Content

2.1 Data Grids in Practice ....................... 32
2.2 Agent & Multi-agent Systems in Practice .................................................... 42
2.3 Multi-agent Knowledge-based System by integrating Fuzzy Logic......................... 48
2.4 Motivation to Develop Multi-agent Knowledge-based System Accessing Distributed Database Grid............. 49
2.5 Conclusion................................................... 51

References.................................................. 52
Chapter 2

Literature Survey

This chapter provides details about the literature survey conducted to carry out the research work. It delivers the details about the work accomplished in the area of data grid, agent & multi-agent systems and knowledge-based systems by integrating fuzzy set theory and fuzzy logic. It also covers the aspects like study of related work, issues and challenges, techniques and methodologies and more. It also enlightens the need of distributed database grid, the available middleware for data grid, the need of integration for data grid and agent applications. Furthermore, it presents the motivation behind the development of a multi-agent knowledge-based system accessing distributed database grid.

Section 2.1 of this chapter presents the details about data grids, its applications and data grid middleware. Section 2.2 explains about agent and multi-agent systems and related work in this area. Section 2.3 demonstrates the integration of fuzzy set theory and logic with multi-agent knowledge-based systems. Section 2.4 presents the motivation behind the development of a multi-agent knowledge-based system accessing distributed database grid.

Literature survey is supported by a critical assessment of the work done so far to show how the current study relates to what has already been done. For studying the research work conducted by various researchers, the libraries and knowledge resource centers of various universities like Sardar Patel University - V V Nagar, Charotar University of Science & Technology – Changa, Gujarat University – Ahmedabad have been visited. Several scholarly books, dissertations, journal articles, technical reports, conference proceedings, websites and other internet resources online and in physical form related to research area have been studied.

Apart from that, fruitful literatures have been collected from various e-journal gateways like J-Gate, Springer Online and many more. A public search engines and digital libraries containing scientific and academic papers like CiteSeer have been visited to get details about the work accomplished in the research area. Teachers, students, academicians, experts, developers from various universities and industries have been
Chapter 2: Literature Survey

contacted for identifying the feasibility and application of integration of multi-agent knowledge-based system and distributed database grid in business and commercial applications. The different techniques and tools necessary to implement the research have been studied. The requirements of developers as one of the inputs for deriving the framework have been noted.

By conducting a literature survey, a critical & analytical summary and synthesis of the current knowledge of my research area has been developed. This is also extended with the comparison and relevance with different theories and findings through exploration of past and current work in the research area.

2.1 Data Grids in Practice

2.1.1 Introduction to Data Grid

Grid computing provides an Information and Communication Technology (ICT) infrastructure to facilitate seamless and secure sharing of heterogeneous and geographically distributed resources (e.g. storage, computation, data, algorithms, applications and networks). Great resources, efforts and funding have been placed into regional, national and international initiatives in grid core technologies, grid infrastructure and grid applications. According to Heinz Stockinger [14], grid computing in general comes from high performance computing, supercomputing and clusters computing where several work stations or processors are connected via a high-speed network in order to execute a mutual computational task. Initially, grid computing was meant to span a local area network but then it was also extended to the wide area network such as the Internet. A grid itself is supposed to connect computing resources over the wide area network.

The grid research field can further be divided into two large sub domains: Computational Grid and Data Grid. A computational grid is an extension of the former cluster computer where large computational tasks execute at distributed computing resources such as workstations. A data grid deals with efficient placement, management and replication of huge amounts of data resources.
2.1.2 Databases and Data Grid

A database is a collection of related data. A Database Management System (DBMS) is responsible for the storage and management of one or more databases. A DBMS supports a variety of database paradigm such as relational, object or object-relational. A Database System (DBS) is created to manage a specific database by using a DBMS. Many grid applications commonly employ more than one DBS. DBS and DBMS offer a set of services that are used to manage, manipulate and access the data [31]. These services may include query and transaction services. A service typically provides a set of related operations. Distributed Database Management System (DDBMS) is software for managing databases stored on multiple computers in a network. A distributed database contains a set of databases stored on multiple computers that typically appears to applications as a single database. Consequently, a DDBMS can simultaneously access and modify data in several databases within a network. A DDBMS mainly classified into two types.

❖ **Homogeneous Distributed Database Management Systems**

In a homogeneous distributed database management system, all sites have identical DBMS software and they are aware of each other and agree to cooperate in processing the requests sent by users. To change schema or software, each site surrenders part of its autonomy. Homogeneous DDBMS appears to the user as a single system. In homogeneous database management systems, the operating system, the data structures and the database applications used at each site must be same or compatible with each other. The homogeneous DDBMS is much easier to design and manage.

❖ **Heterogeneous Distributed Database Management Systems**

In a heterogeneous distributed database management system, different sites may use different schema and different DBMS software. Difference in a schema is a major concern for query and transaction processing. Sites may provide only limited facilities for cooperation in transaction processing and may not be aware of each other. In heterogeneous DDBMS, different nodes may have different hardware & software. They may have different and incompatible data structures at various nodes or locations. At each of the locations, different database applications, data models or computers and operating systems may be used.
For example, one location may have latest relational database management system, while another location may store data using conventional files or an old version of database management system. Similarly, one location may have the UNIX operating system, while another may have the Windows NT as an operating system. Heterogeneous DDBMS are usually used when individual sites use their own hardware and software. Here, translations are required to allow communication between different sites. In this system, the users must be able to make requests (such as query processing) in a database language at their local sites and usually the SQL database language is used for the same. In Heterogeneous DDBMS, often, a user at one location may be able to read but not update the data at another location.

Data grids basically deal with providing services and infrastructure for distributed data intensive applications. Such applications need to access, modify and transfer huge datasets stored in distributed data storage resources [43]. The following capabilities are needed to derive maximum benefits from data grids:

- Data grid should able to search through several available data resources for the required data set. It should also be able to discover suitable data resources in order to access these data.
- Data grids should able to transfer large sized datasets between resources in a timely manner (transfer time should be as short as possible).
- Users of data grids should able to manage multiple copies of their data. Also, the access permissions and privileges to access data should be managed by data grids.
- Data grids should able to select suitable local or remote computational site in order to process data on them.

Some of other paradigms with similar requirements to support a distributed data intensive infrastructure are peer-to-peer file-sharing networks, content delivery networks and distributed databases [43].

Figure 2.1 shows a high level view of a worldwide data grid consisting of computational and storage resources in different countries that are connected by high speed networks [43]. The thick lines in the Figure 2.1 show high bandwidth networks. They link the major data centers. The thin lines in the Figure 2.1 show lower capacity networks that connect the subsidiary data centers. Basically, the data generated from an experiment, instrument or a network of sensors is stored in its principal storage site. It is then transferred to the
other storage sites around the globe on request through the data replication mechanism [43].

Figure 2.1: A High Level View of Data Grid

To locate the dataset, users may query their local replica catalog. For that, users must have been granted with appropriate access rights and permissions. If data is present in their local replica catalog, they are fetched from there. Otherwise, the data is fetched from a remote repository. Apart from this, data may be transmitted to a computational site for further processing.

The computational site can be high configuration computer, super computer or even a desktop. Once, the processing has been completed on computational site, the result may be sent to the machines owned by the individual users, shared repository, data storage site and so forth.

There are significant differences exist between distributed databases and data grids. Table 2.1 shows the differences between Distributed DataBases (DDB) & Data Grids.
Chapter 2: Literature Survey

<table>
<thead>
<tr>
<th>Distributed Databases (DDB)</th>
<th>Data Grids</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manages at DBMS level</td>
<td>Manages at the Application level</td>
</tr>
<tr>
<td>Access for both, read and write</td>
<td>Access for only read. Rarely for write</td>
</tr>
<tr>
<td>Mostly homogeneous, implementation is DBMS specific and so not interoperable</td>
<td>Mostly heterogeneous, implementation is application specific and so interoperable</td>
</tr>
<tr>
<td>Equal distribution of data</td>
<td>Unequal distribution of data</td>
</tr>
<tr>
<td>For structured data</td>
<td>For structured &amp; unstructured data</td>
</tr>
<tr>
<td>Component sites are equal in terms of their processing capability</td>
<td>Database nodes may vary drastically in terms of their availability and processing cost</td>
</tr>
<tr>
<td>Appears to application as a single database. i.e. data integration is done by DBMS and one node acts as a centralized DBMS</td>
<td>Appears to user as a single database. i.e. data integration is done by the application (and generally through web services) and decentralized</td>
</tr>
<tr>
<td>Tightly synchronized</td>
<td>Asynchronous by nature</td>
</tr>
<tr>
<td>No middleware services are used. i.e. user or DBA has to manage the data access &amp; integration</td>
<td>Middleware services are used to manage data access &amp; integration</td>
</tr>
</tbody>
</table>

Table 2.1: Differences between Distributed Databases (DDB) & Data Grid

The Need for Databases in Grid Environments: Mario Antonioletti et al. [25] explain the need for databases in grid environments. According to them, early grid applications are mainly motivated by an attempt to provide scientific-based computing. They were often closely associated with devices or tools that read and/or generate flat files. Early grid toolkits were providing support for files rather than for the management of structured data [45]. However, over time, the file management systems and registries associated with grid toolkits themselves became complex, and database management systems were increasingly used to store grid metadata [29].

Contemporaneously, the requirements of the scientific computing community have become more sophisticated. For example, biological and astronomical communities generate large quantities of data that increasingly use databases for data storage and data retrieval. Also, engineering, medical research, healthcare and many government & legacy
systems can also take advantages of grid infrastructure to access and integrate multiple and
distributed collections of structured data. This data needs to be made available and
accessible to users and their applications, which makes these fields ideal candidates to
adopt grid-based infrastructures.

2.1.3 Data Grid Applications

Saleh Saeed AlZahrani [38] has presented some applications of data grids in various
fields. They are as follows:

**UK National Grid Service (NGS):** The National Grid Service (NGS) [National Grid
Service (NGS), www.grid-support.ac.uk.] provides UK researchers with coherent
electronic access to computational and data and other resources for their research,
regardless of location. The NGS is funded by the Joint Information Systems Committee
(JISC), the Engineering and Physical Sciences Research Council (EPSRC) and the
Council for the Central Laboratory of the Research Councils (CCLRC). The NGS was
formed in October 2003 and entered full production in September 2004. It is led and
coordinated by the Science and Technology Facilities Council (STFC) in collaboration
with the Universities of Oxford, Manchester, Edinburgh and the White Rose Grid at the
University of Leeds. Researchers from different institutions across the UK can use its
infrastructure to obtain standardized access to computing, data and other large scale
facilities. They may also collaborate with researchers from other countries.

**The Sakai Project:** Sakai [Sakai project, www.sakaiproject.org] is a free, open source
online Collaboration and Learning Environment. It provides support for a variety of
operations such as ad hoc group collaboration, teaching and learning, portfolios and
research collaboration and so forth. The Sakai community encourages developers to create
new functionalities and improve the existing ones.

**Enabling Grids for E-Science (EGEE) Project:** The Enabling Grids for E-science
(EGEE)[The Enabling Grids for E-science (EGEE) project, www.eu-egee.org] project
integrates applications from many scientific fields to provide scientists from more than 90
institutions in 32 countries with a seamless grid infrastructure. This infrastructure operates
24 hours a day. EGEE's grid of more than 20,000 CPUs containing some 5 petabytes (5
million GB) of storage provides the solution to the time and resource constraints of
traditional Information Technology (IT) infrastructures. It can run an average of 20,000
concurrent jobs at a time. This capacity means that it is users' needs rather than system constraints that decide the use to which the grid is put. The grid provides a wide range of bandwidth and computing power and large storage capacity that satisfies all types of resource needs.

**Sloan Digital Sky Survey (SDSS) Project:** SDSS [The Sloan Digital Sky Survey (SDSS), www.sdss.org] is the most ambitious astronomical survey project ever undertaken. It is a part of a change in the way science is conducted. It now processes vast amounts of data very quickly, so that it has become feasible for SDSS to map a quarter of the sky, producing detailed 3-D images of more than 100 million celestial objects in that segment, as well as determining their positions and brightness. Its range extends to 10,000 known quasars at the edge of the visible universe.

**TeraGrid:** TeraGrid [TeraGrid project, www.teragrid.org] is an open scientific discovery infrastructure combining leadership class resources at eleven partner sites to create an integrated, persistent computational resource coordinated through the University of Chicago's Grid Infrastructure Group working in partnership with Indiana University and several computing centers including the National Center for Supercomputing Applications, as well as the University of Chicago/Argonne National Laboratory and the National Center for Atmospheric Research.

### 2.1.4 The OGSA Grid Environment

Web services are emerging from the business computing world with the aim of supporting business-to-business relationships in a language-neutral and platform independent way. They are interoperable, distributed and loosely-coupled components which interact with each other through the messages-passing mechanism. A collection of specifications is currently progressing through standardization bodies such as the OASIS (Organization for the Advancement of Structured Information Standards) and W3C (World Wide Web Consortium) in order to achieve the goal of interoperability. Starting in 2001, researchers, initially led by Globus [45] and IBM, began developing new grid standards, technology and infrastructure [25]. The aim was to merge the understanding developed through the design and deployment of early grid applications (such as Globus Toolkit 2) with the promising web services middleware solution. Their goal was to allow grid developers to make use of huge commercial investment in web services infrastructure and tooling. The result was the Open Grid Services Architecture (OGSA) [16]. It is a high-level framework
designed to support dynamic virtual organizations that share independently administered data and resources seamlessly across a network of heterogeneous and distributed computers. This architecture is still under development and will define the major functional components required to meet said requirements. OGSA-DAI (Open Grid Services Architecture – Data Access and Integration) is one of these high-level functional components.

**Databases and the OGSA Grid Environment:** According to Mario Antonioletti et al. [25], it has become increasingly clear that if the OGSA (Open Grid Services Architecture) support a wide range of communities, then database integration is essential. They present an approach to integrating databases into the OGSA grid environment so that communities may expose and share structured data resources and applications may access and update data stored in them. According to them, there are two main dimensions of complexity to the problem: reconciling implementation differences between server products within a single database paradigm (IBM, Oracle, Microsoft, etc.), and the variety of database paradigms (object, relational, XML etc.).

The majority of the currently deployed database management systems supports OGSA grid integration. Although, each DBMS is the result of many hundreds of person-years efforts to provide a wide range of valuable programming interfaces and tools, and important properties such as performance, security and dependability. As these attributes are required by applications, they believe that building new grid-enabled database management systems from scratch is unrealistic, infeasible and uneconomic.

The emergence of the Open Grid Services Architecture (OGSA), to complement the ongoing activity on web services standards, agrees to provide a service-based platform that can meet the needs of both business and scientific applications. They describe the design and implementation of OGSA-DAI, a service-based architecture for heterogeneous and distributed database access in the grid environment. The approach involves the design and development of grid data services which allows clients to discover the properties of structured data stores and to access their contents. The original focus was to provide support for access to Relational, File and XML data, but the overall architecture has been designed to be extensible to accommodate different storage paradigms.
2.1.5 Open Grid Services Architecture – Data Access and Integration (OGSA-DAI): A Middleware Infrastructure for Data Grids

The Open Grid Services Architecture – Data Access and Integration (OGSA-DAI) project was established to provide a uniform way to access data resources in alignment with OGSA. It enables client applications to submit request documents in order to perform a set of tasks on a remote data resource. The project was funded by the UK e-Science core program and is working closely with the Global Grid Forum Data Access and Integration Services (GGF DAIS) work group. The aim of the OGSA-DAI project is to explore the requirements and approaches to integrate databases into the OGSA grid environment in order to influence both the emerging database standards and established database vendors.

OGSA-DAI has been designed, developed and released as a collection of services for integrating database access and integration operations with the core capabilities of OGSA. Thus it allows structured data resources to be seamlessly integrated into OGSA grid applications. The goal of OGSA-DAI is to provide a uniform service interface for data access and integration with databases exposed to the grid and hiding differences such as database types, database driven technology, data delivery mechanisms, data formatting techniques and data querying techniques. Furthermore, OGSA-DAI services can provide the basic operations that can be used by the higher level services to offer greater functionalities, such as a multiple data federation, data replication and distributed queries. The OGSA-DAI infrastructure also promotes the design and development of efficient applications by providing support for grouping multiple requests on an OGSA-DAI service into a single message sent to a service. This reduces delay by increasing the granularity of interactions. It also reduces both, the number of messages exchanged and the quantity of data transferred.

In a grid setting, the European Data Grid has developed Spitfire [49], a web service interface to relational databases for metadata management. Spitfire [49] has developed an infrastructure that allows a client to query a relational database over GSI (Grid Security Infrastructure)-enabled HTTP(S). To represent the query and its result, an XML-based protocol is used. The system supports role-based security, in which, a client can specify the role they wish to adopt for a query execution. A mapping table resides on the server that checks for authentication of the client to take on this role. OGSA-DAI has implemented a similar approach to Spitfire for authentication. However, it differs in
supporting multiple access languages, asynchronous and third-party delivery [25]. Clients send rich documents which may contain multiple activities (such as, a set of queries) to OGSA-DAI servers. When compared with database services that only support fine-grained interactions (e.g. single queries), this reduces the number of interactions (important for increasing performance of server and application) and offers the opportunity for the server to perform local optimizations. Furthermore, unlike other approaches, OGSA-DAI allows developers to model their own activities and make them available to client or consumers. This feature has been used by a number of research groups and could be used to expose specialist functionality such as the local data mining capability to database consumers [25].

There is a clear relationship between OGSA-DAI and other data grid functionalities like efficient movement of data [17] or data replication. For example, a data replication service could use OGSA-DAI to read data from or write data to structured data storage resources [28]. Higher-level grid data management systems, such as Chimera [11] or SRB (Storage Resource Broker) [32] could use OGSA-DAI either to access structured data from grid environment, or to provide access to their own metadata. OGSA-DAI provides a range of mechanism which support access, integration, sharing, management and coordinated use of data on the grid.

**Research on Data Grid:** Samatha Kottha et al. [39] tested a usability of grid application in RNAi screening. It was evaluated in accessing data from bio-databases using the queries. These queries were executed by a potential user of RNAi screening. The observations show that OGSA-DAI has some considerable overhead compared to a JDBC connection but provides additional features like security which in turn are very important for distributed processing in life sciences.

Zhou HUANG et al. [54] have presented the GEOBARN – A practical grid Geospatial Database System which combines the grid computing technique with geospatial database and develop a prototype GEOBARN for reliable data management and effective distributed geospatial query execution. They observed that, system availability and efficiency can be improved after applying grid computing to geospatial data processing and sharing. Eduard V. Snezhko et al. [9] have devoted to consideration of large-scale data management, methodological, software developing and medical imaging issues concerned with the distributed computing in a nationwide system.
The distributed computing architecture employs the Unicore as a middleware solution, as well as the Message Passing Interface for utilizing the grid infrastructure. Hsin-Chuan Ho et al. [15] hope to integrate idle computer facilities in academies by grid computing and the techniques of access grid and data grid so as to save cost and make best use of resources. For those academies that have insufficient budgets, they can then obtain better services and enormous teaching resources through the techniques of access grid and data grid.

Luo Y, Jiang L and Zhuang TG [21] presented a grid-based model for integration of distributed medical databases. They have presented a prototype system for integration of heterogeneous medical data resources based on grid technology with OGSA-DAI. A model provides a standard framework for accessing and integrating data resources in a distributed environment.

2.2 Agent & Multi-agent Systems in Practice

2.2.1 Agent Technologies & Application Domains

Agent technologies have recently emerged as a promising tool to solve complex and difficult problems. An agent is a software program that proactively performs tasks on behalf of the user [27, 36]. There exists no single definition for agents, but a lot of discussion. Almost every author seems to propose own needs and ideas what leads to a variety of definitions depending on the targeted problem area [44].

The expressed spectrum determines reasonable application areas of agents such as telecommunications, user interfaces, network management, information gathering and electronic commerce [40]. Russel and Norvig described this multiplicity aspect in following way [33]: "The concept of an agent is meant to be a tool, not an absolute characterization that divides the world into non-agents and agents." The Foundation for Intelligent Physical Agents (FIPA) describes agents as computational processes that implement the autonomous and communicating functionality of an application [12]. Mostly corresponding architectures, systems and points of view are based on using attributes as defining entities. For example, Wooldridge and Jennings characterize agents as software-based computer systems with certain properties like reactivity, autonomy, pro-activeness and co-operation [51].
Figure 2.2 shows the agent and their interaction with the environment. The agent accepts external environmental factors and incoming messages from other agents. It produces outgoing messages. It also takes actions according to the environmental factors [13].

A multi-agent system is a loosely coupled network of problem solver entities that work together to find answers to problems that are beyond the individual capabilities or knowledge of individual entity [46]. Many agent-oriented methodologies have been developed based on a variety of notations, concepts, techniques and methodological guidelines. Multi-agent systems are applied in the real world to graphical applications such as computer games. Agent systems are used for entertainment. Some of the examples are films, virtual reality etc. They are also used for coordinated defence systems.

Other applications of multi-agent systems include transportation, logistics [24], graphics, GIS (Geographical Information System) and so forth. Furthermore, multi-agent systems are widely being advocated to use in networking and mobile technologies to achieve functionalities such as, high scalability, automatic and dynamic load balancing, self-healing networks and so forth. Some of other agent application domains are Commercial Applications, Industrial Applications, R & D Applications, Medical Applications, Government Systems, Educational Applications and Entertainment.
Research on Agent and Multi-agent Systems: Vera Maria B. Werneck et al. [47] discussed the work which was part of a broader project which aims at analyzing important aspects of modeling and developing different multi-agent systems using several methodologies. Saint-Germain B et al. [37] presented the development of a multi-agent manufacturing control system for an industrial application. M. Caridi and S. Cavalieri [22] used multi agent systems in production planning and control. Dimitar Nedev and Veselina Nedeva [8] describe the applicability of agent-based technology in the e-learning systems. Antonio Moreno [4] described the usage of three multi-agent systems that tackle diverse problems, such as the management of organ transplants across a nation, the access to medical information of a city and the intelligent management of the data of the palliative patients of a hospital. S Srinivasan, Dheeraj Kumar and Vivek Jaglan [34] proposed a multi-agent architecture for integrated dynamic scheduling of the steel pipe industry. The scheduling systems of these processes have different objectives and constraints. They operate in an environment where there is a substantial quantity of real-time information concerning production failures, supplier information, customer requests and order processing. An agent is allocated to each process, which independently seeks an optimal dynamic schedule at a local level. The local objectives, real-time information and information received from other agents are taken into account.

A multi-agent education software system has been developed by Shanghua Sun, Mike Joy and Nathan Griffiths [42], which incorporates learning objects. It is based upon a learning style theory as the foundation for its adaptivity. M. P. Tariq, M. Waqar Mirza and R. Akbar [23] presented a multi-agent based approach for the development of a timetable scheduling system for educational institutes. The information is captured and saving agent captures data from user and saves it into the database.

2.2.2 Agent, Web Services and SOA

Computer systems have evolved since their inception as to be able to tackle more and more complex problems. Moreover, system architecture and methodologies for programming must evolve accordingly. The evolution can be described from a centralized monolithic single-threaded systems to distribute componentized parallel systems. Currently, a new methodology for building such complex distributed systems is being widely discussed, named as Service-oriented Architecture (SOA).
A service-oriented architecture presents a solution to expand upon the scope of a program. The program treats its components and functions as individual and autonomous services. A service-oriented architecture is basically a collection of such services. These services communicate with each other in order to accomplish a given task. The communication may involve either simple data passing or it may involve two or more services coordinating specific activity. Therefore, some means of connecting such services to each other is needed. SOA is a methodology for building distributed systems using independent components, named services. Services communicate by exchanging messages, and their interfaces are described in a machine-processable way [26].

The definition of SOA is independent of technology. However, the most widely used implementation of SOA is in web services. Here, services communicate using established Internet standards, namely HTTP (Hyper Text Transfer Protocol) and XML (eXtensible Markup Language). These make services interoperable [26].

A web service is a way of communication between two electronic devices over the WWW (World Wide Web). It is a service that is "always on" as in the concept of utility computing. The W3C defines a web service as "a software system designed to support interoperable machine-to-machine interaction over a network such as WWW". A web service is also termed as a software entity available at a network address over the web or the cloud. An interface to a web service is depicted in a machine-processable language (specifically Web Services Description Language (WSDL)). Other systems interact with the web service in a manner prescribed by its description using SOAP (Simple Object Application Protocol) messages. They are typically conveyed using HTTP with an XML serialization in conjunction with other web related standards [50].

Figure 2.3 presents the architecture of a web service. It provides a set of functionalities to businesses and individuals that enables universal access to these functionalities. A web service has moved from a tightly-coupled structure to a loosely-coupled structure using existing web protocols. Examples of these protocols are Hyper Text Transfer Protocol (HTTP) and Simple Mail Transfer Protocol (SMTP). As shown in the Figure 2.3, web service architecture consists of three entities:
Service Providers: service providers create web services and publish these services to the outside world by registering the services with service brokers using a registry.

Service brokers: service brokers maintain a registry such as UDDI (Universal Description, Discovery and Integration) of published services.

Service Requesters: Service requesters find required services by searching the service broker's registry. Requesters then bind their applications for the service provider to use particular services.

Research on Agent, Web Services and SOA: S Srinivasan, Dheeraj Kumar and Vivek Jaglan [34] explained the web services and a service-oriented style of architecture which is widely seen as the basis for a new generation of distributed applications and system management tools. In this paper, they have introduced the key concepts, relationships and benefits of these technologies. They have indicated how these technologies can be combined to develop highly scalable application systems that can span across a management and ownership domains, irrespective of the hardware and software platforms deployed in each. Savas Parastatidis et al. [41] aimed to clarify the fundamental tenets of SOA and their relevance to Internet scale computing (or grid computing). They show how to apply the principles of SOA to building Internet scale applications using web services technologies and how to avoid software pitfalls by adhering to a number of deliberately simple architectural constraints.
coming from distributed datasets in the data grid environment and facilitates the multiple database federation by imparting the standards and services incorporated into the data grid. For that, it uses various APIs (Application Programming Interface) to implement transactions among data resources which are enabling applications to federate multiple databases at a single time.

This layer consists of services which interact with knowledge layer and application layer. Some of the examples of such services are data source service (to find the best available data source available in data grid), information service (to manage and monitor the current active services available in the data grid), and metadata service (to find the physical address of the logical data). The services of the information layer provide a holistic view of data to the higher level layers i.e. knowledge and application layers.

3.1.4 Knowledge Layer

The knowledge layer is directly accessible by application layer and it is built upon the information layer. It may implement the artificial intelligence techniques like fuzzy set theory & fuzzy logic to convert the data into its meaningful form called knowledge. It fetches the data from information layer and applies a specific set of services and activities which extracts the knowledge from the fetched data. The extracted knowledge is then after presented and delivered to application layer which may use for analysis and decision making process by client applications and users.

3.1.5 Application Layer

This is the highest layer of an architecture and accesses all other layers for resources where and when required. It also allows access to remote data resources and digital libraries, and provides overall management of all applications running. This layer can be customized according domain specific applications and used by several data intensive research and business applications. The users are directly accessing the application layer in order to access the lower layers to fulfill their requests. The application layer then passes the requests to activities and services reside in the lower layers and send the response back to the users about their status of the request.
3.2 The Generic Framework for Integration of Multi-agent Knowledge-based System and Distributed Database Grid

The grid community has historically focused on tools, applications and infrastructure for reliable and secure resource sharing within dynamic, heterogeneous and geographically distributed virtual organizations. On the contrary, the agent community has focused on autonomous problem solvers that can act flexibly & automatically in uncertain and dynamic environments as stated earlier. As the scale and ambition of grid and agent deployments increase, agent systems requiring robust infrastructure and grid systems require autonomous, flexible behaviors [3, 4]. This research work aims to provide the generic framework which provides an integrated approach to leverage the power of both grid and agent technologies. Moreover, it integrates the knowledge-based component to provide intelligent decisions with justification for real time data for analysis and decision making that can be directly accessible by client applications and users. The heterogeneous and dynamic nature of the data grid environment makes it a challenging work for flexible, large-scale, coordinated service sharing among dynamic and disparate collections of individuals, institutions and resources. Agents are autonomous, adaptive and cooperative by nature and emerging as dynamic, flexible and extensible mediators for facilitating grid services in a data grid environment. The generic framework follows agent-oriented approach for modeling, designing and developing agent mediated grid services.

In a distributed computing environment, multi-agent system can be used for a variety of reasons. Having multiple agents could speed up a system's operation by providing a method for parallel computation. Another benefit of multi-agent systems is their scalability. Since they are naturally modular, it should be easier to add new agents to an existing system to add new capabilities. Systems whose capabilities and functionalities are likely to change over time or across agents can benefit from this advantage [8].

Apart from this, there is an emerging need for multi-agent systems that can manage databases of scientific and commercial applications in more generalized and implicit way. The generic framework hides the heterogeneity and provides transparent and single point access of heterogeneous & geographically distributed databases to client applications and users and handles the data access and integration through multiple collaborative agents.
Nowadays, data within the research and business organizations generally resides in multiple and heterogeneous databases. These databases may owned by different administrative domains. Therefore, both research and business organizations require these heterogeneous and geographically separated data on a real time basis for analysis and decision making. Data integration challenges have also been addressed by ETL (Extraction, Transformation and Loading) approach earlier. ETL approach comprises with three steps: extract, transform and loading. ETL approach extracts data transforms it to fit operational needs and loads it into the end target. However, the ETL approach fails to integrate the real time data due to the latency involved in cleaning, transforming and moving the data. ETL approach is having its own limitations as they fail to integrate the data in real time for which organizations generally strive. The generic framework explained in following section overcome the limitations of ETL as it accesses and integrates the data on a real time basis from geographically distributed database grid. The framework also shows how different heterogeneous data resources are integrated via grid middleware and how knowledge extraction, presentation and delivery process is performed on distributed data grid environment by using multiple task agents. It allows combining different types of information as a single entity to gain a more complete scenario and to integrate the similar types of information about different entities. The detailed view of the architecture of the generic framework for integration of multi-agent knowledge-based system and distributed database grid is presented in Figure 3.2.

The Figure 3.2 demonstrates the generic framework for integration of multi-agent knowledge-based system and distributed database grid. The framework consists of four layers: Grid Fabric Layer, Middleware Layer, Knowledge Layer and Application Layer. Each layer is separated by a dotted line. The layers are connected in a bidirectional way. The grid fabric layer is the lowest layer of the generic framework. Middleware layer is built upon the grid fabric layer. The middleware layer provides the core and user level services which are wrapped up in agents. Different collaborative agents are contained by middleware layer which are responsible for realization of requests made by the client applications and users or upper level layer. The knowledge layer is built upon the middleware layer. The knowledge layer consists of knowledge-based component which may implement through artificial intelligence techniques. This will turn the data into something more than just a collection of data. The services offered by the knowledge layer used for analysis and decision making. Knowledge-based systems are providing a new
3.1.2 Communication Layer

The communication layer is built upon the data layer. This layer includes the communication and authentication protocols required for data grid transactions such as exchange of data between data resources and verification of the identity of users and several resources. This layer also contains connectivity protocols used to accomplish the activities and services such as enabling secure initiation, resource monitoring, control of resource-sharing operations and many more.

3.1.3 Information Layer

The information layer rests above the communication layer. It provides the data access and integration services. It is also responsible for data delivery and data transformation. Apart from this, it offers some core and user level services like metadata management, security, grid information service and so forth. It allows the sharing and management of data.
Bepperling A. et al. [5] proposed the integration of web service and a multi-agent system (MAS) in order to build a control architecture which is suitable for automated reconfigurability. Integration of the concept of Service Oriented Architecture (SOA) through web services into the factory automation domain makes it possible to abstract and encapsulate the functionality of the devices as services. Aaron Welch et al. [2] proposed to apply a service-oriented approach to data collection for a generalized multi-agent system. Here, multiple services may easily request to work on different sets of data independently. An appropriate SOA implementation allows for a limitless number of unique services to operate on exactly what agent information end users are interested in, without any necessary prior anticipation of particular services.

A. M. Riad, H. and A. El-Ghareeb [1] described how to utilize SOA to integrate web services and software agents in Course Management Systems (CMS). They highlighted the unlimited advantages of web services and its capabilities to facilitate software agents’ integration within information systems. Victor Pankratius et al. [48] presented a service-oriented architecture of Learning Content Management System (LCMS). It was based on web services, which was extended to support software agents. Here, for the distributed retrieval of LOs (learning Objects), agents were used for personalized searches according to user specifications. In this scenario, agents were also able to query metadata inside web services. They have used the web services to implement an entire LCMS functionality including storage and retrieval of learning content.

2.2.3 Agent Enabled Grid

Agent technology is increasingly used in grid systems. Also, there are many projects which have been built based on grid, web service and agent technologies.

Ian Foster, Nicholas R. Jennings and Carl Kesselman [18] discussed why grid and agents need each other. They reviewed the current state of the art in both areas: grid and agent systems. They have also reviewed the challenges that concern the two communities and proposed research and technology development activities that can allow for mutually supportive efforts. As Ian Foster and some other people pointed [18, 20], the grid provides the infrastructure of sharing large scale datasets and solving complex tasks. For that, it needs a flexible management for node resources and autonomous agents have this ability by coordinating with each other. Apart from this, the grid community has historically focused on “brawn”: interoperable infrastructure and tools for secure and reliable resource
sharing within dynamic and geographically distributed Virtual Organizations (VOs), and applications of the same to several resource federation situations. In contrast, those working on agents have focused on “brains”: the development of methodologies, theories and algorithms for autonomous problem solvers. These problem solvers can act flexibly in uncertain and dynamic environments in order to achieve their aims and objectives [10]. As the scale and ambition of grid and agent deployments increase, agent systems requiring robust infrastructure and grid systems requiring autonomous, flexible behaviors [18]. For that, the new architecture is required to combine both grid and agent technologies.

A grid service is a web service that complies with a set of conventions that defines how a client interacts with a grid service. These conventions, and other Open Grid Services Architecture (OGSA) mechanisms associated with grid service creation and discovery, provide the controlled, fault resilient, and secure management of the distributed applications.

**Research on Agent Enabled Grid:** Yufei Bai et al. [52] developed an agent-based spatial query execution mechanism. They introduced an agent model for cooperating complex spatial query optimization task in a grid environment.

Saleh AlZahrani [38] focused on the role of multi-agent system in regionally distributed architecture for dynamic e-learning environment (RDADeLE), its components and regional grid architecture. Most of the business and academic institutions and training centers around the world have adopted the grid technology in order to create, deliver and manage the learning materials through the web. It contains the specification and implementation of agents but incorporating the OGSA-DAI in order to produce grid services is remaining to be performed in the future.

### 2.3 Multi-agent Knowledge-based System by Integrating Fuzzy Logic

Knowledge-based System (KBS) is one of the branches of artificial intelligence. They are usually embedded in other applications to perform knowledge specialized tasks. Knowledge-based systems are the result of a long investigation process performed by artificial intelligence scientists.

Claudiu Ionuț Popirlan et al. [7] demonstrated the use of intelligent software agents in knowledge-based systems. Difficult tasks need to be decomposed into smaller sub-tasks
and each sub-task should be solved with the most appropriate reasoning technique without increasing the execution time. They introduced intelligent software agents for data analysis and finding the expert user (operator) in a contact center environment. P.R. Vundavilli et al. [30] developed the expert system using fuzzy logic for abrasive water jet machining (AWJM) process. It is to be noted that the performance of AWJM in terms of depth of cut depends on various process parameters such as diameter of focusing nozzle, water pressure, abrasive mass flow rate and jet traverse speed.

Fuzzy logic was conceived in the USA by Prof. Lotfi A. Zadeh in the early 1965's [53]. Abdur Rashid Khan, Hafeez Ullah Amin and Zia Ur Rehman [3] depicted adaptation of expert systems technology by using fuzzy logic to handle qualitative and uncertain facts in the decision making process. They have observed that, human behaviors are mostly based upon qualitative facts and cannot be numerically measured. Also, it has been hardly to decide correctly. For that, they proposed an application of an expert system with fuzzy logic in the teacher’s performance evaluation. C. C. Yee and Y. Y. Chen [6] proposed a performance appraisal system using a multifactorial evaluation model which deals with appraisal grades which are often expressed vaguely in linguistic terms. The proposed model was used for evaluating staff performance based on specific performance appraisal criteria. Also, it was in collaboration with one of the information and communication technology companies in Malaysia with reference to its performance appraisal process.

S. Maria Wenisch, G.V. Uma and A. Ramachandran [35] have shown an integrated and holistic approach to knowledge management system for natural resource management needs to take local indigenous knowledge as one of its components for achieving sustainability. The paper builds a fuzzy inference system from the fuzzy indigenous knowledge system on soil. The inference rules are framed from the fuzzy indigenous knowledge on soil as IF...THEN structures. The relationships between various factors influencing the suitability of soil for crops are produced as the output of the suitability fuzzy inference system.

2.4 Motivation to Develop Multi-agent Knowledge-based System Accessing Distributed Database Grid

Several research and development projects have been carried out so far in the area of multi-agent systems, knowledge-based systems and data grid. Also, there are many
projects which have been built based on grid, web service and agent technologies. The following limitations have been observed in general.

Distributed database systems are having its own limitations as stated in section 2.1.2 of this chapter. In this section, we mentioned the need to integrate databases in grid environment in order to develop the distributed database grid as many scientific and commercial applications strives for mechanisms to access and integrate multiple and distributed collections of structured data. This data needs to be made available and accessible to distributed set of users and their applications, which makes these applications ideal candidates to adopt grid-based infrastructure.

Apart from this, agent technologies have proven their excellence in order to solve large and complex problems, specifically in the area of distributed computing. Therefore, there is a need arising to develop an agent-enabled distributed database grid system. Moreover, organizations generally use the structured data for analysis and decision making purposes and they need some mechanism for knowledge extraction, presentation and delivery.

To achieve the above said goals, there is a need to design and develop a generic framework to access and integrate different heterogeneous data resources via grid middleware by incorporating multiple collaborative agents. Also, knowledge extraction, presentation and delivery process should apply on these data sets to turn this data into something more than just a collection of data.

Agent technology is being used increasingly in grid systems. As we have discussed above, the grid provides the infrastructure of sharing large scale datasets and solving complex tasks, but it needs a flexible management for node resources. On the other side, autonomous agents have the ability to coordinate with each other to provide dynamic, open and flexible environment [19]. As a result, there are several advantages have been observed to integrate the agent technology with grid systems.

Moreover, as stated earlier also in this chapter, the grid communities have historically focused interoperable infrastructure and tools for secure and reliable resource sharing within dynamic and geographically distributed Virtual Organizations (VOs) [18, 20] and agent communities have focused on the development of methodologies, theories, models and algorithms for autonomous problem solvers that can act flexibly in uncertain and dynamic environments in order to achieve their aims and objectives [10]. Moreover, grid
systems require independent, flexible behaviors and agent systems require robust infrastructure [18]. Therefore, this research work is aimed to provide an integrated approach to impart multi-agent and knowledge-based systems technology in the data grid to leverage the power of both grid and agent.

This research produces the novel dynamic, flexible and extensible framework based on both agent and grid technology. The developed framework shows how different heterogeneous data resources are integrated via OGSA-DAI and how knowledge extraction, presentation and delivery process is performed in a distributed data resource environment by using multiple task agents. The developed framework hides the heterogeneity of the database nodes from the client applications and handles the data access and integration in an efficient manner. An agent based interface is provided to the client applications to access the middleware in a platform and language independent way. The agents support the integration and sharing of heterogeneous databases. The multi-agent system embedded in the data grid is having many features like coherence, interoperability and scalability which strengthen the whole architecture. The agents of developed generic framework are used to expose heterogeneous data resources from and to a grid via OGSA-DAI.

To provide the knowledge-based and human oriented assistance, fuzzy set theory and rules are used. The developed framework also 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. The fuzzy expert system modeled by incorporating agents is developed on the top of the distributed database grid. It is aimed to accommodate the knowledge and expertise with reasoning capabilities that will provide a great support to executives for decision making in domain specific research and commercial applications.

2.5 Conclusion

This chapter represents the survey of literature conducted for carrying out the research work. Data grids basically deal with providing services and infrastructure for distributed data intensive applications. Agent technologies have proven their excellence in order to solve large and complex problems, specifically in the area of distributed applications. Therefore, there is a need arising to develop agent-enabled distributed database grid architecture. This chapter demonstrates the details about data grids, databases and data
grid, grid application systems, grid middleware, multi-agent & knowledge-based systems, agent enabled grid and so forth. It also focuses on available applications and research work so far done in the above area. After studying and analyzing the current scenarios in above mentioned areas, this chapter also explains about a motivation for the development of multi-agent knowledge-based system accessing distributed database grid. In the next chapter, we will discuss about the generic framework in detail. The developed generic framework offers a variety of services and activities to client applications and users. The next chapter will discuss about such services and activities in detail.

References


