Chapter 1

Introduction and Overview

This chapter provides an introduction and overview of the research work. It discusses the background of the study and presents the purpose and significance of the study. It also discusses the approach used to perform the research work. The objective of the research work has been discussed in detail and the roadmap of the thesis has also been provided. Moreover, it discusses the fundamental concepts regarding data grid, agent and multi-agent system, knowledge-based system and fuzzy logic & fuzzy set theory.

Section 1.1 of the chapter focuses on the research problem and specifies objective of the research work. Section 1.2 discusses the fundamental concepts in detail. Section 1.2.1 gives an introduction to grid and grid computing and provides the classifications of the grid. Section 1.2.2 discusses the various aspects of agent and multi-agent systems like the definition and characteristics of agent and multi-agent systems, agent application domains and advantages of multi-agent systems. Section 1.2.3 explains the fundamental concepts of knowledge-based systems and presents the knowledge-based system architecture. Section 1.2.4 discusses the basics of fuzzy logic, fuzzy sets, fuzzy membership functions, linguistic variables and hedges. It also provides properties of fuzzy sets.

1.1 Research Problem

1.1.1 Background of the Study

Data plays a crucial role in all types of cross-organizational research and applications. The grid is an effective infrastructure for the coordinated use and sharing of distributed resources in a dynamic manner [13], enabling the temporary pooling of resources to solve specific problems [5]. It does not refer only to resources such as Central Processing Unit (CPU) power, storage amenities and memory, but also to data sources. Data grids rely on the coordinated sharing of and interaction across multiple autonomous database management systems. Data grids provide transparent and single point access to heterogeneous and autonomous data resources stored in grid nodes. A data grid can include and provide transparent access to semantically related data resources. These data
resources are maintained in different syntaxes and managed by different software systems. Also, they are accessible through different protocols and interfaces [6].

Early grid applications focused principally on the movement of file-based data storage and replication. Many grid applications have already used databases for managing metadata and increasingly many are associated with large databases of domain-specific information (e.g. biological or astronomical data). One of the aims of the data grid is to promote systematic sharing of scientific and academic data. According to Pearson [7], “The prospect exists for literally billions of data resources, from large databases to individual collections, from major instruments such as the Large Hadron Collider and telescopes to portable instruments, medical scanners, networks of environmental sensors and wearable devices and petabytes of data being accessible and available in a grid environment”. In such scenario, data are generated from several resources which may include sensors, special instruments and wearable devices. If this prospect is realized, it is expected that many of the advances to flow from the grid will come from applications that can combine information from multiple data sets.

Multi-agent systems have received much attention in recent years because of their usability in complex and distributed environments. The characteristics of the agents offer several functionalities which can help to discover a seamless system across distributed, heterogeneous, dynamic, open organization that the user can access from any location. Through realization of agents, users are provided agent-based interface which is used to connect end users or external systems together with the multi-agent system in a knowledge-based fashion. Knowledge-based system is used to provide intelligent decisions with justification. It is a system that draws upon the knowledge of human experts captured in a knowledge base to solve problems that normally require human expertise. Knowledge-based system can be implemented with the concept of an expert system. The expert system technology can be implemented by incorporating fuzzy logic to handle qualitative and uncertain facts in the decision making process. Fuzzy set theory and rules are used to achieve the advantages of uncertainty along with other advantages of knowledge-based systems. Fuzzy logic is a superset of conventional (Boolean) logic that has been extended to handle the concept of a partial truth i.e. Truth values between entirely true and entirely false [28]. Fuzzy sets and logic are used to represent uncertainty, which is critical for the management of real world systems. We may numerically weighted
the linguistic terms like excellent, very good, good, average, poor or bad by assigning priorities to these qualitative facts through the realization of fuzzy logic [2].

Moreover, the grid community focuses on "brawn": interoperable tools and infrastructure for secure and reliable resource sharing within dynamic and geographically distributed Virtual Organizations (VOs) [12, 13] and applications of the same to various resource federation scenarios. In contrast, agent community focuses on "brains": on the development of theories, methods, and algorithms for autonomous problem solvers that can act flexibly in uncertain and dynamic environments in order to achieve their aims and objectives [21]. Moreover, grid and agent deployments increase by the time and therefore agent systems requiring robust infrastructure and grid systems requiring autonomous and flexible behaviors [15]. Therefore, this research work is aimed to provide an integrated approach to leverage the power of both grid and agent technologies in knowledge-based fashion.

1.1.2 Purpose and Significance of the Study

Emerging modern e-science and e-business applications require vast amounts of data which are being stored in many different formats by using different storage solutions. Currently, these applications are facing a common challenge that is managing a distributed data explosion. Devices such as Medical Imaging Instruments, Detectors, Multi-sensor Instruments, Micro-arrays and many more are producing amounts of data that are rapidly exceeding the capacities of their current local data storage and computing environments. In many cases, data are distributed from the point of beginning and being formed by different research groups. The examples of such scenarios are genome and protein analysis data produced by many research laboratories in the world, biological databases containing patient data from a variety of hospitals and many more. Moreover, experiments in radio-astronomy or in particle physics fields are generating terabytes and petabytes of data per year which need more data processing power than ever can be located on a single site. These data are utilized by researchers all over the world [6]. This will allow researchers to combine different types of information on a single entity to gain a more thorough picture and to aggregate the similar types of information about different entities [29].

In the research community, much of modern research involves analyzing data arising from experiments and simulations held in existing repositories often owned by different organizations and partners. With online storage and provisioning of data, sharing data has
become a crucial part of modern research in all fields. Currently, the increasing social demands of high quality data resources to process scientific and business applications cannot be fulfilled only by the available data resources which generally reside in the individual administrative domain. This has created a demand for solutions which offers the data access and integration. Thus data plays a fundamental role in such kinds of cross-organizational research and alliances, these organizations can be jointly deemed to form virtual organizations (VOs). Size of data, diversity of data sources and data freshness requirements makes data integration a complex task and this data may be geographically distributed. It is impossible to handle such large and diverse amounts of data within a single organization. Early grid applications focused principally on the storage, movement and replication of file-based data. Therefore, the need for the integration of database technologies with grid middleware is widely recognized.

As data, can be distributed across multiple geographical locations and use different storage formats and access mechanisms, a solution to facilitate data access, integration and sharing would be advantageous. The architecture of many scientific and business applications operate in heterogeneous and dynamic information environment can be modeled by integrating the concept of an agent. An agent is a piece of software program which acts proactively on behalf of the user [37]. Agents can be used in complex systems where expertise and resources are distributed like database grid. The agent paradigm is successfully employed in those applications where autonomous, heterogeneous, distributed and loosely-coupled resources and systems need to cooperate in order to achieve a common goal. They use the modular approach and provide the reusability. In open distributed systems, independent components cooperate to achieve individual and shared goals. Both, systems and individual components as a whole, are designed to cope with change and evolution in the number and nature of the participating entities. Such systems are important for large scientific collaborations, enterprise systems and sensor networks [15]. An agent-based framework allows an application to be modeled, designed and developed as information environments being composed of different co-operating agents.

Apart from this, organizations generally strive for the data for analysis and decision making purposes. In any organization, this heterogeneous and geographically separated data are required on a real time basis for analysis and decision making. Fuzzy logic based expert system is used to enhance the performance and reliability in the decision making
process. As data are heterogeneous and distributed, knowledge extraction, presentation and delivery from heterogeneous data sources available in the distributed database grid environment are challenging research and development issues. In the research carried out, the aim was to develop the fuzzy expert system modeled by incorporating agents on the top of the distributed database grid. It is also aimed to accommodate the knowledge and expertise with reasoning capabilities that will provide a great support to executives for decision making in any research and commercial applications.

1.1.3 Approach of the Study

The objective of the research work is to develop a framework of multi-agent knowledge-based system in order to access heterogeneous and geographically distributed databases on a data grid. In the research carried out, the integrated generic framework of multi-agent knowledge-based system accessing distributed database grid is developed. The framework hides the heterogeneity and provides transparent and single point access of heterogeneous and geographically distributed databases to client applications and users and handles the data access and integration through multiple agents.

The framework also shows how different heterogeneous data resources are integrated via grid middleware and how knowledge extraction, delivery and management are performed within the distributed data grid environment by using multiple task agents. The framework consists of four layers: Grid Fabric Layer, Middleware Layer, Knowledge Layer and Application Layer for effective data access and integration.

The Grid Fabric Layer works as a data layer which is consisting of various relational heterogeneous database resources and network resources used by the data grid. Middleware Layer is a key component of the generic framework. It mainly provides the core services, user level services and data services. Through middleware services, client applications and users are enabled to access and manage various heterogeneous databases through secure and transparent grid system and implement the grid interoperability characteristics. They also hide the underlying heterogeneity and dynamics of those databases. It also performs several configurations, security and workflow activities thorough integration of various task agents which are working within a multi-agent environment. The knowledge layer of the generic framework consists of knowledge-based component through which content is turned into something more than just a collection of data. It contains a specific agent through which knowledge base component is
implemented. A knowledge-based system is having two essential components: knowledge base and inference engine. The knowledge base is a repository of domain knowledge and meta knowledge. An inference engine is a software program which infers the knowledge available in the knowledge base. The advantages of knowledge-based systems may realized by incorporating fuzzy interface. Fuzzy logic representations are more intuitively satisfying than classical Boolean logic as well as it is more precise and compact compared to classical rule based representations. Fuzzy Logic Controller (FLC) is being widely and successfully applied in different areas. FLC can be considered as a knowledge-based system incorporating human knowledge in their knowledge base through fuzzy rules and fuzzy membership functions. The application layer is the highest layer of generic framework and calls on all other layers for resources where and when required. The application layer directly accesses the knowledge layer or the use of the data services offered by middleware layer through an interface facilitator agent.

Earlier grid applications focused principally on the storage, movement and replication of file-based data. However, nowadays many of them are associated with large databases of domain-specific information. Several grid middleware technologies are available to realize data grid. We have used OGSA-DAI (Open Grid Services Architecture - Data Access and Integration) as a grid middleware. OGSA-DAI allows data resources (e.g. relational, XML databases and files) to be federated and accessed via web services on the web or within grids or clouds. To implement agents and multi-agent systems, we have used web services. Data can be queried, transformed, updated and combined in various ways through these services.

1.1.4 Objective of the Research

The objective of the research work is to develop a framework of multi-agent knowledge-based system accessing heterogeneous and geographically distributed databases on a data grid. Data grid is a grid computing system that deals with large amounts of controlled, shared and managed distributed data. A software agent that exhibits some form of artificial intelligence assists users and acts on their behalf in performing various tasks. The research aim is to develop a model for integration of data grid technology with agent technology. The said model can be useful to develop applications in research, education and business fields. This framework exploits facilities of the grid and the agent to access, integrate and analyze data and/or information collected from all over distributed database grid.
environment. Apart from this, to demonstrate the research work, an experimental system has also been designed and developed.

The agents of the generic framework are used to expose heterogeneous data resources from and to a data grid via grid middleware. OGSA-DAI (Open Grid Services Architecture - Data Access and Integration) is a grid middleware provides a solution which makes it easy to publish and share data across organizational boundaries, and develop applications which use both public and personal data resources, through a secure, extensible framework based on web service standards. The designed framework provides an extensible interface through which a client application can access the data stored in geographically distributed relational databases without being aware of their physical locations. Following are the major objectives of the research work:

- To design a framework that hides the heterogeneity and provides transparent and single point access of heterogeneous and geographically distributed databases to users and client applications and handles the data access and integration through multiple agents;
- Development of a standardized interface which allows data resources, such as relational, XML databases and file-based data, to be federated, accessed and integrated across the Internet via OGSA-DAI (Open Grid Services Architecture - Data Access and Integration);
- Development of an agent based interface to the client applications to access the middleware in a platform and language independent way by integrating agent and multi-agent system within the data grid;
- To design a web-based interface to add and manage the database nodes into the data grid;
- To automate the process of configuration and deployment of database resources in OGSA-DAI (a grid middleware) environment;
- Multiple database federation and metadata management through a grid middleware;
- To hide the heterogeneity of the database type, database drivers and database locations for client applications;
- Implementation of the agents which exposes heterogeneous data resources from and to a grid via OGSA-DAI middleware;
• To access the underlying data grid into knowledge-based fashion using fuzzy set theory & rules and implementation of a fuzzy interface; and
• Documentation of knowledge for future use and extension.

1.1.5 Road Map of the Thesis

Chapter 2 of the thesis provides details of literature survey conducted for research work. This chapter also includes details about the work accomplished in this area, study of related work, issues and challenges, techniques, methodologies and so forth. Chapter 3 presents the developed integrated generic framework of multi-agent knowledge-based system accessing distributed database grid. It specifies the architecture of generic framework and its components, functionalities like standardized interface, metadata management, multiple database federation, role-based security etc., organization of multiple agents worked upon it and the integration and the importance of knowledge-based component through a fuzzy interface in detail. This chapter also describes various components and activities involved in each layer of the generic framework.

Chapter 4 provides the detailed methodology for development of the framework mentioned in the chapter 3. It covers detailed aspects of implementation of OGSA-DAI (Open Grid Services Architecture - Data Access and Integration) and integration of the same with multi-agent system. It shows the sample code snippets and screen layouts necessary to explain the methodology used to implement the framework. Chapter 5 introduces the implementation details of the design and development phases of an experimental system. The experimental system implemented based upon the framework and methodologies described in the above chapters. It includes the implementation details of the experimental system like entities of the system, integration of databases into database grid, classes and packages to implement agents, implementation of fuzzy interface and so forth. Chapter 6 discusses the results, conclusion and future extension of the research work. It evaluates the outcomes and benefits of the developed generic framework.
1.2 Fundamental Concepts

1.2.1 Fundamental Concepts of Grid

a) Introduction to Grid

Grid computing is gaining a lot of attention within the IT industry. Initially, it was used within the academic and scientific community, but later on standards, enabling technologies, products and toolkits become available that allow businesses to use and reap the advantages of grid computing. Even though "The Grid" is still just a vision, grid computing is already a reality. Many people believe that "The Grid" will be connections created between several million computers from all over the world, which may include supercomputers, data vaults, desktops, laptops, instruments like mobile phones, sensors and telescopes, and owned by thousands of different people to form a single, huge and powerful computer. Grid computing is the federation of computer resources from multiple administrative domains to reach a common goal. The grid can be perceived as a distributed system with non-interactive workloads that involve a large number of files. Grid computing is distinguished from conventional high performance computing systems such as cluster computing as grids tend to be more loosely coupled, geographically scattered and heterogeneous [39].

"The Grid" takes its name from an analogy with the electrical "Power Grid". Accessing computation power from a computer grid would be as simple as accessing electrical power from an electrical power grid. When one's plugs an electronic appliance requiring electrical power into a receptacle, one expects that power of the correct voltage is available. They need not to know the actual source from where power is coming. The local utility company provides the interface into a complex network of generators and power sources and provides an acceptable quality of service for energy demands. The power grid infrastructure provides a virtual generator and thus eliminating the need for each house or neighborhood having to obtain and maintain its own generator of electricity. The generator is highly reliable and adapts to the power needs of the consumers based on their demand [3].

The function of grid computing is similar to this. Once the proper kind of infrastructure is established, a user accesses a virtual computer that is reliable and adaptable to the user's requirements. This virtual computer consists of many diverse computing resources. But
these individual resources are not visible to the user, as the user of electric power is unaware of how their electricity is being generated. Table 1.1 depicts features and similarities between “The Grid” and conventional “Electric Power Grid”.

<table>
<thead>
<tr>
<th>Features</th>
<th>Similarities</th>
</tr>
</thead>
<tbody>
<tr>
<td>Availability and Origin</td>
<td>In both, Electric Power Grid and The Grid, users are unaware of the origin of where the power (electric power or computer power) is coming to their place. They only know about availability of the power to use.</td>
</tr>
<tr>
<td>Accessing Resources</td>
<td>Electric Power Grid links together various transmission stations, power stations, transformers, power lines and more. Similarly, The Grid links together various computing resources like servers, PCs, workstations, storage elements and more. Both provide the mechanism needed to access them.</td>
</tr>
<tr>
<td>Utility Services</td>
<td>Electric Power Grid and The Grid, both provide utility services. Users need to pay according to their usage.</td>
</tr>
</tbody>
</table>

*Table 1.1: Features and Similarities between “The Grid” and Conventional “Electric Power Grid”*

**b) Grid Computing**

Grid computing provides the infrastructure and technology to explore new ways of doing scientific research. It provides a way to share data, data storage space, computing power, results and more. Researchers can tackle bigger questions than ever before than span across different areas like from disease cures and disaster management to global warming and the mysteries of the universe. The grid can be used by several types of users according to their needs. Initially, scientists use grid computing for their research. But, nowadays, usage of grid computing has spanned across different domains. Some of the examples are as follows.

In bio-science, biologists are using grids to simulate thousands of molecular drug candidates on their computer. Their aim is to find a molecule, which is able to block specific disease proteins. In an area of geological science, earth scientists are downloading hundreds of gigabytes of data every day using grids to track ozone levels via satellites. In the area of physical science, high energy physicists are using grids in their search for a
better understanding of the universe. Each year, the Large Hadron Collider is producing 10 petabytes of data to store and analyze which are relying on a grid of thousands of desktops. Thousands of physicists in dozens of universities around the world want to analyze this data.

Engineers are using grids to study alternative fuels, such as fusion energy. Artists use grids to create complex animations for feature films. Social scientists use grids to study the social life of a variety of species, the makeup of mankind, the secrets of history and so forth. Education involves administrators, parents, mentors, students and so, is a very natural application of grid technologies. Teachers and educators can use e-libraries and e-learning centers which are benefiting from grid-based tools for accessing distributed data and creating virtual classrooms with distributed students, resources and tutors.

Global enterprises and large corporations have people, sites, data and resources distributed all over the world. Grids will allow such organizations to carry out large-scale modelling or computing by simultaneously using the resources at their many sites. In field of medical science, several patient databases, drug discovery databases, medical image archives and specialized instruments such as MRI machines, CAT scanners etc. can be connected through a grid to enhance diagnosis procedures [39].

Grid computing means different things to different individuals. There are many definitions exist for grid computing. However, the definitions and terminologies given by Ian Foster and Kesselman are highly acceptable. According to them, grid computing is a growing technology that facilitates the executions of large-scale resource intensive applications on geographically distributed computing resources. It facilitates secure, flexible, large-scale resource sharing among dynamic collections of individuals, institutions, and resource. It also enables communities (generally called “virtual organizations”) to share geographically distributed resources as they pursue common goals [12].

Grid computing can be seen as a journey along a path of integrating various technologies cover and solutions that move us closer to the final goal. It provides distributed computing infrastructure technologies which are evolving in support of cross-organizational application and resource sharing, virtualization across technologies, platforms, and organizations. This kind of virtualization is only achievable through the use of open standards. Such open standards help to ensure that applications can transparently take advantages of whatever appropriate resources can be made accessible to them.
environment that offers the ability to share and transparently access resources across a
distributed and heterogeneous environment requires the technology to virtualize certain
resources. It also requires technologies and standards in the areas of systems management,
scheduling, accounting, security and so forth [4]. Standards such as the OGSA (Open Grid
Services Architecture) and products based on OGSA provide the necessary framework.
According to Ian Foster and Kesselman, the following criteria should be followed to form
a grid.

- Coordinates resources which are not under centralized control;
- Uses standard, open, general-purpose protocols and interfaces; and
- Provides significant qualities of service.

Also, it should offers following benefits.

- Exploits underutilized resources;
- Resource balancing;
- Virtualize resources across an enterprise like data grids, compute grids; and
- Enables collaboration for virtual organizations.

There are five important functionalities [39] offered by grid computing mentioned below.

- It provides global resource sharing which is very essential component.
- It provides the secure access as in grid computing resource users and resource
  providers may unknown to each other and thus sharing resource conflicts with
  security strategies.
- It provides the efficient and balanced use of resources among its users.
- It provides the remote access where the distance between resource provider and
  users make no difference.
- It operates on open and interoperable standards where everyone from globe can
  contribute in development grid computing strategies and infrastructure.

Resource sharing and coordinated problem solving are important issues considered in a
grid. Sharing should be always conditional. Issues such as payment, negotiation, trust,
policy etc. should be well defined and formed. It is used for coordinated problem solving
like in distributed data analysis, computation, collaboration etc. A formal architecture,
composed of five layers has been previously created to assure this standardization [42].
Figure 1.1 shows these five layers and their relationships.
1. Fabric Layer: Fabric layer is the lowest layer of the layered architecture. It includes the physical devices or resources like computers, storage systems, networks, network devices, instruments and more.

2. Connectivity Layer: This layer is above the fabric layer. This layer includes the communication and authentication protocols required for grid network transactions. For example, the protocol required to share and use data between resources is included in the connectivity layer.

3. Resource Layer: Resource layer includes a set of connectivity protocols. These protocols are used to enable different services like secure initiation, resource monitoring, control of resource sharing etc.

4. Collective Layer: Collective layer rests above the resource layer. It contains APIs (Application Programming Interface), protocols and services which are very essential to implement transactions among resources. Examples of such transaction are resource discovery, resource access, job scheduling in grid network and so forth.

5. Application Layer: Application layer is the highest layer of the layered architecture. It eventually calls on all other layers in order to use, discover and access the resources of grid network.

Figure 1.1: A Layered Architecture of Grid
c) Classifications of Grid

There are different types of grid architectures and topologies exist to suffice different types of business requirements and problems. Some grids are designed to take advantage of extra processing resources, while some grids are designed to support collaboration between various organizations.

The selection of a grid type is based mainly on the business problem that is being solved. Selection of the proper type of grid architecture should be done by considering the specific business requirements and problems. It is also having a direct impact on the grid solution design. A business that wants to tap into unused resources for calculating risk analysis within their corporate data center will have a much different design than a company that wants to open their distributed network to create a federated database with one or two of their main providers and therefore different types of grid applications will require different designs, based on their particular and exclusive requirements [4]. Grids are generally classified in categories of computational grid, collaboration grid, network grid, utility grid and data grid. Figure 1.2 represents classifications of grid.

**Computational Grid:** A computational grid is a combination of hardware and software infrastructure. It aims to provide reliable, pervasive, dependable and inexpensive access to high-end computational capabilities [12]. These grids provide secure access to a huge pool of shared processing power suitable for high throughput applications and computation intensive computing.

**Collaboration Grid:** As the popularity, usage and span of the Internet increase, there has been an increased demand for better collaboration between different entities. Such advanced collaboration is possible using the grid. For instance, users from different places in a virtual enterprise can work on different modules of a project without even disclosing technologies or components used.

**Network Grid:** A network grid provides communication services which are fault-tolerant and high-performance. Each grid node is acting as a data router between two communication points. It provides data caching and other facilities to speed up the communications between such points.
Utility Grid: In utility grid, not only data and computation cycles are shared but the software is also shared. The main services provided through utility grids are software, special equipments, instruments and more. For instance, the application executes on one specific machine. All users send their data to that machine for processing and receive the result back.

Data Grid: A data grid is a grid computing system that deals with data. It provides the controlled sharing and management of large amounts of distributed data. Computational grids are more suited for aggregating resources while data grids mainly focus on providing secure access to distributed, heterogeneous and disparate data resources. Scientific and engineering applications require access to bulk of data, and often these data are extensively scattered. A data grid provides seamless access to the local or remote data required to complete compute intensive calculations. In data grid, the focus is on synthesizing new information from various resources such as geographically distributed repositories, digital libraries, or databases. Data grids provide an infrastructure to support data publication, data handling, data discovery, data storage and data manipulation of large volumes of data actually stored in various heterogeneous databases and file systems.
Chapter 1: Introduction and Overview

Through collaboration, data grids can also include resources such as a federated database. Within a federated database, a data grid makes a group of databases available that function as a single virtual database. Through this single interface, the federated database provides a single query point, data modeling, and data consistency. Data grids also harness data, storage, and network resources located in different administrative domains. They follow local and global policies to govern how data can be used, schedule resources efficiently and provide high speed and reliable access to data. Organizations interested in data grids typically have IT (Information Technology) initiatives to expand data analysis and data mining abilities. In such a way, they are maximizing the utilization of an existing storage infrastructure investment and reducing the complexity of data management [4].

A data grid can expand data capabilities in several ways. First, the databases or files can span across many systems and thus have larger capacities than on any single system. Data transfer rates can be improved by such spanning. Data can be duplicated throughout the grid to serve as a backup and can be hosted on or near the machines which most likely to need the data, in association with advanced scheduling techniques. Sharing is not limited to files rather it also includes other types of resources like software, specialized devices, licenses, services and many more. These resources are virtualized to provide uniform interoperability among heterogeneous grid users. The users of the grid can be members of several real and virtual organizations [4].

1.2.2 Fundamental Concepts of Agent and Multi-agent Systems

a) Introduction to Agents

In the area of AI (Artificial Intelligence), agent oriented technology has been introduced a new paradigm for analyzing, designing and developing independent & reusable software components. Agents are sophisticated entities that act autonomously and proactively on behalf of their users. They are used to solve a growing number of complex problems across open and distributed environments [31]. An agent refers to a component of software and/or hardware that is capable of acting in a certain way to accomplish its tasks and/or goal on behalf of its users. Ferber [18] proposes the following definition that something can be called an agent if it is a physical or virtual entity that follows following characteristics.

- An agent is able to act in an environment.
• An agent can communicate with other agents.
• An agent is driven by a group of tendencies.
• An agent has its own resources.
• An agent is able to perceive its environment (in a limited way).
• An agent has a partial representation of its surrounding environment (and possibly none).
• An agent has skills and offers services.
• An agent may be able to reproduce.
• An agent acts to satisfy its objectives, by taking account of the resources and skills that it possesses, and according to its perception, of its representations and the communications that it receives.

According to Croft [9], an agent is an entity that is authorized to act for another. Intelligent Software Agent (ISA) is a software entity which uses Artificial Intelligence (AI) in the pursuit of the goals of its clients [9]. Artificial Intelligence is the imitation of human intelligence by mechanical means. Clients can reduce human workload by delegating to ISAs tasks that normally would require human-like intelligence. According to [38], on Internet, agent is a program that gathers information or performs some other service without your immediate presence and on some specific schedule. An agent program, using the parameters provided, searches all or some part of the Internet, gathers interested information and presents it on a daily or other periodic basis. An agent is sometimes called a bot (short for robot), as it does the intended work automatically with necessary hardware/software [11].

b) Agent Types

An agent is an umbrella term that covers a range of other more specific agent types. Agents can be classified accruing to various parameters. This section describes the classification of agents according to their nature and role they play.

Collaborative Agents: Collaborative agents interact with each other to share information. They emphasize autonomy and cooperate with other agents in order to perform certain tasks for their owners in open and time-constrained multi-agent environments [32]. These agents are quite useful when the problem domain is too large or resources are widely distributed. Also, collaborative agents help to solve the large scale problems which are too
large for a centralized single agent. They are also seemed to be a solution to inherently distributed problems. An example for collaborative agents includes ADEPT [33] used in business process re-engineering [32].

**Interface Agents:** Interface agents are ideal means to provide a user friendly environment to work with a highly technical application. Interface agents help users to interact with the users. Collaborative agents collaborate with other agents, but interface agents collaborate with the user. Interface agents reduce efforts for the end user and application developer and it can adapt, over time, to its user’s preferences and habits [32]. In literature, there are several interface agents use such as Calendar Agent [35] as assisting its user in scheduling meetings and learning the preferences and commitments of its users, and Letizia [14], a keyword and heuristics-based search agent assisting in web browsing.

**Mobile Agents:** Mobile agents are software agents that are capable of roaming wide area networks, moving to the foreign hosts, performing tasks on there and returning home having performed the responsibilities set [32]. Mobile agents offer an alternative for network and distributed computing. Mobile agents are used in situations where the information must be retrieved from various resources. Sony’s Magic Link PDA is an example mobile agent product. It assists in managing the user’s email, fax, phone, and pager as well as linking the user to TeleScript enables messaging and communication services. TeleScript is an interpreted object-oriented and remote programming language [32].

**Information Agents:** With the advent of information and communication technology (ICT), a vast amount of information is available on the World Wide Web (WWW). After WWW, it becomes very difficult to manage information. Information agents are responsible for managing, manipulating, or collecting information from many distributed sources. These agents meet the needs for information management issues [32]. An example for information agents is Jasper agent [17]. It is used to store, retrieve, summarize and inform other agents of information useful to them found on WWW.

**Intelligent Agent:** An intelligent agent is a software system that can send information to and receive it from other agents using appropriate protocols. Such intelligent agents are able to create action plans, set multiple objectives, process the received information and perform reasoning through a set of AI techniques. An intelligent agent also has a knowledge base and an inference engine. The knowledge base is a repository of
knowledge and rules. Inference engine infers the content stored in the knowledge base in order to generate new knowledge.

**Hybrid Agents:** Hybrid agent is a single agent which combines two or more different agent's philosophies to gain the maximum benefit. The purpose of this hybridization is to achieve dual advantages of different agent types.

c) **Characteristics of Agents**

As stated earlier, an agent is an entity that works on behalf of its users therefore they must be capable of working in an autonomous way without human intervention. For that, agents should have certain characteristics. Table 1.2 shows the typical characteristics of agents.

<table>
<thead>
<tr>
<th>Agent Characteristics</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autonomy</td>
<td>Agents must have the capability to work autonomously without human intervention.</td>
</tr>
<tr>
<td>Cooperation</td>
<td>In order to accomplish the tasks and/or goals, agents must interact with its environment, users and other agents.</td>
</tr>
<tr>
<td>Learning</td>
<td>Agents should be able to learn from the entities with which they interact.</td>
</tr>
<tr>
<td>Reactivity</td>
<td>Agents should be having the capability to perceive their environment and respond to the changes made in that environment.</td>
</tr>
</tbody>
</table>

*Table 1.2: Typical Characteristics of Agents*

d) **Agent Application Domains**

The agent-oriented approach offers an ability to solve problems that can already be solved in a significantly better way (cheaper, more efficient, more natural, easier or faster) [11]. There are several application domains that agents are applied to solve problems. Some of them are described as follows:

**Industrial Applications:** In industrial applications, agents are mostly used in process control, manufacturing, and aviation industries (e.g. Air-traffic control).

**Academic Applications:** In academic applications, agents are used to implement course ware systems, intelligent tutoring systems, advisory systems and many more.
Commercial Applications: In commercial applications, the information management, electronic commerce, and business process management are the areas in which agents are applied. Kashbah [1] is an example of an agent working in the area of e-commerce.

Medical Applications: Medical informatics is a major growth area in computer science. Agents are used in the areas of healthcare, patient monitoring and diagnosis.

Entertainment: The other field for using agents is entertainment. The agents have an obvious role in computer games, interactive theater, and related virtual reality applications. Such systems tend to be full of autonomous and semi-autonomous animated characters and can be implemented through agents [32].

Apart from the above areas, the agent technology is also popular in the area of Research & Development, Government Services, Travels and Tourism Applications etc.

e) Introduction to Multi-agent Systems

A Multi-agent System (MAS) is a loosely coupled network of software agents. These agents interact to solve problems that are beyond the individual capacities or knowledge of individual problem solver. One of the current factors fostering multi-agent development is the increasing popularity of the Internet. It provides the basis for an open environment where agents interact with each other to reach their individual or shared goals. Agents are autonomous entities capable of carrying out specific tasks by themselves or through cooperation with other agents. The multi-agent system is best suitable for autonomous, loosely-coupled, heterogeneous and distributed systems. Such systems need to interoperate in order to achieve a common goal. Multi-agent systems offer a decentralized model of control. They use message passing mechanisms for communication purposes and are usually implemented from an object-oriented perspective [30]. The term multi-agent system can also be used to describe all types of systems composed of multiple autonomous components displaying the following characteristics [20]:

- Each agent has inadequate capabilities to solve a problem;
- There is no centralized system control; data are decentralized; and
- Computations are asynchronous.

Some of the multi-agent system based applications are education, workflow management, network management, air-traffic control, digital libraries, personal digital assistants, business process engineering, e-mail filtering, information management, data mining,
electronic commerce and many more. Agents can share knowledge using any agreed language, within the limitations of the system's communication protocol.

f) Advantages of Multi-agent Systems

The following are some of advantages exhibited from multi-agent systems over a single agent or centralized approach:

- Centralized systems generally suffer with resource limitations, performance bottlenecks or even critical failures. A multi-agent system is a decentralized system and therefore does not suffer from the single point of failure problem. Also, it is having capability to distribute computational and data resources and capabilities across a network of interconnected agents.
- Multi-agent systems allow for the interconnection and interoperation of multiple existing legacy systems. An agent wrapper can be built up around such systems to incorporate them into an agent society.
- Multi-agent systems generally work with multiple, autonomous and collaborative agents which are interacting with one another. This provides a way of representing user preferences, team planning, task allocation, open environments and so forth [31].
- Multi-agent systems are able to retrieve, analyze and coordinate information coming from sources which are spatially distributed.
- Multi-agent systems enhance overall system performance. Also it possesses characteristics like reliability, extensibility, responsiveness, computational efficiency, robustness, maintainability, flexibility and reuse.

1.2.3 Introduction to Knowledge-based Systems

a) Fundamental Concepts

A knowledge-based system (KBS) is a system that draws upon the knowledge of human experts captured in a knowledge base to solve problems that normally require human expertise. A knowledge-based system is a system that uses artificial intelligence techniques in problem-solving processes to aid human decision-making, learning, and action. They are artificial intelligent tools working in a narrow domain to provide decisions with justification. Several knowledge representation techniques are used to acquire and represent knowledge. Such techniques are scripts, rules, frames and so forth.
The basic advantages offered by such system are documentation of knowledge, decision support, reasoning and explanation. Typical KBS goes beyond the decision support philosophy to indicate the expert system technology in the decision making framework.

Expert systems (ES) provide the tools and techniques perfected by artificial intelligence (AI) researchers to deduce decision influences based on codification of knowledge. The codification of knowledge uses the principles of knowledge representation. With the availability of advanced computing & communication facilities and other resources, organizations are demanding more sophisticated computer systems, which might require intelligence. The society and industry across a world are becoming more knowledge oriented and rely on different experts’ decision making ability. KBS can act as an expert system on demand, anytime and anywhere. KBS can save money by leveraging expert advices and thus allows users to function at a higher level and supporting consistency. KBS can be considered as productive tool and it has knowledge of more than one expert for a long period of time. Moreover, a KBS is a computer-based system, which extracts knowledge from data and information. These systems are capable of understanding the information under process and can take decision based on the residing information/knowledge in the system whereas the traditional computer systems do not know or understand the data/information they process.

b) Architecture of Knowledge-based System

A knowledge-based system consists of user interface, a knowledge base and a controlled program for inference from the knowledge base called the inference engine [34]. The inference engine infers the knowledge available in the knowledge base. It is a computer program that tries to derive answers from a knowledge base. This knowledge base can be used as a repository of knowledge in various forms. Figure 1.3 represents an architecture of knowledge-based system.

Figure 1.3: Architecture of Knowledge-based System (KBS)
Chapter 1: Introduction and Overview

Fuzzy logic introduced a new era for knowledge-based systems and expert systems in particular. Fuzzy logic is being integrated in many expert systems for real world problems. It has brought the notion of graded membership between complete membership and non membership. With the use of fuzzy logic, it is now feasible to emulate human thinking using computers.

1.2.4 Fuzzy Logic and Fuzzy Expert Systems

Knowledge-based systems are systems which are designed to emulate human thinking to solve problems and provide advices [16]. One kind of knowledge-based systems is an expert system. Although it is widely used in various applications, such systems are not able to model real world problems which are full of ambiguities and vagueness. When fuzzy logic was introduced by Dr. Lotfi Zadeh in 1965 [40], it did not get the attention of expert system's researchers. The idea of fuzzy logic was to show that there is a world behind conventional logic. Fuzzy logic provides a proper way to model human thinking. Although fuzzy logic has been introduced forty years ago, it is recently getting the attention of artificial intelligence researchers. Fuzzy logic is being used to develop expert systems for handling ambiguities and vagueness associated with real world problems. Expert system, which uses a collection of fuzzy sets and rules to facilitate reasoning, is called a fuzzy expert system [24].

a) Fuzzy Logic Basics

Fuzzy logic is a superset of conventional (Boolean) logic that has been extended to handle the concept of partial truth, i.e. truth values between "completely true" and "completely false". Dr. Lotfi Zadeh of UC, Berkeley [40] has introduced the concepts of fuzzy set theory and fuzzy logic in the 1965's as a means to model the uncertainty of natural language. According to him, rather than regarding fuzzy theory as a single theory, we should regard the process of “fuzzification” as a methodology to generalize any specific theory from a crisp (discrete) to a continuous (fuzzy) form.

b) Fuzzy Sets

Fuzzy logic is based on fuzzy sets. A fuzzy set is a class of objects with continuum grades of membership [40]. A fuzzy set is an extension of a conventional (Boolean) set. A fuzzy set has elements belonging to it to some degree of membership. This degree varies from 0
to 1. For example, the set of young girls is a fuzzy set. In conventional or crisp logic, the
degree of membership is either 0 for non membership or 1 for complete membership.
Whether, fuzziness results from imprecise boundaries of fuzzy sets [40, 36]. It is based on
emulating human thinking where elements are linguistic variables. Linguistic variables are
the variables, whose values are sentences, not numbers. There are different categories of
linguistic variable values. Linguistic variable values could be primary terms, which are
labels for specific fuzzy sets or they could be hedges such as very regarding the atomic
value [41].

There are two main reasons why classical logic systems cannot cope with problems in
which knowledge is approximate. First, they do not provide a means for representing the
meaning of propositions expressed in a natural language when it is imprecise. Also they do
not provide a mechanism for inference in the cases where knowledge is represented
symbolically in a meaning representation language [25]. On the other hand, fuzzy logic
provides a way of representing knowledge about systems which are either too complex or
too imprecise. Second, with the growing complexity of the system, the ability to develop
precise analytical models of the system diminishes to a point beyond which it becomes
impossible [10]. This approach allows making inference using approximate knowledge to
make correct and logical decisions on actions to be performed.

In classical set theory subset U of a set P can be defined as a mapping from elements of P
to elements of the set \{0, 1\}.

\[
U: P \rightarrow \{0, 1\},
\]

Where the value 0 represents non-membership and the value 1 represents the complete
membership. Furthermore, the statement X is in U is true if there exists and ordered pair
\((X, 1)\). X is in U is false if there exists and ordered pair \((X, 0)\).

In fuzzy theory, subset U of a set P can be defined as a set of ordered pairs with the first
element of S and the second element a value from the interval \([0, 1]\). The central notion of
fuzzy systems is that truth is represented by a number between 0 and 1. The truth value 0,
represents absolute falseness, whereas truth value 1, represents absolute truth. The entire
set of in-between values represents the degrees of membership or degree of truth. The set
P is the universe of discourse for U. The mapping is the membership function of U [41].
c) **Membership Function**

The membership function provides the degree of membership to the fuzzy set, for every element in the universe of discourse. It maps the elements of $X$ with a number in the interval $[0, 1]$.

$$\mu_A(x) : X \rightarrow [0, 1]$$

Where $\mu_A(x)$ is the membership function of a fuzzy set $A$ [19].

On the other hand, the fuzzy set $A$, may represent as a set of ordered pairs [22]:

$$A = \{(x, \mu_A(x)) ; x \in X, \mu_A(x) \in [0, 1]\}$$

It is important to distinguish the fact that membership functions are possibility functions. They are not probability functions. When the closure $\mu_A(x)$ is to 1, the more $x$ belongs to the fuzzy set $A$.

**d) Linguistic Variables and Linguistic Hedges**

Linguistic variable is a key concept of fuzzy logic which exploits the tolerance of imprecision [36]. It is a label of a fuzzy set which is characterized by a membership function. Therefore, a linguistic variable can be described as a variable whose value can be illustrated both quantitatively using a membership function and qualitatively using a linguistic term [8]. Therefore, it is a word or a sentence used in a natural language. Let us take an example of a linguistic variable “temperature”. Its values are assumed to be “very hot”, “hot”, “cold”, “very cold” [23]. The values, the linguistic variable can take, are formed from a primary term (“hot”), its antonym (“cold”), modifiers (“very”, “not”...) and connectives (“and” and “or”). Therefore, a possible value of “temperature” could be “not very hot and not very cold” [10].

When a fuzzy set is assigned to a linguistic variable, the possible values of that linguistic variable are being constrained, called possibility distribution [26]. The concept of a linguistic variable is the essence of fuzzy logic applications because it models the way humans think, perceive and reason [27]. There exist linguistic variables of the form “very tall.” The fuzzy adjective “very” is a linguistic hedge. Other linguistic hedges can be slightly, extremely, more or less, highly etc. [10].
e) Properties of Fuzzy Sets

If we assume that, there are three fuzzy sets $A$, $B$ and $C$ of a universe of discourse $U$, a set of properties regarding these three sets can be derived by simply applying the rules for complement, intersection and union of fuzzy sets. Table 1.3 lists the properties of fuzzy sets.

An important difference between fuzzy logic and crisp (traditional Boolean) logic are the rules of inclusion. In fuzzy set theory and fuzzy logic, the union of a fuzzy set $A$ and its complement $A'$ is not equal to the universal set. They are a subset of it. Besides, the intersection of a set $A$ and its complement $A'$ in crisp logic is always the empty set. However, in fuzzy logic, this is not the case. Here, this intersection set may have non zero membership values.

<table>
<thead>
<tr>
<th>Property Name</th>
<th>Relation</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Commutativity</strong></td>
<td>$A \cap B = B \cap A$</td>
</tr>
<tr>
<td></td>
<td>$A \cup B = B \cup A$</td>
</tr>
<tr>
<td><strong>Associativity</strong></td>
<td>$(A \cap B) \cap C = A \cap (B \cap C)$</td>
</tr>
<tr>
<td></td>
<td>$(A \cup B) \cap C = A \cup (B \cap C)$</td>
</tr>
<tr>
<td><strong>Distributivity</strong></td>
<td>$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$</td>
</tr>
<tr>
<td></td>
<td>$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$</td>
</tr>
<tr>
<td><strong>Absorption</strong></td>
<td>$A \cup (A \cap B) = A$</td>
</tr>
<tr>
<td></td>
<td>$A \cap (A \cup B) = A$</td>
</tr>
<tr>
<td><strong>Idempotency (of same power)</strong></td>
<td>$A \cup A = A$</td>
</tr>
<tr>
<td></td>
<td>$A \cap A = A$</td>
</tr>
<tr>
<td><strong>Exclusion</strong></td>
<td>$A \cup A' \subseteq U$</td>
</tr>
<tr>
<td><strong>(Law of excluded middle)</strong></td>
<td>$A \cap A' \supseteq U$</td>
</tr>
<tr>
<td><strong>(Law of contradiction)</strong></td>
<td></td>
</tr>
<tr>
<td><strong>DeMorgan's Laws</strong></td>
<td>$(A \cap B)' = A' \cup B'$</td>
</tr>
<tr>
<td></td>
<td>$(A \cup B)' = A' \cap B'$</td>
</tr>
<tr>
<td><strong>Boundary conditions</strong></td>
<td>$A \cup U = U$</td>
</tr>
<tr>
<td></td>
<td>$A \cap U = A$</td>
</tr>
<tr>
<td></td>
<td>$A \cup \emptyset = A$</td>
</tr>
<tr>
<td></td>
<td>$A \cap \emptyset = \emptyset$</td>
</tr>
</tbody>
</table>

*Table 1.3: Properties of Fuzzy Sets*
1.3 Conclusion

This chapter defines the problem of research, clears general scenario and states the objective of the research work. The various aspects, issues and challenges faced by data grid and multi-agent knowledge-based systems have been discussed in detail. In current scenario, data is distributed across multiple geographical locations. It uses different formats and access mechanisms. Therefore, a solution which provides data access, integration and sharing would be advantageous. As the scale and ambition of the grid and agent technologies have been increased, an integrated approach is required to leverage the power of both grid and agent technologies. This chapter introduces the basic concepts of grid computing, multi-agent systems and knowledge-based systems. In the next chapter, we will discuss about the literature survey conducted to carry out the research work. It will explain the need of distributed database grid, the available middleware for data grid, the need of integration for data grid and agent applications and clears the motivation behind the development of a multi-agent knowledge-based system accessing distributed database grid.

References

17. J. Davies, R. Weeks, Revett M.J., Communicating information agents for WWW, Ipswich IP5 3RE UK, BT Laboratories, 1996.


