaluation IS good AND Termwork_Evaluation IS poor THEN Performance IS Average;

RULE 14: IF Attendance IS good AND Exam_Evaluation IS good AND Termwork_Evaluation IS good THEN Performance IS Good;

RULE 15: IF Attendance IS good AND Exam_Evaluation IS good AND Termwork_Evaluation IS excellent THEN Performance IS Good;

RULE 16: IF Attendance IS good AND Exam_Evaluation IS excellent AND Termwork_Evaluation IS poor THEN Performance IS Good;

RULE 17: IF Attendance IS good AND Exam_Evaluation IS excellent AND Termwork_Evaluation IS good THEN Performance IS Good;

RULE 18: IF Attendance IS good AND Exam_Evaluation IS excellent AND Termwork_Evaluation IS excellent THEN Performance IS Excellent;

RULE 19: IF Attendance IS excellent AND Exam_Evaluation IS poor AND Termwork_Evaluation IS poor THEN Performance IS Poor;

RULE 20: IF Attendance IS excellent AND Exam_Evaluation IS poor AND Termwork_Evaluation IS good THEN Performance IS Good;

RULE 21: IF Attendance IS excellent AND Exam_Evaluation IS poor AND Termwork_Evaluation IS excellent THEN Performance IS Good;
RULE 22: IF Attendance IS excellent AND Exam_Evaluation IS good AND Termwork_Evaluation IS poor THEN Performance IS Good;

RULE 23: IF Attendance IS excellent AND Exam_Evaluation IS good AND Termwork_Evaluation IS good THEN Performance IS Good;

RULE 24: IF Attendance IS excellent AND Exam_Evaluation IS good AND Termwork_Evaluation IS excellent THEN Performance IS Excellent;

RULE 25: IF Attendance IS excellent AND Exam_Evaluation IS excellent AND Termwork_Evaluation IS poor THEN Performance IS Good;

RULE 26: IF Attendance IS excellent AND Exam_Evaluation IS excellent AND Termwork_Evaluation IS good THEN Performance IS Excellent;

RULE 27: IF Attendance IS excellent AND Exam_Evaluation IS excellent AND Termwork_Evaluation IS excellent THEN Performance IS Excellent;

**Inference Method:** An output fuzzy set is obtained from the input variables and rules by applying the inference procedure. In case of several rules are active for the same output membership function, it is necessary that only one membership value is chosen. This process is entitled as ‘fuzzy inference’ or ‘fuzzy decision’. Several authors like Mamdami, Takagi Sugeno and Zadeh have developed a range of techniques for fuzzy decision making and fuzzy inference [7]. In order to develop an experimental system, we have used the Mamdami’s max-min inference method as it is typically used in modeling human expert knowledge and it is given below [8]:

\[
\mu_c(y) = \max_k \left[ \min \{ \mu_A(input(i), \mu_B(input(j)) \} \right], k = 1, 2, 3, 4... r.
\]

This expression determines an output membership function value for each active rule. When one rule is active, an AND operation is applied between input values. The smaller input value is chosen and its membership value is determined as the membership value of the output of that rule. This method is repeated to determine output membership functions for each rule. To sum up, graphically AND (min) operation is applied between inputs and OR (max) operations are between output.
Defuzzification of Performance Value

Defuzzification is the process of converting a fuzzy number into a crisp value. It is the process of producing a quantifiable result in fuzzy logic based on given fuzzy sets and corresponding membership degrees. Defuzzification is interpreting the membership degrees of the fuzzy sets into a specific decision or real value. After completing the fuzzy decision process, the obtained fuzzy number must be converted to a crisp value. This process is known as defuzzification. Several methods have been developed so far for defuzzification. In our experimental system, we have used Center of Gravity (CoG) Method (also known as CoA – Center of Area) for defuzzification. An example of the center of gravity method is given below [3].

5.3.4 Advantages and Outcomes of an Experimental System

To demonstrate the applicability and usability of the generic framework for integration of multi-agent knowledge-based system and distribute database grid, we have implemented an experimental system for students’ performance evaluation. The following are some of the advantages and outcomes of the students’ performance evaluation.

- The system evaluates the performance of the students at regular interval, generates the appropriate reports for the same and makes these reports available to the students.
- The reports are generated on regular interval as a part of a continuous evaluation process and provide a chance for the students to increase their performance rate before the final exams are conducted.
- The university may accomplish the assessment of their students’ performance in a timely and user friendly manner. Also, the performance may disaggregate by
institutes, instructor, gender, ethnicity, economic status and disability; have a meaningful impact on the achievement of students throughout their career.

- The generated performance reports provide a means for monitoring of each institute’s progress individually that leads towards progress of an entire university.

- To store the data of students, the different institutes may use different storage systems and data formats with differences in their schema, access rights, metadata attributes and many more. Therefore, the system provides a single and a virtualized view of such institutional data assets and allows data providers of institutes to retain control of their data.

- As the experimental system operates in a data grid environment, new institution may add as a new database grid node in the system without much modification in code. Also, it offers scalability and extensibility and implemented a strategy used to avoid a single-point-of-failure.

- With the availability and accessibility of individual student performance evaluation reports, the staff and faculty members may uncover answers to questions about the way students learn so that they can form the strategies to increase student rate of learning.

- The performance reports may provide to the parents to make them aware about the status of student’s regularity and academic performance. So that, they may identify the root cause at an early stage before it affects negatively on student’s future performance.

- The system is implemented as a web-based system so that a user can access it from any location with appropriate privileges.

Further, in the future, the developed experimental system can be integrated with other systems or modules within the domain.

- **Advantages of Applying Fuzzy Set Theory and Fuzzy Logic**

  - Linguistic values used to make it simpler to way human think.
  - Allows the solution to previously unsolved problems.
  - Rapid prototyping is possible as knowledge is not required before starting phase of development.
  - Cheaper to make than conventional system as easier to design.
  - Increased robustness.
Simpler knowledge acquisition and representation.

A few rules are used to describe great complexity.

5.3.5 An Example Session

In this section, we have demonstrated the sample screens and code snippets of an experimental system. As we have mentioned earlier, we have implemented an experimental system with OGSA-DAI, JAVA EE (JAX-WS) and jFuzzyLogic. Currently, there are two users of the system: Administrator and Student. An administrator is a user who has all privileges to access the system. The student is a user who has limited privileges to access the system.

We have provided the generic framework for integration of multi-agent knowledge-based system in the chapter 3. Also, we have provided the computational model for the generic framework in the chapter 4. Based on that, we have provided an architectural view of students’ performance evaluation in the section 5.3.1 of the same chapter. The implemented experimental system operates in a data grid environment and consists of multiple collaborative agents used to provide data grid services, application specific services and knowledge-based services. Among of them, the data grid services like resource management, data access & integration and grid administrative are already demonstrated with related screen layouts and code snippets in the previous chapter 4, section 4.2 (Implementation Scenario). Therefore, in this example session, our focus is to provide screen layouts and code snippets for application specific services and knowledge-based service (i.e. through fuzzy interface agent).

An Example Session at Student Side

1. A Screen Layout to Log in the System: It provides a visual interface to all system users’ to log into the system in order to use the services offered by the system. This screen is used at both the side: User side and Administrator Side. Figure 5.11 demonstrates the screen layout to log into the system. The user enters a set of username and password. It will be validated by user management agent. Upon valid username and password, a student can access his or her profile.
2. **A Screen Layout to Change Password:** Once the user is successfully logged in the system, he can change password by navigating to the following screen. User management agent provides the service to change the password. Here, he has to enter his old password and provide the new password. Figure 5.12 demonstrates the screen layout to change password.

3. **A Screen Layout to View Basic Details of Student:** Once the student successfully logged in, he can see his basic profile which includes his id, name, address etc. by visiting the following screen. It contains the student’s basic information and also provides the facility to edit the basic information of the student. Student profile agent
provides these details to students. The data are provided on a real time basis by a distributed database grid node owned by specific institution where a student is enrolled and his data are stored. Also, updated records automatically insert into appropriate distributed database grid node through the system. Data access & integration agent is responsible to retrieve the above data from heterogeneous and distributed database grid. Figure 5.13 demonstrates the screen layout to view basic details of the student.

4. **A Screen Layout to View Course Announcements**: This is the added functionality to view the recent course announcement. The data shows here are originating from respective distributed database node. Figure 5.14 demonstrates the recent course announcement.

5. **A Screen Layout to View Course Details**: This screen shows the detailed academic profile of the student like his current semester, course, course fees, subject list etc. Also, it provides the backlog carriage information. For that, it accesses and integrates the data from at least two distributed database nodes: one is to get the student’s course
details and other is to get the student’s exam result of last semester. Here, in experimental system, the last semester result resides in exam database node and it may use a different database management system to store the data. (In a typical university domain, generally exam section provides the limited access of such type of data). So, to provide such details, the student profile agent needs to access and integrates the data from distributed heterogeneous database nodes and provides a uniform interface to users for viewing the details. For that, it has to cooperate with data access & integration agent. Figure 5.15 demonstrates the screen layout to view course details of the student.

![Figure 5.15: A Screen Layout to View Course Details](image)

6. A Screen Layout to View Performance Reports: As we have discussed in earlier section also, there are three parameters considered in order to evaluate the student’s performance. In our experimental system, these data are provided by specific
institution which is registered as a database grid node in the system. The data access & integration agent and progress monitoring agent are responsible to provide the said three parameters to the path advising agent. The path advising agent applies fuzzy set theory and logic discussed in earlier section to evaluate the student’s performance. Here, the following screen layout shows the performance of the student based on the reports generated on regular interval by authorized user i.e. administrator or faculty. When a student clicks on ‘Details’ button, it shows the detailed report.

Apart from this, the agent also compares the last two performances of the student and generates the advice based on that. Figure 5.16 demonstrates the screen layout to view performance reports of the student. It also shows the generated advice based on last two performances.

As soon as the new reports will be generated by authorized users like administrator or faculty, the list of reports will be updated as shown in Figure 5.17.
Figure 5.17: A Screen Layout to View Updated List of Performance Reports

7. A Screen Layout to View Detailed Performance Report with Fuzzy Graphs: This screen layout shows the detailed report of the student’s performance. It shows the graphs generated during the fuzzification & the defuzzification process for input and output parameters. The graph also shows the degree of membership of these parameters. Figures 5.18 and 5.19 demonstrate the two detailed reports of student’s performance. For each report, the same types of detailed reports are generated through an agent.
## Chapter 5: Implementation of Experimental System

### Performance of Abhlish between 2012-07-02 and 2012-07-08

<table>
<thead>
<tr>
<th>Subject</th>
<th>Total Days</th>
<th>Present Days</th>
<th>Percentage Attendance</th>
<th>Total Marks</th>
<th>Marks Obtained</th>
<th>Percentage Exam</th>
<th>Total Marks</th>
<th>Marks Obtained</th>
<th>Percentage Term-work</th>
</tr>
</thead>
<tbody>
<tr>
<td>MB201</td>
<td>4</td>
<td>3</td>
<td>75.00</td>
<td>15</td>
<td>10</td>
<td>66.66</td>
<td>5</td>
<td>3</td>
<td>60.0</td>
</tr>
<tr>
<td>MB202</td>
<td>6</td>
<td>3</td>
<td>50.00</td>
<td>15</td>
<td>12</td>
<td>80.0</td>
<td>5</td>
<td>2</td>
<td>40.0</td>
</tr>
<tr>
<td>MB203</td>
<td>0</td>
<td>2</td>
<td>40.00</td>
<td>15</td>
<td>13</td>
<td>86.66</td>
<td>5</td>
<td>5</td>
<td>100.0</td>
</tr>
<tr>
<td>MB204</td>
<td>5</td>
<td>1</td>
<td>20.00</td>
<td>15</td>
<td>11</td>
<td>73.33</td>
<td>5</td>
<td>3</td>
<td>60.0</td>
</tr>
<tr>
<td>MB205</td>
<td>0</td>
<td>0</td>
<td>100.00</td>
<td>15</td>
<td>10</td>
<td>66.66</td>
<td>5</td>
<td>4</td>
<td>80.0</td>
</tr>
</tbody>
</table>

Attendance is 5.6, Exam Evaluation is 7.47, Term-work Evaluation is 6.8 and Performance is 8.51.

![Performance Graph](image1)

**Performance:**
- Membership function for performance levels:
  - VeryPoor
  - Poor
  - Average
  - Good
  - Excellent

**Performance:**8.51 (CenterOfGravity)

- Membership function for performance levels:
  - VeryPoor
  - Poor
  - Average
  - Good
  - Excellent

**Attendance**

- Membership function for attendance levels:
  - VeryPoor
  - Poor
  - Average
  - Good

**ExamEvaluation**

- Membership function for exam evaluation levels:
  - VeryPoor
  - Poor
  - Average
  - Good

**TermworkEvaluation**

- Membership function for term work evaluation levels:
  - VeryPoor
  - Poor
  - Average
  - Good

---

**Figure 5.18:** A Screen Layout to View Detailed Performance Report with Fuzzy Graphs - Case 1
### Chapter 5: Implementation of Experimental System

#### Development of Multi-agent Knowledge-based System Accessing Distributed Database Grid

**Figure 5.19: A Screen Layout to View Detailed Performance Report with Fuzzy Graphs - Case 2**

<table>
<thead>
<tr>
<th>Subject</th>
<th>Total Days</th>
<th>Present Days</th>
<th>Percentage Attendance</th>
<th>Total Marks</th>
<th>Marks Obtained</th>
<th>Percentage Exam</th>
<th>Total Marks</th>
<th>Marks Obtained</th>
<th>Percentage Term-work</th>
</tr>
</thead>
<tbody>
<tr>
<td>MB201</td>
<td>4</td>
<td>3</td>
<td>75.00</td>
<td>15</td>
<td>6</td>
<td>40.0</td>
<td>10</td>
<td>6</td>
<td>60.0</td>
</tr>
<tr>
<td>MB202</td>
<td>6</td>
<td>4</td>
<td>66.66</td>
<td>15</td>
<td>8</td>
<td>53.33</td>
<td>10</td>
<td>7</td>
<td>70.0</td>
</tr>
<tr>
<td>MB203</td>
<td>5</td>
<td>3</td>
<td>60.00</td>
<td>15</td>
<td>6</td>
<td>40.0</td>
<td>10</td>
<td>8</td>
<td>80.0</td>
</tr>
<tr>
<td>MB204</td>
<td>5</td>
<td>3</td>
<td>60.00</td>
<td>15</td>
<td>7</td>
<td>46.66</td>
<td>10</td>
<td>7</td>
<td>70.0</td>
</tr>
<tr>
<td>MB205</td>
<td>6</td>
<td>4</td>
<td>80.00</td>
<td>15</td>
<td>5</td>
<td>33.33</td>
<td>10</td>
<td>5</td>
<td>50.0</td>
</tr>
</tbody>
</table>
An Example Session at Administrator Side

The screen layouts for Resource and Query Browser are already demonstrated in the chapter 4 in section 4.2 (Implementation Scenario). In this section, we provide the remaining screen layouts.

1. A Screen Layout after successfully Logged in the System: The following screen provides the access mechanism for main services offered by the system to a privileged user i.e. admin. Figure 5.20 demonstrates the screen layout after successfully logged in the system as an admin.

![Figure 5.20: A Screen Layout after Successfully Logged in the System](image)

2. A Screen Layout for Create Users: Admin is the privileged user for creating new users in the system. Figure 5.21 demonstrates the visual interface to create new users in the system. A user management agent is responsible for creating and maintaining the users of the system.
3. *A Screen Layout to Generate Report:* As we can see in the Figures 5.17 and 5.18, a student can view his performance report and also get the advice about his performance track. To provide such reports, admin is the privileged user who generates these performance reports for the student by accessing the services offered by multiple collaborative agents. A system provides different options to generate the report like institute wise, course wise or semester wise as we can see in Figure 5.22. Admin can generate the report simultaneously for all the students.

As we mentioned earlier, in university, the data of the students reside and manage by individual node. The location and database type used may also vary. For that, some mechanism need to be established to access the data from such heterogeneous and geographically distributed database nodes. Our system accomplishes this task by implementing the grid-based model for integration of geographically distributed & heterogeneous educational resources for knowledge extraction & delivery. We can see from the Figure 5.22 that, an administrator is a privileged user who generates such reports between two given dates. To accomplish the above task, in the experimental system, we have implemented data access & integration agent, progress monitoring agent and path advising agent and they need to cooperate with each other to achieve the goal. The generated reports can further viewed by student as shown in the Figures 5.17 and 5.18. The Figure 5.22 demonstrates the visual interface to generate the performance reports.
4. *A Screen Layout to Show Report*: An administrator can also view the report about the performance of the students. There are various options available for the same as we can see in Figure 5.23. After selecting specific options, a report is generated by a report generator agent as demonstrated in Figure 5.24. Currently, the report generator agent implements one type of report i.e. pie chart by collaborating with other agents remain in the system. In future, the agent can be extended by adding the capability to generate more types of charts such as bar chart, line chart etc. As we can see in the Figure 5.24, the data comes from various heterogeneous and distributed database nodes on a real time basis. Also, after integrating such data, fuzzy set theory and fuzzy logic are applied for knowledge extraction and delivery.
5. A Screen Layout for Search Option: An administrator can perform various search operations by using search agent. Currently, a search agent provides the functionality
to search for the students who are having specific performance. Figure 5.25 and Figure 5.26 demonstrate the search functionality. We can see that, the search agent simultaneously fetches the data from heterogeneous and distribute database grid to generate the result. For that, it has to cooperate with data access & integration agent. In future, the search agent can be extended by adding more searching functionalities and searching methodologies.

![Figure 5.25: A Screen Layout for Search Option](image1)

![Figure 5.26: A Screen Layout Showing Search Result](image2)
Code Snippet to Implement Fuzzy Logic Based Performance Evaluation

As we have mentioned earlier also, we have used the jFuzzyLogic Library to implement the fuzzy control logic to evaluate students’ performance [3]. For that, we have to first develop the .fcl file which contains the code to be executed by the path advising agent. The following is the code snippet of the above mentioned .fcl file.

```csharp
// Block definition (there may be more than one block per file)
FUNCTION_BLOCK test1

// Define input variables
VAR_INPUT
    Attendance : REAL;
    Exam_Evaluation : REAL;
    Termwork_Evaluation: REAL;
END_VAR

// Define output variable
VAR_OUTPUT
    Performance : REAL;
END_VAR

// Fuzzify input variable 'Attendance'
FUZZIFY Attendance
    TERM poor := (0, 1) (3,1) (5, 0);
    TERM good := (3, 0) (5,1) (7,0);
    TERM excellent := (5, 0) (7, 1) (10,1);
END_FUZZIFY

FUZZIFY Exam_Evaluation
    TERM poor := (0, 1) (3,1) (5, 0);
    TERM good := (3, 0) (5,1) (7,0);
    TERM excellent := (5, 0) (7, 1) (10,1);
END_FUZZIFY

FUZZIFY Termwork_Evaluation
    TERM poor := (0, 1) (3,1) (5, 0);
    TERM good := (3, 0) (5,1) (7,0);
    TERM excellent := (5, 0) (7, 1) (10,1);
END_FUZZIFY

// Defuzzify output variable 'Performance'
DEFUZZIFY Performance
    TERM Excellent := (7,0) (9,1) (10,1);
    TERM Good := (5,0) (7,1) (9,0);
    TERM Average := (3,0) (5,1) (7,0);
    TERM Poor := (1,0) (3,1) (5,0);
    TERM VeryPoor := (0,1) (1,1) (3,0);
    // Use 'Center Of Gravity' defuzzification method
    METHOD : COG;
    // Default value is 0 (if no rule activates defuzzifier)
    DEFAULT := 0;
END_DEFUZZIFY

RULEBLOCK No1
    // Use 'min' for 'and' (also implicit use 'max'
    // for 'or' to fulfill DeMorgan's Law)
    AND : MIN;
```
// Use 'min' activation method
ACT : MIN;
// Use 'max' accumulation method
ACCU : MAX;

// Rule 1 to Rule 27... already mentioned in the above section
END_RULEBLOCK
END_FUNCTION_BLOCK

5.4 Conclusion

This chapter introduces implementation details of an experimental system. The grid-based model has been developed as a case study. The said model is used for integration of geographically distributed & heterogeneous educational resources reside in the typical university domain for knowledge extraction & delivery. Apart from this, we have also demonstrated an experimental system (students’ performance evaluation) implemented based on the generic framework and grid-based model. In an experimental system, various multiple collaborative agents are working together. Some of them are resource management agent, data access & integration agent, user management agent, path advising agent and so forth. This chapter includes various implementation aspects of an experimental system in detail. For that, essential screen layouts and code snippets have been provided. The advantages and outcomes of the experimental system have also been explained. The next chapter will discuss about the results, outcomes and future extension of the research work.

References


