A Collaborative e-Science Architecture for Distributed Scientific Communities

by

Tran Vu Pham

Submitted in accordance with the requirements for the degree of Doctor of Philosophy

The University of Leeds
School of Computing

October 2006

The candidate confirms that the work submitted is his own and that the appropriate credit has been given where reference has been made to the work of others.

Acknowledgements

I would like to express my very special thanks to my supervisors - Dr Lydia Lau and Professor Peter Dew - for the invaluable guidance, advice, help and encouragement I have received for this work.

I would like to thank Professor Michael Pilling, an expert in the domain of reaction kinetics, combustion and atmospheric chemistry from School of Chemistry. With his enthusiasm, he has provided very useful input to the case study of this work.

Many thanks to other chemists from School of Chemistry for their help on the case study. Especially, thanks to Dr Kevin Hughes for his help on the building simulation and analysis services for chemical reaction mechanisms. Many thanks to Dr Andrew Rickard, Dr Lisa Whalley and Jenny Stanton for their participation in the evaluation of the Collaborative e-Science Architecture.

Many thanks to my friend and also my companion, Dr Mohammed Haji, for the useful discussions and encouragement I have received during my time at the School of Computing.

Last of all, I would like to acknowledge the support from my beloved wife, who is about to give birth to our first son, and my entire extended family. They have been giving me endless support and encouragement to complete this work. To whom, saying thanks would never be enough.

Abstract

Modern scientific research problems are getting more and more complicated. Addressing these problems require knowledge and expertise from a wide range of scientific disciplines. The instruments required for modern scientific research problems are also complex and expensive. In addition, the amount of research data generated by experiments on these problems is getter bigger to an extent that might not be manageable by any individual organisations. All of these factors have made global distributed collaborations become increasingly important in modern scientific research. Dealing with distributed collaborations at such a large scale has given rise to a new subject called e-Science.

Grids have been widely accepted as promising infrastructures for e-Science. Grids enable the sharing of large-scale computational resources and experimental datasets in distributed virtual organisations. Web-based collaborative portals are commonly used as environments for interactions amongst distributed collaborators. Collaborators in a Web-based environment are subject to certain level of centralised administration and control. Their interactions have to be routed through a central server. This has been seen as inflexible and does not scale well with respect to the heterogeneity of distributed user communities.

This thesis reports an investigation on a Collaborative e-Science Architecture (CeSA), which is an integration of Grid and Peer-to-Peer computing infrastructures using service oriented architecture, for supporting distributed scientific collaborations. CeSA leverages the advantages of Peer-to-Peer computing in supporting direct collaborations amongst end users and the capability of providing large-scale computational resources and experimental datasets. The investigation addressed two important issues with regard to the CeSA: (i) usability of the CeSA from users' point of view and (ii) an efficient resource discovery mechanism for the Peer-to-Peer environment.

The usability was evaluated using the reaction kinetic research group in Leeds as a case study. An instance of the CeSA was prototyped for the evaluation. Feedback collected from the users was positive.

An adaptive resource discovery approach has been introduced for the P2P collaborative environment of the CeSA. This adaptive approach takes into account the resource distribution and characteristics of scientific research communities. A learning mechanism, based on a classification of user interests using ontology, is used to adaptively route search queries to peers which are most likely to have the answers. Simulation results showed that this approach can efficiently improve query hit rates and also scale well with the increasing of network populations.

Declarations

Some parts of the work presented in this thesis have been published in the following articles:

Pham, Tran Vu; Dew, Peter M.; Lau, Lydia M. S.; Pilling, Michael J. (2006) Enabling e-Research in Combustion Research Community in: *The 2nd IEEE International Conference on e-Science and Grid Computing Workshops*, Amsterdam December 2006, IEEE Computer Society Press. (to appear)

Pham, Tran Vu; Lau, Lydia; Dew, Peter (2006). An adaptive approach to P2P resource discovery in distributed scientific research communities in: *Sixth International Workshop on Global and Peer-to-Peer Computing (GP2P) in conjunction with IEEE/ACM International Symposium on Cluster Computing and the Grid 2006.*

Pham, Tran Vu; Lau, Lydia M. S.; Dew, Peter M.; Pilling, Michael J. (2005) Collaborative e-science architecture for reaction kinetics research community in: *Proceedings of the Challenges of Large Applications in Distributed Environments Workshop (CLADE2005)*, pp. 13-22 IEEE Computer Society Press.

Pham, Tran Vu; Lau, Lydia M. S.; Dew, Peter M.; Pilling, Michael J. (2005). A collaborative e-Science architecture towards a virtual research environment in: S J Cox & D W Walker (editors) *Proceedings of the 4th UK e-Science All Hands Meeting (AHM'05)*, EPSRC.

Pham, Tran Vu; Lau, Lydia M S; Dew, Peter M. (2004). The integration of grid and peer-to-peer to support scientific collaboration in: Michaelides, D & Moreau, L (editors) *Proceedings of GGF11 Semantic Grid Applications Workshop*, pp. 71-77.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Distributed collaborations are becoming increasingly important in modern research. As research problems are getting increasingly complex, there is increasing need for a wide range of highly specialised expertise for interdisciplinary research to address these complex problems (Katz & Martin 1997, Lee & Bozeman 2005). The volume of scientific data required for solutions to these complex problems is getting bigger, to a size that might not be manageable by any individual organisation. It was expected that the Large Hadron Collider (LHC), based at CERN, will produce petabytes of data each year for each experiment, when operational (Hey & Trefethen 2002). In climate research, a single model run on an atmospheric model can easily generate tens of terabytes of data (Office of Science - U.S. Department of Energy 2002). In the report by US National Research Council in 1993, the doubling time for the body of scientific information was 12 years (National Research Council 1993). Scientific instruments required are also increasingly expensive, while research funding for scientists is getting tighter (National Research Council 1993). Therefore, expensive resources have had to be pooled at a regional, national or international level (Katz & Martin 1997). Again, the LHC is a typical example of this case.

In addition, collaborations will result in faster advancements and higher research quality (Kraut et al. 1986). As two or more scientists get involved in a collaborative research project, the research quality can be cross-monitored during the process. Through collaboration, skills and expertise can be transferred amongst scientists involved (Katz & Martin 1997). Lee & Bozeman (2005) also showed that collaboration could also improve productivity of research work.

There are also political reasons for research collaborations, especially when collaboration across institution is used as criteria for funding. A particular example is from the European Commission, which requires researchers to seek collaborative partners before applying for financial support (Katz & Martin 1997).

All of the above factors have made the collaborations across disciplines and across institutions become vital in modern research. Thus, promoting and supporting scientific collaborations are becoming increasingly important.

1.2 The Challenge

A number of programmes and projects have been set up to promote and support scientific collaboration worldwide. In the UK, the e-Science programme was started in 2000 by the Research Councils UK (NeSC 2006). In 2004, the Joint Information Systems Committee (JISC) started the Virtual Research Environments Programme (VRE 2006). Most recently, JISC has announced e-Infrastructure Programme, which will begin in September 2006 (Farnhill, James 2006). In the US, similar programmes, such as National Collaboratories (since 2001) (DOE - Office of Science 2005) and Cyberinfrastructure (since 2003) (Atkins et al. 2003) have also been started. The European Commission has also got involved in these activities by funding a number of projects such as “Enabling Grids for e-Science” (EGEE 2006) and DataGrid (The DataGrid Project 2006). In Japan, the Earth Simulator Center have also involved in a number of collaboration projects in Earth Science using the Earth Simulator super computer (ESC 2006). There are even more projects at institutional and organisational levels.

The kinds of collaborations addressed by these programmes and projects include: (i) the sharing of very large scale data collections and high performance computing resources, such as available storages and CPU cycles, (ii) the bringing back access to high performance visualisation to scientific research communities, and, (iii) the collaborative activities amongst individual scientists, such as the sharing of day to day working data, working papers or even just a chat message to inform others about the availability of an interested paper.

Grids have widely been accepted as a key infrastructure for sharing and linking high-end resources in these programmes and projects. Web services, with the capability to provide flexible integration and interoperability amongst distributed applications, have also been adopted by the community as means for delivering resources within the grid environment. Accessing to grid resources is made possible through portals via Web services.

Collaborations amongst individual scientists are quite often supported by Web-based collaborative portals. Examples are British Atmospheric Data Centre (BADC 2006) and Collaboratory Multi-Scale Chemical Science (CMCS 2005). A scientist can gain access to a collaborative portal from anywhere with a simple Web browser. Other applications, such as visualisation tools, can also be installed on the Web server to provide users with greater capability.

However, the support for collaborations from Web-based collaborative portals is indirect. All the collaborations have to be done over resources held by third party servers, as shown in Figure 1.1. This collaboration model has its own limitations. Firstly, it lacks of the support for cross community collaborations. This kind of collaborations is common in scientific research communities, where multidisciplinary research is usually the case. Secondly, it is the inflexibility to support distributed collaborations in distributed loosely coupled communities as every collaboration activity has to be done via the central server (Tian et al. 2003). Thirdly, common critiques about traditional Web-based architecture, the underlying architecture or Web-based collaborative portals, where a single Web server application serves many Web clients, are susceptible to single-point of failure and scalability problem. When the workload increases, the Web server becomes the bottleneck (Liu & Gorton 2004). Other factors such as control and sense of ownerships over shared resources may also be issues of centralised approaches.

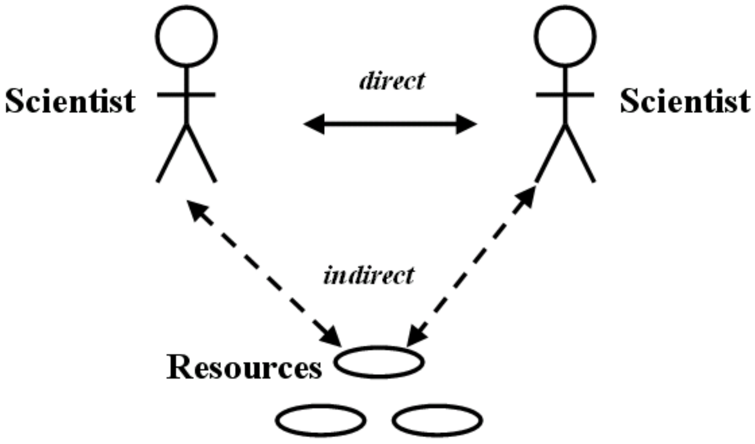


Figure 1.1: Direct and indirect support for collaborations

The challenging problem is how to sufficiently support collaborations in distributed scientific communities. “Researchers must have access to useful computer facilities, networks, and data sets but must also be able to work in an environment that fosters cooperation amongst individuals with differing academic traditions, approaches to and priorities in research, and budget constraints” (National Research Council 1993). The kinds of collaboration that need to be addressed have to be able to enable the sharing of computational instruments amongst research institutions as well as information and ideas amongst individual scientists. The integration of Grid computing and Web-based collaborative environment using Web services can support the collaborations to a certain extent. However, the use of Web-based architecture limits scientific collaborations from its full potential.

1.3 The Potential from Peer-to-Peer Computing

Peer-to-peer (P2P) is popularised by many desktop file-sharing applications such as Napster (Shirky 2001), Kazaa (Kazaa 2006) and eMule (eMule 2006). Although P2P file sharing applications have also been blamed for supporting violation of copyright laws by the movie industry, with a proper use, P2P also has other potential in addition to desktop file sharing. For instance, it has been used for Internet phone system (skype 2006), for distributing services to a community (GSC-Chinook 2006) and for collaborative teamwork (Groove Networks 2006).

P2P is a decentralised computing model, in which peer applications can directly communicate with each other without going through any third party server. It is able to support direct collaboration between scientists, shown as direct collaboration in Figure 1.1. This is the key characteristic that makes P2P different from Web-based architecture. The ability to provide direct communication allows users in P2P environment to dynamically and autonomously establish their own communities without being regulated by any third party administration. Cross community communication and, hence, collaboration are made easier. Users of P2P application can share resources directly from their computers. Hence the sense of ownership over the shared properties is maintained. Users can also revoke any resource from sharing at anytime. Furthermore, P2P applications often provide means for real-time communications, such as instant messaging or internet phone, which are highly suitable for direct collaborations amongst distributed scientists. On the technical aspect, as P2P is decentralised, where computation is taken place at the edges, it is more scalable when the number of users increased. The bottleneck problem can also be avoided. Single-point of failure never exists in P2P.

The above characteristics show that P2P computing model can potentially be employed to develop a better collaborative environment for supporting distributed scientific collaborations. It could be a complement to Web-based architecture and Grid computing.

1.4 Research Objectives

The focus of this research is on an investigation into use of a P2P based collaborative environment on top Grid computing resources to support distributed collaborations amongst scientists. The overall aim is to develop a collaborative e-Science architecture using a combination of the Grid and P2P computing together with other distributed computing technologies, such as Web services, to address the current limitations of Web-based architecture. In order to meet this goal, following objectives need to be achieved in this research:

(i) To understand the characteristics of and requirements for distributed collaborations within scientific communities. These characteristics and requirements will be helpful for a better understanding of the problem domain under study. They form the basis for the collaborative architecture to be developed.

(ii) To have a detailed specification of the collaborative e-Science architecture. The specification needs to clearly specify how a P2P environment is integrated with Grid computing resources. It also provides in detail technologies involved in the integration. Functional components and the relationships amongst these components also need to be specified.

(iii) To get an insight into the usability of the proposed architecture within potential user communities. This is the key issue of any collaborative system. It is the users who will eventually decide the success of a collaborative system.

(iv) To have a suitable resource discovery method for the P2P collaborative environment. As P2P is a decentralised architecture, resource discovery is always an important issue. There are a number of resource discovery methods that exist for P2P. However, the scientific communities have distinctive characteristics and requirements for resource discovery from other social communities. Therefore, it is necessary to have an investigation on a suitable method for the P2P collaborative environment of the architecture.

Other technical issues such as security and connectivity are always important to any distributed computing system. They are also important issues for the collaborative e-Science architecture to be developed. However, in this research the priority is given to the functional aspects of the collaborative architecture. Once the functionality of the architecture has been understood, further study will address other issues in incremental manner.

1.5 Research Questions

To achieve the above objectives, the following questions need to be answered.

Q1. What are characteristics of scientific collaborations? What are the requirements for a collaborative system to efficiently support collaboration in distributed scientific communities?

Q2. How a P2P environment can be integrated with Grid computing resources in a collaborative e-Science architecture in order to efficiently support collaborations in scientific communities?

Q3. How potential users react to functionalities provided in the new collaborative architecture, in terms of supporting their day-to-day collaborative activities?

Q4. What constitute an efficient resource discovery method for the P2P environment of collaborative e-Science architecture? What is a suitable one?

1.6 Research Methodology

Methodology and method might be used to mean different things in literature (Mingers 2001). In the context of this thesis, *research methodology* is referred to "a combination of the process, methods, and tools which are used in conducting research" (Nunamaker & Chen 1990). A *research method* is a "particular activity" such as analysing a survey or conducting a controlled experiment to do research (Mingers 2001).

In order to answer research questions, a combination of different research methods are used in this research. The main body of the research methodology is system development, which has been recognised as a research methodology (Nunamaker & Chen 1990). The result of the development process provides concrete objects for evaluation.

1.6.1 System Development

System development is applied for specification of the collaborative architecture (question Q2). It is an iterative process. The result of an earlier iteration is used as input for the next iteration until a satisfactory system is achieved. An iteration consists of the following activities:

- Identify objectives and requirements
- Design system architecture
- Develop prototype system
- Evaluate the prototype system

This incremental approach is used in order to identify and resolve any possible risks that may occur during system development process such as technology constraints.

1.6.2 Quantitative and Qualitative Evaluations

Quantitative and qualitative are two common classes of methods for evaluation in research. Quantitative methods rely on statistics and controlled experiments. Quantitative methods are difficult in studies undertaken within a social context as there are many uncontrolled variables and they are not always quantifiable (Kaplan & Duchon 1988).

Qualitative methods, on the other hand, are based on observation and understanding of phenomenon in the context of study. Qualitative methods provide less explanation of variances in terms of statistics but can yield richer interpretation of phenomenon under study. Qualitative approach is preferable in behavioural research (Kaplan & Duchon 1988).

This research uses both qualitative and quantitative for two different purposes. Qualitative approach is used for evaluation of the collaborative architecture in a potential user community (question Q3). Quantitative approach is for evaluating the performance of resource discovery methods in P2P environments in order to find a suitable one (question Q4). The following are the two methods used:

i. ***Case Study***: Case study is a popular qualitative method. It is suitable for addressing research questions of type why or how (Yin 1994). In this research, a case study based on interviews and questionnaires is used to get an analysis on potential users' reactions on the functionality provided by the proposed collaborative architecture. Case study also helps to clarify characteristics and requirements of scientific collaborations (question Q1).

ii. ***Experiment by Simulation***: Simulations are used to evaluate and analyse performances of candidate P2P resource discovery methods for the collaborative architecture. The evaluations and analyses are based on quantitative data collected during the simulations.

A summarisation on different methods used to address the research questions is shown in Table 1.1.

	Methods			
Questions	**Literature Review**	**Case Study**	**System Development**	**Experiment by Simulation**
Q1	X	X		
Q2			X	
Q3		X	X	
Q4	X	X	X	X

Table 1.1: A summary on methods used to address research questions

As shown in the Table 1.1, answering a research question may involve a number of different methods. For example, answers for question Q1 can be found from research literature and case study (by interviewing potential user communities). A combination of case study and system development (for a system prototype) is necessary to answer question Q3. Answers for question Q4 require a range of methods from literature reviews (for requirements and potential approaches), case study (for requirements) and system development (for prototype developments) as well as simulations.

1.7 Thesis Outline

The next chapter, Chapter 2, is a review on research literature on collaboration technologies. It first reviews on characteristics of scientific research collaborations and their requirements for supporting infrastructure. Then the review focuses on the current supporting information technology infrastructures for scientific research collaborations.

Chapter 3 discusses the current limitations in supporting scientific collaborations and motivation for a new architecture. It then provides a detailed description on the development of the Collaborative e-Science Architecture.

Chapter 4 presents a case study. In the case study, the Reaction Kinetics research community is described as a typical scientific research community. The community is used to illustrate characteristics and requirements of scientific research communities to be identified in Chapter 2. These concrete requirements will then be used to develop a system prototype and to evaluate the proposed architecture based on the prototype in subsequent sections. The latter part of this chapter provides details on an experiment and evaluation of the architecture using the prototype system.

In Chapter 5, technical challenges that need to be dealt with in order to successfully implement the proposed architecture are identified. Resource discovery in distributed and decentralised P2P environment is identified as one of the challenges. A proposed solution, based on the use of classification ontology, to resource discovery problem will be discussed. Details of experiments on the proposed solution and experimental results will also be provided.

Chapter 6 concludes this thesis by summarising the research findings and major outcomes of this project. The reflection on what have been done on the project and potential areas for future will also be discussed.

Chapter 2

Technologies for Supporting Distributed Scientific Collaborations

This chapter is a background review on technologies for supporting distributed scientific collaborations. It firstly focuses on key characteristics of modern scientific collaborations. These characteristics should ideally form the requirements for supporting technologies. The second section of this chapter discusses briefly various types of technologies for supporting scientific collaborations, ranging from infrastructures such as Grids to basic communication tools such as instant messengers. A number of related projects for supporting distributed collaborations are also reviewed in the section follows.

2.1 Scientific Collaborations

Collaboration started to appear in scientific community in the 17th and 18th century when the community turned into professionalisation as means of gaining and sustaining recognition and advancement in professional hierarchy (Beaver & Rosen 1978). The traditional form of collaboration is co-authoring of research work and publication. This basic kind of collaboration has been used as measurement to study the structure of scientific collaboration networks (Newman 2001*a*, Newman 2001*b*, Newman 2001*c*) as well as to assess the level of collaboration within scientific communities (Beaver & Rosen 1978, Beaver & Rosen 1979*a*, Beaver & Rosen 1979*b*, Katz & Martin 1997).

In modern scientific research, as explained in Section 1.1, the collaborations go beyond co-authoring activities, although this form of collaborations is still popular. It involves the sharing of complex and expensive equipments amongst distributed research institutions. This is a result of the increasing complexity of research problems, which require complex and expensive instruments that no single research institution can afford (Kraut et al. 1986, National Research Council 1993, Katz & Martin 1997, Lee & Bozeman 2005). Resolving complex research problems also involves huge amount of experimental data and computationally intensive applications. In addition to instruments and data, gathering a wide range of highly specialised expertise for interdisciplinary research problems is also an important characteristic of scientific collaborations. (Katz & Martin 1997, Lee & Bozeman 2005)

Scientific collaborations are now happening at a global scale. One such example comes from research in particle physics. Each experiment conducted on the LHC will involve a collaboration of over a hundred institutions and over a thousand of physicists from Europe, USA and Japan(Hey & Trefethen 2002). Another example is the combustion research community. A consortium from the combustion community is building an infrastructure for promoting collaborations across Europe and the US (PrIMe 2006).

Although scientific collaborations are important in modern scientific community, competitions also exist within the community, due to the desire for social recognition (Hagstrom 1965). Competition has two contradicting effects on collaborations. On one hand, it motivates scientific researchers to collaborate to increase research productivity. On the other hand, it may deter collaborators from sharing knowledge to maintain their competitive edges. Lacking of a proper protection of their personal knowledge may keep scientists away from collaborations.

Informal communication has a very important role in scientific research collaborations (Hagstrom 1965, Edge 1979, Kraut et al. 1986, Kraut, Egido & Galegher 1990). Informal communications can bring scientists with the same or similar research interest together. This creates opportunities for new research collaborations. The frequency of informal communication can help to maintain the threads of a collaborative relationship over time. Kraut et al. (1986, 1990) also showed that physical proximity has direct influence on the quality of informal communication. As a consequence, physical proximity has great influence on the scientific research collaboration.

In a summary, today's scientific collaborations have the following common characteristics:

- The collaborations involve the sharing of complex and expensive research instruments and huge volume of data.

- Knowledge and expertise from different disciplines are required for tackling big complex interdisciplinary research problems.
- The collaborations happen not only within the boundary of a particular institution but also at a global scale.
- There exist competitions amongst collaborators for social recognition, although collaborations are necessary to improve research productivity.
- Informal communication has an important role in collaboration process.

Ideally, technologies that are designed to support scientific collaborations need to support these characteristics. They have to be able to enable the sharing of research instruments, such as computational capability, network and storage, and research datasets in huge volume. The supporting technologies also need to facilitate the sharing of knowledge and expertise across disciplines at a global scale. However, in order to encourage scientists involved in the collaborations, the technologies should also be capable of protecting their personal resources during the collaboration processes. As informal communication has an important role in supporting collaborations, the collaborations should exploit this characteristics.

2.2 Collaboration Technologies

Collaboration technologies are referred to as technologies that support collaboration activities amongst people from distributed locations. The technologies reviewed in this section include those that have been used for or those that are capable of supporting various aspects of scientific collaborations discussed in Section 2.1. They include technologies that enable the sharing of back-end computational resources and large datasets such as Grid computing. The discussion also includes technologies for end user interactions such as communication tools (e.g. video phone, email and instant messengers), teamwork coordinating tools (e.g. group calendars) and collaborative environments (e.g. Web-based environments and P2P environments).

2.2.1 Service Oriented Architecture

Generally, Service Oriented Architecture (SOA) refers to "a style of building reliable distributed systems that deliver functionality as services, with the additional emphasis on loose coupling between interactive services", in which a service is "a software component that can be accessed via a network to provide functionality to a service requester" (Srinivasan & Treadwell 2005). A service is usually a business function, implemented in software, wrapped with a formal documented standard interface. It could be accessible through the interface using standard messaging protocols (Papazoglou 2003). The internal properties of a service are encapsulated.

2.2.1.1 The Basic Service Oriented Architecture

The basic SOA defines three kinds of participants: service provider, service client and service discovery agency with three operations: publish, find and bind for interactions amongst the participants as shown in Figure 2.1 (Papazoglou 2003).

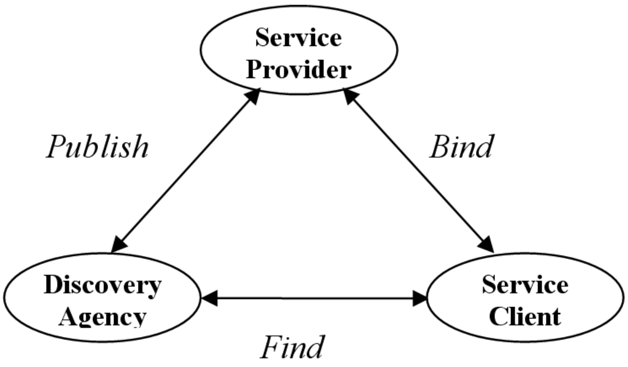


Figure 2.1: Basic Service Oriented Architecture

- *Service providers*: are software agents that provide services to others. Service providers are responsible for *publishing* description of their services through *service discovery agencies*.
- *Service clients*: are software agents that request for execution of a service. A service client needs to *find* information about services of its interest through *service discovery agencies* and then *bind* with the *service provider* which provides the service for execution.
- *Service Discovery Agencies*: hold registries of published services and help service clients to locate their services of interest.

A more market oriented view of SOA described by De Roure, Jennings & Shadbolt (2003), in which service owners (providers) interact with service consumers (clients) in marketplaces owned by market owners. The role of marketplaces in this view corresponds to the role of discovery agencies in the basic view of SOA. Market owners set up rules to govern interactions between service consumers and service providers in their marketplaces. Once a service consumer and a service owner agree on a particular service, they bind together in a service contract.

2.2.1.2 The Extended Service Oriented Architecture

The extended SOA adds in additional composition and management layers on top of the basic SOA as depicted in Figure 2.2 (Papazoglou 2003, Papazoglou & Georgakopoulos 2003).

Service composition layer, in the middle of the extended SOA deals with composing basic services, with limited capabilities, into composite services, with advanced functionality, to meet specific application requirements. The functionalities that the composition layer contributes to the extended SOA include service coordination, monitoring, conformance and quality of service (QoS) composition.

On top of the extended SOA, service management layer provides functionalities that serve two purposes: to manage the service platform, deployments of services and their applications and to provide support for open service marketplaces. For instance, in supporting the applications, the service management may provide application performance statistics that support assessment of application effectiveness. In terms of supporting the marketplaces, it may create opportunities for service consumers and service providers to meet and conduct business.

2.2.1.3 Benefits of Service Oriented Architecture

The loose coupling feature of SOA offers great values to applications in distributed environments. Services can be flexibly integrated into applications, once their interfaces and locations are discovered. The internal architecture of a service could be replaced or updated without the need of changing the integrated applications, which are using the service, as long as the service interface is preserved. If a service that an application is using fails to function, it will be easy to locate another service with the same capability and interface to replace the faulty service. Hence, SOA based applications are more fault tolerant.

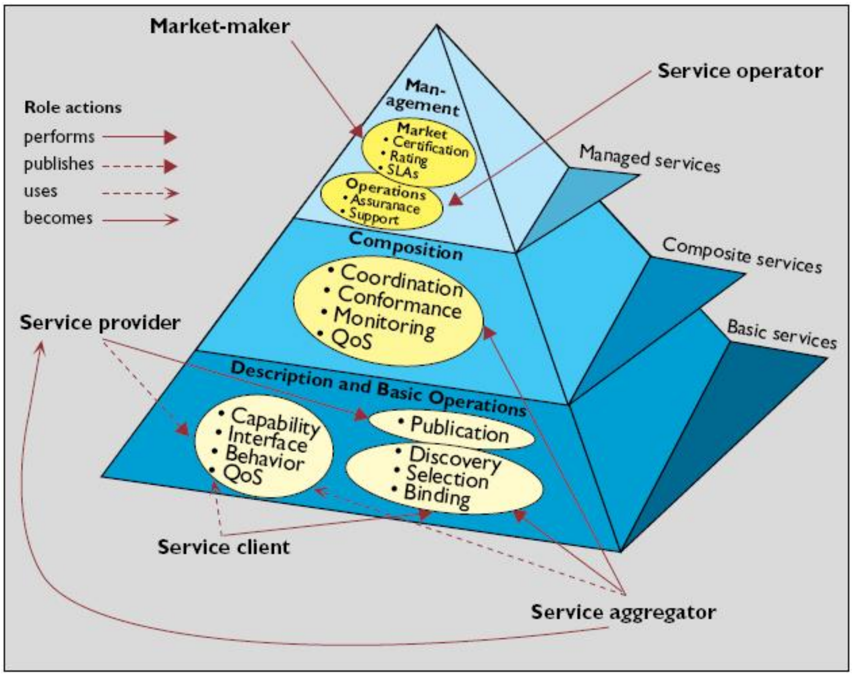


Figure 2.2: Extended Service Oriented Architecture
(Papazoglou 2003, Papazoglou & Georgakopoulos 2003)

2.2.2 Web Services

Web services are the most well-known implementation of SOA. Web services create a new paradigm for distributed application integration by offering more flexibility and interoperability, which is an important requirement for distributed application integration in heterogeneous environments (Pierce et al. 2002).

Web services are "self-contained, modular business applications that have open, Internet-oriented, standards-based interfaces" (UDDI Consortium 2001). This definition stresses on *Internet-oriented* and *standard-based interfaces* to ensure that Web services are flexible and interoperable in distributed environments. A more precise definition used by the W3C Web services working group, which links Web services to associated enabling technologies, to guarantee their capability (W3C Web Service Architecture Working Group 2004):

> "A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunc-

tion with other Web-related standards"

The definition quoted identifies key enabling technologies for Web services:

- eXtensible Markup Language (XML): offers a standard, flexible and extensible data format for serialization of data

- SOAP: provides a standard, extensible and composable framework for packing and exchanging XML messages.

  SOAP originally was an acronym for Simple Object Access Protocol, which is about remote procedure calls. However, the current use of SOAP in the context of SOA does not reflect the meaning of its origin. In SOA, its interpretation is extended to Service Oriented Architecture Protocol. A SOAP message in SOA contains information needed to invoke a remote service or results of a service invocation (W3C Web Service Architecture Working Group 2004).

- Web Services Description Language (WSDL): provides a model and an XML format for describing Web services (Chinnici et al. 2003).

2.2.3 The Semantic Web

The Semantic Web is an extension to the current Web, in which information is given well defined meanings, better enabling computers and people to work in cooperation (Berners-Lee et al. 2001, Hendler et al. 2002). Three basic components of the Semantic Web are ontology, Resource Description Framework (RDF) and agent computing.

2.2.3.1 Ontologies

An ontology is formally defined as "an explicit specification of a conceptualisation" (Gruber 1993). In this definition, a conceptualisation is an abstract, simplified view of the world. In a more practical view, an ontology is simply "a published, more or less agreed conceptualisation of an area of content" (De Roure et al. 2005). Ontology provides a commonly agreed set of vocabularies. They can be used to describe things in real world (e.g. resources, objects, concepts, or processing capabilities) in a way that is understandable to machines. Hence, it enables automatic processing, sharing and reuse of machine understandable contents across various applications. In the context of the Semantic Web, ontologies provide a common set of vocabularies for representation of knowledge to support automatic reasoning.

2.2.3.2 Resource Description Framework

Resource Description Framework (RDF) expresses meaning using ontologies (W3C 2004*b*). Each RDF statement is a triple which consists of a subject, a predicate and a object. The predicate describes the relationship between the subject and the object. High level RDF-based ontology languages such as OWL (W3C 2004*c*) are capable of representing inference rules in ontologies to provide further reasoning power.

2.2.3.3 Agent Computing

The third component of the Semantic Web is software agents. Ontologies and RDF help to encode human knowledge in a machine understandable way. Software agents can interpret and act on the encoded knowledge. It is software agents that realise the full power of the Semantic Web.

Although the Semantic Web was envisioned with lots of potential, it has not gained much success at a large scale as expected. This is due to its complex format and requirement for high cost of translation and maintenance from users that makes it difficult to implement the Semantic Web at a large scale (McCool 2005, McCool 2006). However, its introduction has motivated a wide range of applications of ontologies and related technologies in other areas, including Web Services, Grid and P2P computing. In supporting scientific collaborations, the Semantic Web technologies can be used for capturing and sharing scientific knowledge and data. An example usage of the Semantic Web technologies is in CombeChem project (Newman 2006). The Semantic Web can also be used to automate the process of data and service discovery, as in myGrid (myGrid 2006).

2.2.4 Semantic Web Services

Semantic Web Services are Web services marked up with semantics using the Semantic Web technologies (McIlraith et al. 2001). In more detail, a Semantic Web service is associated with a service profile (what the service does), a service model (how the service work) and a service grounding (how to access the service). These descriptions of a Semantic Web service are encoded using Web service ontology (e.g. OWL-S (W3C 2004*a*)) to enable computer agent to discover, execute, compose and interoperate with the Web service automatically (Sollazzo et al. 2002).

2.2.5 Grid Computing

Grids are widely recognised as promising infrastructures for pooling together high-end resources to support distributed collaborations. The term *Grid* in computing is analogous to *grid* in electrical power grids (Foster & Kesselman 1999, Chetty & Buyya 2002). It was initially to address the increasing demand for computing power for computationally sophisticated purposes, motivated by greater sharing of computational results, new problem solving techniques and tools, the increase in demand driven access to computational power and utilization of idle capacity (Foster & Kesselman 1999, Chetty & Buyya 2002). For this reason, grids were initially computation oriented and defined as "hardware and software infrastructures that provides dependable consistent, pervasive and inexpensive access to high-end computational capabilities" (Foster & Kesselman 1999). The computational capabilities referred in this definition include CPU cycles, memory, storages and data. The grids later defined via problems they are addressing, so called the grid problem: "flexible, secure, coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organisations" and "the sharing is not primarily file exchange but rather direct access to computers, software, data and other resources, as is required by a range of collaborative problem-solving and resource brokering strategies merging in industry, science, and engineering" (Foster et al. 2001). In order to avoid misconception that any networked system, such as a cluster of computers and a network file system, could also be called a grid, a three point checklist was introduced as criteria to define a grid (Foster 2002):

i. Coordinates resources that are not subject to a centralised control

ii. Uses standard, open, general purpose protocol and interfaces

iii. Delivers nontrivial qualities of service

These three points are reflected in the definition by Buyya: "Grid is a type of parallel and distributed system that enables the sharing, selection, and aggregation of geographically distributed 'autonomous' resources dynamically at runtime depending on their availability, capability, performance, cost, and users' quality-of-service requirements" (Buyya 2002).

As defined, the Grid problem identifies supporting distributed collaboration by enabling the sharing of computing resources as a main requirement that a Grid needs to address. Grid computing is able to provide consistent, pervasive, dependable, transparent access to high-end computing resources in a seamless, integrated computational and collaborative environment (Baker et al. 2002). It makes "possible for scientific collaborations to share resources on an unprecedented scale, and for geographically distributed groups to work together in ways that were previously impossible" (Foster 2002).

In the context of this thesis, Grids are referred to as networked hardware and software infrastructures that provide consistent, pervasive, dependable, transparent access to high-end computing resources in a seamless, integrated computational and collaborative environment. High end computing resources provided by Grids can be CPU cycles, memory, storage and huge volume datasets.

Grid computing has evolved through three generations, as classified by De Roure, Baker, Jennings & Shadbolt (2003):

- The first generation involved primarily solutions for sharing high performance computing resources in distributed environment. A typical project associated with this first generation technology is I-WAY (Foster et al. 1997).

- The second generation of Grid technologies introduced middleware to address issues of scalability, heterogeneity and adaptability in distributed environments with the focus on large scale computational power and huge volumes of data. There were a number of Grid projects in this second generation, ranging from core Grid technology projects (e.g. Globus version 2 (Foster & Kesselman 1997)) to Grid resource brokers and schedulers (e.g. CONDOR (CONDOR 2006), Nimrod/G (Buyya et al. 2000)), Grid portals and integrated Grid applications (e.g. DataGrid (The DataGrid Project 2006)).

- The third generation of Grid systems is still under development. It addresses the requirements for distributed collaboration in virtual environments. This generation adopts service oriented approach and stresses on the importance of automation enabled by agent computing and knowledge technology. The Open Grid Services Architecture (OGSA) (Foster et al. 2002) implemented in the Globus Toolkit version 3 and currently version 4 (The Globus Alliance 2006) and the Semantic Grid (De Roure, Jennings & Shadbolt 2003) are typical representations of the third generation Grid technologies.

2.2.5.1 Open Grid Service Architecture

Open Grid Service Architecture (OGSA) is becoming a standard for building Grid infrastructures and applications. It adopts a service oriented architecture and Web service standards to enable flexible and interoperable integration of distributed applications in heterogeneous Grid environments. In service orientation view, virtualised resources are represented as services and are peers to other services in the architecture. OGSA specification version 1.0 identified a standard and relatively invariant set of capabilities that need to be addressed in order to meet requirements for Grid applications (Foster et al. 2005):

- *Execution Management Services*: address problems with executing a unit of work, including their placement, provisioning and lifetime management.
- *Data Services*: are used to move data, manage replicated copies, run queries, update and transform data to new format.
- *Resource Management Services*: deal with the management of resources themselves (e.g. rebooting a host), the resources on Grid (e.g. resource reservation and monitoring) and OGSA infrastructure.
- *Security Services*: facilitate the enforcement of security related policy within Grid environments.
- *Self-Management Services*: help reduce the cost and complexity of owning and operating IT infrastructure.
- *Information Services*: access and manipulate information about applications, resources and services in Grid environments.

Figure 2.3 shows how OGSA capabilities (in forms of services) are positioned in three-tier view of Grid infrastructures. The figure is based on the Grid infrastructures described by Foster et al. (2005). The standard capabilities of OGSA are fitted in middle tier of the Grid infrastructures. They operate on base resources and provide services to user applications.

2.2.5.2 Web Services Resource Framework

When SOA and Web Services were adopted in the Grid architecture, Grid developers found the need for transient and stateful Web Services to satisfy requirements from Grid environments. As a result, Grid Services were introduced. A Grid service was defined as a “Web service that provides a set of well-defined interfaces and that follows specific conventions” (Foster et al. 2002). The interfaces address the discovery, dynamic creation, lifetime management, notification and manageability of Grid services. The conventions address the naming and upgradeability. The interfaces and conventions are specified in Open Grid Services Infrastructure (OGSI) Version 1.0 (Tuecke et al. 2003).

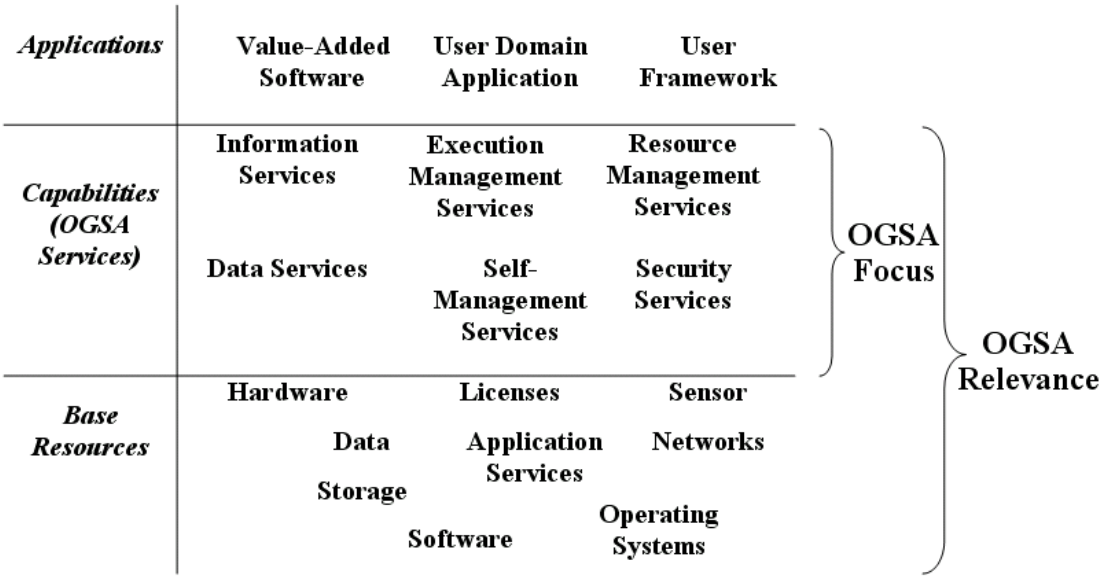


Figure 2.3: Conceptual service oriented view of Grid infrastructures
Based on Foster et al. (2005)

However, the arguments from Web services community are that Web services have no state and that interactions with Web services are stateless (Vogels 2003, Foster et al. 2004). The states are of resources that Web services act upon. There were also critiques about OGSI (Czajkowski, Ferguson, Foster, Frey, Graham, Maguire, Snelling & Tuecke 2004):

- too much detail in one specification
- does not work well with existing XML and Web services tools
- too object oriented

For this reason, Web Services Resource Framework (WSRF) was introduced as a reconciliation (Czajkowski, Ferguson, Foster, Frey, Graham, Sedukhin, Snelling, Tuecke & Vambenepe 2004). WSRF separate Web Services and resources. Web services in WSRF are stateless. The resources associated with Web services are transient and stateful. WSRF is being accepted as a new standard for services in Grid environments. WSRF is being implemented in a number of toolkits, such as Globus Toolkit version 4.0 (The Globus Alliance 2006) or WSRF.Net (Wasson, Glenn 2006).

2.2.5.3 The Semantic Grid

The Semantic Grid is an application of the Semantic Web into Grid computing. The relationship of the Semantic Grid and the Grid connotes a similar relationship that exists between the Semantic Web and the Web (De Roure, Jennings & Shadbolt 2003). "The Semantic Grid vision is to achieve a high degree of easy-to-use and seamless automation to facilitate flexible collaborations and computations on a global scale, by means of machine-processable knowledge both on and in the Grid" (De Roure et al. 2005). Five key enabling technologies that have been identified for the Semantic Grid are Web services, software agent, Semantic Web services, metadata, and ontologies and reasoning. These five key technologies collectively address various requirements for the Semantic Grid.

2.2.6 Portals

A portal is "network service that brings together content from diverse distributed resources using technologies such as cross searching, harvesting, and alerting, and collate this into an amalgamated form for presentation to the user" (Awre 2003). In line with this definition, a Web portal is a portal implemented as an Web application. This is the most common form of portals. In a service oriented view, a Web portal is "a Web-based application that acts as an gateway between users and a range of different high-level services' (Chohan et al. 2005).

From a user point of view, "a portal is a, possibly personalised, common point of access where searching can be carried out across one or more than one resource and the amalgamated results are viewed" (Allan et al. 2004).

Another concept associated with portal is portlet. A portlet is a window which contains some content on a portal (Allan et al. 2004).

2.2.6.1 Portal Architecture

Portals are built on Web architecture and technologies. At the very top level, a portal architecture logically consists of three layers as depicted in Figure 2.4:

- *Clients*: Where users interact with the system. Portal clients are commonly Web browsers. There can applications acting as clients of a portal.

- *Portal*: presents aggregated contents to clients. A portal server may also provide other services to users such as managing user profiles, sessions and states, workflow orchestration, authentication.

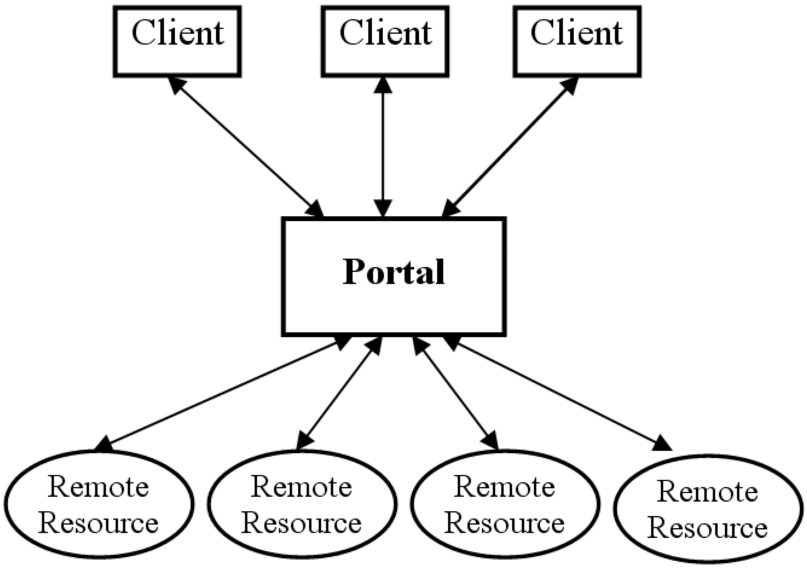


Figure 2.4: Top level view of portal architecture

- *Remote resources*: are contents that the portal presents to its users. The remote resources could be in various forms such as Web contents, files, databases or Web services.

In physical implementation, the portal layer might consist of many Web servers to address the scalability, security as well as the management of different functionalities.

SOA and Web services are being adopted to develop portal applications (Allan et al. 2005). They provide a flexible and interoperable way for integration of distributed contents into portals. Two emerging standards help to make such an integration easier:

- *Java portlet interface JSR-168*: To enable interoperability between portlets and portals. JSR-168 defines a set of APIs for addressing the areas of aggregation, personalisation, presentation and security (Java Community Process 2006).

- *Web Services For Remote Portlets (WSRP)*: defines a set of interfaces and related semantics which standardise interactions with remote portlets. This allows portals to use contents from other portals via their portlet containers without having to write unique code for interacting with each content component (Thompson 2006).

2.2.6.2 Portal Applications

Web portals can be used for a number of different applications. They can be used for e-Commerce applications, such as Amazon[1], or eBay[2]. Portals can also be used to provide information resources, such as the British Academy Portal[3]. In supporting distributed scientific collaborations, the following applications of portals are most important: Web-based collaborative portal and Grid application portals.

2.2.6.3 Grid Application Portals

Grid application portals provide access to services and other type of resources in Grid environments to end users. The common Grid services accessible through Grid application portals are authentication, job management and Grid information services. Examples of Grid application portals include generic HPCPortal projects (Allan 2006), the Open Grid Computing Environments (OGCE) Portal software (OGCE 2006) and NGS Portal for community users to access to National Grid Service in the UK (NGS 2006).

2.2.6.4 Web-based Collaborative Portals

A Web-based collaborative portal is a kind of Web-based collaborative environment, which is an integrated Web-based application that provides facilities for distributed users of a community to perform various collaboration activities. British Atmospheric Data Centre (BADC 2006), BioCoRE for the biologists (BioCoRE 2006, Bhandarkar et al. 1999) and Collaboratory Multi-Scale Chemical Science (CMCS) portal (CMCS 2005, Myers & et al. 2004) are examples of collaborative portals.

Facilities provided by a Web-based collaborative portal commonly include:

- Administration tools: user authentication, security, team management, resource management
- Co-operation tools: team working space
- Coordination tools: group calendars, group information boards
- Resource sharing: shared space for documents and data
- Awareness: through search facilities for identifying relevant resources and well as expertise within the supported community

[1] http://www.amazon.com

[2] http://www.ebay.co.uk

[3] http://www.britac.ac.uk/portal/

- Tools for personalisation
- Communication tools: community information boards, discussion forums, Web chat, video-audio conferences

The advantage of Web-based collaborative portals is that a user can perform collaborative work anywhere with a simple Web browser and internet connection. The portal approach also helps to enrich the resources for collaborative activities by integrating different remote resources such as visualisation tools into the environment. Functionalities of collaborative portals and Grid application portals are sometimes integrated in single portal applications to enhance their capabilities, such as HPCPortal (Allan 2006) or BioCoRE (BioCoRE 2006).

However, as a Web-based collaborative environment is centrally administrated, there were worries about privacy of shared documents stored on the server (Lau et al. 1999). Furthermore, a Web-based collaborative environment is also susceptible to a single point of failure (the central Web server) and scalability if the processing is done centrally (Liu & Gorton 2004).

2.2.7 Peer-to-Peer Computing

P2P is popularised by many desktop file-sharing applications such as Napster (Shirky 2001) and currently Kazaa (Kazaa 2006) or eMule (eMule 2006). P2P file sharing applications have been blamed for supporting violation of copyright laws by movie industry. Indeed, it the human beings that violate the laws, not the technology itself. P2P also has many other potential apart from desktop file sharing. For instance, it has been used for Internet phone system (skype 2006), for distributing services to a community (GSC-Chinook 2006) and for collaborative teamwork (Groove Networks 2006).

In essence, P2P is “a network-based computing model for applications where computers share resources via direct exchanges between the participating computers” (Barkai 2001).

2.2.7.1 Properties of Peer-to-Peer

The definition stresses two fundamental properties of P2P computing: the direct communication and the sharing resources between peer users. These two fundamental properties allow users in P2P environment to communicate directly with each other to dynamically and autonomously establish their own communities without being regulated by any third party administration.

The ability to provide direct communication also allows the users to share resources in a timely manner, especially with the current advance of network bandwidth and personal computer processing power. As resources are shared directly from their computers, users still maintain the sense of ownership on the shared properties and have the right to revoke any resource from sharing anytime.

P2P is a decentralised network-computing model, where computation takes place at the edges. Hence, it is more scalable when the number of users increases. The bottleneck problem, commonly associated with centralised approaches, can also be avoided. Furthermore, P2P applications often provide means for real-time communications, which are highly suitable for direct collaborations amongst scientists. Therefore, not only computing resources but also scientific knowledge could be exchanged more spontaneously.

2.2.7.2 Peer-to-Peer Application Architectures

P2P applications are commonly implemented in three models:

- *Centrally mediated:* in this model, a central server holds a directory of online peers. When requested, the server will initiate the connection between peers. The actual connection is between the peers themselves. This model was implemented in the early version of Napster. MSN Messenger and Yahoo Messenger might also be classified to this category. They are indeed implemented as client-server applications. However, from a user's point of view, the interactions amongst the users are in a P2P manner.

- *Hybrid P2P:* there are a mix of normal peer and super peers, which have higher computing power and connectivity. The network of super peers forms the backbone of the P2P network to maintain connectivity and facilitate resource discovery. Normal peers connect to the network through super peers. Systems implemented in this model include JXTA (CollabNet 2006) and Kazaa (Kazaa 2006).

- *Pure P2P:* in this model, every peer has an equal role. Gnutella (Gnutella 2001, Kan 2001) is an example.

Applications that can support user P2P interactions can be built on system architectures other than P2P. Examples that have been mentioned earlier are MSN and Yahoo Messenger, which are built on client server architecture. Another example application is AccessGrid (Uram 2006), where connected users to a "Virtual Venue" can perform direct (P2P) communication with each other. However, these applications do not have the values that can be provided by a P2P system architecture. For example, in the case of MSN or Yahoo Messenger, if the central server is down, the client applications will not be able to work. In AccessGrid, the “Venue Clients” are totally dependent on the “Virtual Venue”. These problems do not exist in applications built on P2P system architecture, where there is no single point of failure.

2.2.7.3 Applications of Peer-to-Peer Computing

P2P computing model provides lots of potential for building collaborative environments to support scientific research communities, particularly in supporting direct collaborations amongst participants. Capabilities that P2P computing can provide include:

- *File sharing:* for sharing small scale experimental data, working documents. Examples are Napster, Kazaa and eMule.
- *Direct communication:* chat (voice and video), instant messaging. Skype is an example of this kind (skype 2006).
- *Information dissemination:* for disseminating information and resources to members of a community. This is an inverse direction of resource discovery.
- *Sharing computational services:* computational capability, such as ability to run a simulation, can also be shared to other members of a community if Web services are used. Examples of this kind of applications are Triana (Triana 2003) and SETI@home (SETI@HOME 2006).

2.2.7.4 Issues about Peer-to-Peer

There are also issues about P2P computing. In a pure P2P network, where there is no centralised server, connectivity is one of the issue. Every time a peer gets on to the network, it connects to a totally different topology. The peer may not be accessible to another peer even if they are both online at the same time on the same network (Fox & Walker 2003).

Another issue is about scalability of the network. Resource publication and discovery are always important in distributed environment. How to efficiently route a query message in a large distributed P2P network is challenging. Broadcasting method (e.g. Gnutella (Gnutella 2001, Kan 2001)) is straight forward and popular but not efficient. The whole network will soon be flooded with queries if every peer keeps posting. Indexing techniques (e.g. CAN (Ratnasamy et al. 2001), Chord (Stoica et al. 2001) and Pastry

(Rowstron & Drusche 2001)) using distributed hash table have been introduced to address this issue, but this approach requires exact matching of indexed terms. It is not suitable for rich queries.

The last but not least important issue is security. As in P2P, resources on each personal computer are exposed for access to all peers in the network, there is a potential risk to peer computers.

2.2.8 Groupware

Groupware is “software that supports and augments group work”. It is “explicitly designed to assist groups of people working together” (Greenberg 1991). Common examples of groupware are online communication tools such as email, discussion forums, video conference systems and instant messengers.

In distributed communities, these communication tools help to bridge the gap amongst geographically distributed participants. They make the communication amongst people separated by space and time difference become more like face-to-face communication. Particularly, email and instant messengers with their advanced features can help to maintain personal relationships amongst research before and after collaborations by bridging the physical gap. This is a condition for initiating informal communications, which play a very important role in scientific collaborations.

Online communication tools under review are classified into two types: asynchronous communication and synchronous communication.

2.2.8.1 Asynchronous Communication Tools

Asynchronous communication refers to the type of communication that does not require participants to be available to communicate at the same time. Typical asynchronous communication tools are email, Web-based discussion forums.

Email. The first email was used in early 1960s for users of a time-sharing mainframe computer to communicate (Crocker 2006). Although, it far predates the Internet, the modern email systems are running on the Internet environment. Ability to provide asynchronous communication and to carry attachments of any content, together with popular use of the Internet, have made email become a dominant communication tool for Internet users.

Web-based discussion forums. Also known as bulletin boards, Internet forums, message boards, discussion boards and discussion forums. Discussion messages are stored on Web servers. Discussion messages can also be set to be accessible privately to a group of participants to everyone in the public. Many implementation of Web-based discussion forums today provide more advanced features such as list of other logged in users, emotional symbols, avatars, ability sending private messages to a particular user to his/her email.

Advantages of asynchronous communication are in its flexibility. Using asynchronous communication tools, communicators are free from time constraint. They have time for reflection and opportunity to research back to assertions through stored messages (Anderson & Kanuka 1997, Andriessen 2003).

However, it is difficult for participants to socialise when using asynchronous communication tools. Information exchanged through this mode of communication is not as rich as in face-to-face contacts. This limits the users in their ability to communicate (Anderson & Kanuka 1997).

2.2.8.2 Synchronous Communication Tools

Synchronous communication requires the presence of participants at the same time. Messages exchanged through this type of communication are instantly in real-time. Small delays might occur due to network traffic processing. There have been many researches on the ability of synchronous communication tools in distributed community to substitute the lack of face-to-face contacts. Asynchronous communication tools commonly under consideration are based video and audio technology and computer mediated online chat (using Web chat rooms or instant messengers).

Video and audio tools. Video and audio communication tools have been used in distributed communities for a number of purposes: videophones, video-conferences and media spaces.

- *Videophones:* for one-to-one communication. The first commercial videophone system was AT&T's Picturephone. However, this was also a costly failure (Andriessen 2003).
- *Video-conferences:* for meetings groups. AccessGrid is an example of this kind (Uram 2006).

- *Media spaces:* for extending the boundary of a physical office space using video and audio connections. The goals of media spaces are to provide members of a distributed community a sense of physical proximity and the awareness of other members' presence so that social encounters can happen. As a result, informal communication, which plays a very important role in supporting collaborations (Hagstrom 1965, Edge 1979, Kraut et al. 1986, Kraut, Egido & Galegher 1990, Kraut, Fish, Root & Chalfonte 1990, Isaacs et al. 1997), can be initiated.

  There have been a number of different implementations of media spaces: open video and audio connection amongst distributed locations (Fish et al. 1990, Bly et al. 1993), periodically glancing at other desks (Tang et al. 1994, Fish et al. 1992) or providing an overview of people currently in office by an matrix of images updated regularly (Lee et al. 1997, Dourish & Bly 1992).

Researches have shown that video and audio systems have many advantages in supporting distributed communities:

- Providing social awareness (Lee et al. 1997, Bly et al. 1993, Dourish & Bly 1992)
- Increasing social encounters and relationships (Bly et al. 1993, Fish et al. 1992)
- spontaneous communications (Lee et al. 1997)
- improving mutual understanding by forecasting responses and using non-verbal communication through video channel (Isaacs & Tang 1993)

However, there are also a lot of factors that made early video systems fail to achieve their objectives:

- Social embarrassment and camera shyness (Egido 1988, Lee et al. 1997, Obata & Sasaki 1998)
- Lack of audience cues and spatial orientation (Lee et al. 1997, Andriessen 2003, Wainfan & Davis 2004, Bly et al. 1993)
- Cost of hardware and poor quality of video and audio (Whittaker 1995, Kouzes et al. 1996)
- Privacy (Fish et al. 1992)

Online chat. Online chat exists in two forms: through chat rooms on the Web using Web browsers or through instant messenger clients installed on personal desktops such as MSN Messenger or Yahoo Messenger. During a chat session, participants involved exchange text messages to communication. Advanced chat systems such as Yahoo Messenger and MSN Messenger integrates more features into the client such as friend list, voice and video chat, conferences, receiving off-line messages, gaming and radio, etc.

Studies have shown that online chat, particularly instant messengers, could help group members have the awareness of others' presence in distributed environment. Instant messengers are also very useful for casual and friendly communication, for posing short questions and getting quick responses. They are flexible and easy to use. (Quan-Haase et al. 2005, Muller et al. 2003, Herbsleb et al. 2002, Nardi et al. 2000, Isaacs et al. 2002)

2.3 Related Projects for Supporting Distributed Scientific Collaborations

A number of programmes and projects have been set up to develop infrastructures for supporting distributed scientific collaborations. In the UK, many projects of this kind were funded under the e-Science (NeSC 2006, Research Councils UK 2006) and Virtual Research Environment programmes (VRE 2006). The e-Infrastructure, which has just been started in September 2006, is based upon these two programmes for a large scale infrastructure for supporting distributed collaborations (Farnhill, James 2006). In the US, the national collaboratories and Cyberinfrastructure programmes have also undergone for this same purpose (Atkins et al. 2003). European Commission has also funded a number of collaboration projects such as the Enabling Grids for e-Science project (EGEE 2006). Related projects from these programmes are discussed in the following sections. The selected projects are relevant to this work in terms of architectural issues as well as the similarity of problem domains.

2.3.1 UK e-Science Projects

The UK e-Science programme started in 2001. As defined by John Taylor, "e-Science is about global collaboration in key areas of science, and the next generation of infrastructure that will enable it" (NeSC 2006). The focus of e-Science programme is to develop infrastructures to enable distributed collaborations in large-scale science. The Grid is commonly referred as the infrastructure to enable such kind of collaborations.

There have been a large number of projects in e-Science programme. They cover many areas of science, such as particle physics, astronomy, biology, biomedical, chemistry and environmental science. Outcomes of these projects include Grid middleware, services, and tools to support distributed scientific collaborations within their problem domain. The following are most relevant projects from the e-Science programmes.

2.3.1.1 CombeChem

CombeChem was an e-Science pilot project, concentrating on Grid enabled combinatorial chemistry (Frey et al. 2003, Newman 2006). This project was based on pervasive computing and knowledge technologies together with Web services to facilitate the concept of "publication at source" (Frey et al. 2002). This was demonstrated in Smart Tea, a CombeChem subproject (SmartTea 2006). In Smart Tea, pervasive computing devices were used to capture laboratory data at sources. The captured data was then annotated using ontologies. The annotated data was published to other relevant parties using Grid-based services.

The kind of scientific collaborations addressed by the CombeChem includes the sharing of experimental data and scientific knowledge. The sharing in CombeChem is enabled by pervasive computing devices and the underlying Semantic Grid infrastructure.

2.3.1.2 myGrid

myGrid was another e-Science pilot project (myGrid 2006, Goble et al. 2003). The focus of myGrid was on Web services for experiments in biology and workflow orchestration - assembling, adapting and composing - on the services. The project has developed a loosely-couple suite of middleware components to support data intensive in silico experiments, which consisted of a workbench for assembling and running workflows. In myGrid, knowledge technology was used for annotating and discovery of services.

In terms of supporting scientific collaborations, myGrid focused on the sharing of computational resources and scientific knowledge for experiments in forms of Web services. The collaborations between scientists are done implicitly via the services and workflows of services.

2.3.1.3 NERC DataGrid

The NERC DataGrid was jointly funded by UK e-Science programme and Natural Environment Research Council (NERC). It aimed to facilitate the discovery, delivery and use of data in environmental science held in a loosely coupled federation of distributed locations (BADC 2005, Woolf & et al. 2004, Lawrence & et al. 2004). Metadata was used in NERC DataGrid to support the discovery and delivery of environmental data.

The NERC DataGrid project is related to this work in terms of the problem domain - the atmospheric chemistry. However, the NERC DataGrid is focused primarily the sharing of environmental data using a grid infrastructure. The interactions with end users are expected to be on a Web-based collaborative portals.

In general, projects in e-Science programme commonly address the sharing of computational resources, storage and large experimental datasets. The Grid computing, knowledge technologies and services oriented architecture are the main focus. Supporting *ad hoc* collaborations between end users in a distributed scientific community, such as communication to share a research idea, sharing a draft report or sharing reading paper is not covered in e-Science projects, though these activities might be important parts of a distributed collaboration process.

2.3.2 The Virtual Research Environments Programme

The Virtual Research Environments (VREs) programme began in 2004 (VRE 2006). One of its goals is to develop and deploy VREs, which are environments that help researchers manage the complexity of their research by providing an infrastructure specifically designed to support research activities carried out within research teams, on both small and large scales (Allan et al. 2005). VREs are portal based, built on top of Grid infrastructures. Existing portal frameworks such as Sakai (Sakai 2006) and OGCE (OGCE 2006) are being adopted.

Supporting collaborations amongst members of distributed communities is a primary goal of the VRE. Web-based collaborative portals have been chosen as the approach. Using the collaborative portals, end users can perform a range of collaborative work including *ad hoc* collaborations, such as sharing knowledge and data, co-editing research papers and co-monitoring experiments. The end users can also get access to computational resources and data from the Grids via the portal. However, the Web-based collaborative portal approach has some limitations in supporting end users' collaborative activities as well as the sharing of knowledge across community boundaries. More detailed discussion of the issues is provided in the next section (Section 3).

2.3.2.1 GridPP and Enabling Grids for e-Science

GridPP is a collaboration of 19 UK universities, CCLRC and CERN (GridPP 2006). It is building a Grid infrastructure for supporting collaborations amongst particle physicists in the UK and CERN. GridPP is connected to the Worldwide LHC Computing Grid (LCG 2006) and also part of the European Enabling Grids for e-Science (EGEE) network (EGEE 2006).

The EGEE project involves researchers from over 27 countries. It aims to deliver a production grid infrastructure for researchers in many areas of science, including earth science, high energy physics, bioinformatics and astronomy (EGEE 2006).

GridPP and EGEE projects are relevant to this research in the size and scale of communities that are trying to support. The Grid infrastructure is one supporting component. At the current stage, it is not clear that whether or not these two projects are going to use another component, in additional to the Grid infrastructure, for supporting *ad hoc* collaborations.

2.3.3 Collaboratory for Multi-Scale Chemical Science

Collaboratory for Multi-Scale Chemical Science (CMCS) project was funded by US Department of Energy National Collaboratories programme. Other projects under this programme included the Particle Physics Data Grid Collaborative Pilot (PPDG 2006*b*, PPDG 2006*a*) and the National Fusion Collaboratory (FusionGRID 2004).

CMCS used a combination of Grid infrastructure and a Web-based collaborative portal. The Grid was for management of datasets of chemical science. This management of data was based on the use of metadata. The collaborative portal was an environment for end users collaborations (CMCS 2005, Myers & et al. 2004, Myers et al. 2005).

One of the community that the CMCS is supporting as its pilot is the combustion research community, which is related to the reaction kinetic community - the community used for the case study of this research. The CMCS portal approach is similar to the approach of the VRE programme discussed earlier. Hence, the CMCS shares common issues in terms of supporting *ad hoc* collaborations as the VREs.

2.3.4 Triana

Triana is a service oriented problem solving environment (Triana 2003, Taylor et al. 2003). Triana focuses primarily on integrating distributed compute services and composing the services into workflows. The services in Triana can be from P2P environments, Grids or from any providers. The kind of scientific collaborations supported by Triana is the sharing of computational services for scientific computations, such as visualization or simulations. *Ad hoc* collaboration activities are not covered by Triana.

2.3.5 The Process Informatics Model

The Process Informatics Model (PrIMe) was set up by an initiative consisting of combustion researchers from the US and Europe (PrIMe 2006). The aim is to develop a process model to promote collaborations amongst combustion researchers. At the current stage, the PrIMe is adopting a data centre approach for combustion data. The case study in Section 4 has more discussion about PrIMe and the combustion research community as a related community to the reaction kinetics research community.

2.4 Summary

This chapter has discussed the key characteristics of modern scientific collaborations. These characteristics are the basis for the requirements for supporting collaboration technologies. A number of technologies for supporting distributed scientific collaborations have also been reviewed.

Generally, collaboration technologies reviewed fall into three categories:

- The first category consists of technologies for providing and delivering computational and data resources. This category includes Grid computing, SOA and Web services. Grid computing plays a dominant role as its capability to enable the sharing of large-scale computation and data resources. SOA and Web services have been adopted into Grid computing environments to enable flexible and interoperable integration of distributed applications. Using Web services makes the delivery of distributed resources easier and more flexible

- The second category consists of technologies that directly support collaborations amongst end users. Typical of this category are Web-based collaborative portals and basic communication tools of groupware. Basic communication tools have an important role in supporting informal communication, and hence, in initiating distributed collaborations, especially the arising role of instant messengers. Web-based portals are commonly used to provide users access to Grid computing resources and well as environments for collaborations amongst end users. P2P computing can potentially be used to support collaborations amongst end users, benefiting from its decentralised architecture, direct connections with other peers and the ability to utilise unused CPU cycles at the edge of the network. However, currently, P2P is most popular with commercial file sharing applications.

- The third category is composed of technologies for management of knowledge. The key technologies in this category are ontologies, Semantic Web technologies and the Semantic Grid. These knowledge technologies can be used in Grid environments for annotation of Grid resources, services and data to support automatic resource discovery. They can also be used in collaborative environments for personalisation of end users' workspace.

A number of relevant research projects have also been discussed. Each of these related projects used/uses a number of collaboration technologies reviewed above for supporting various aspects of scientific collaborations. Commonly, Grid computing infrastructures are used for sharing computational intensive resources and scientific datasets. Web-based collaborative portals are used as environments for end-users to perform day-to-day *ad hoc* collaboration activities. Although P2P computing has the potential for supporting collaborations amongst end users, its popular applications are still limited in file sharing.

Chapter 3

The Collaborative e-Science Architecture - CeSA

The review on a number of related projects on scientific collaborations has shown that Web-based collaborative portals are commonly used on top of Grid infrastructures as environments for *ad hoc* collaboration activities amongst end users. However, Web-based collaborative portals are based on a centralised model, while the scientific communities are heterogeneous and decentralised. The use of a centralised model for distributed communities could be inflexible (Tian et al. 2003) and bottlenecks might occur (Liu & Gorton 2004).

This chapter firstly explains in detail why the Web-based collaborative portal approach might not be efficient enough to support collaborations in distributed scientific communities. It then discusses the potential of using the decentralised model of P2P computing for supporting *ad hoc* collaborations in distributed communities to overcome the limitations of the Web-based collaborative portals. A novel Collaborative e-Science Architecture (CeSA), which is a combination of a P2P collaborative environment on Grid computing infrastructures, is proposed. The aim is to bring together back-end resources from Grid environment into a P2P collaborative environment to leverage the advantages of both Grid and P2P technologies. The architecture focuses on the support for general *ad hoc* collaboration activities amongst scientists as well as for the sharing of computational capability (e.g. for simulations and analyses) required by the scientific communities. The integration between Grid infrastructures and the P2P collaborative environment of the CeSA is based on a service oriented architecture.

3.1 Limitations of Web-based Collaborative Portals

As Web-based collaborative portals adopt the client-server architecture, community members of a collaborative portal are all connected to a centralised server. All communications amongst members, and hence the collaborations, have to be done via the central server. Furthermore, communities operating on Web-based collaborative portals are usually subjected to centralised administration and authentication. This makes it difficult (or sometimes impossible) for distributed members of a community to perform collaborations with members of other communities using different collaborative portals.

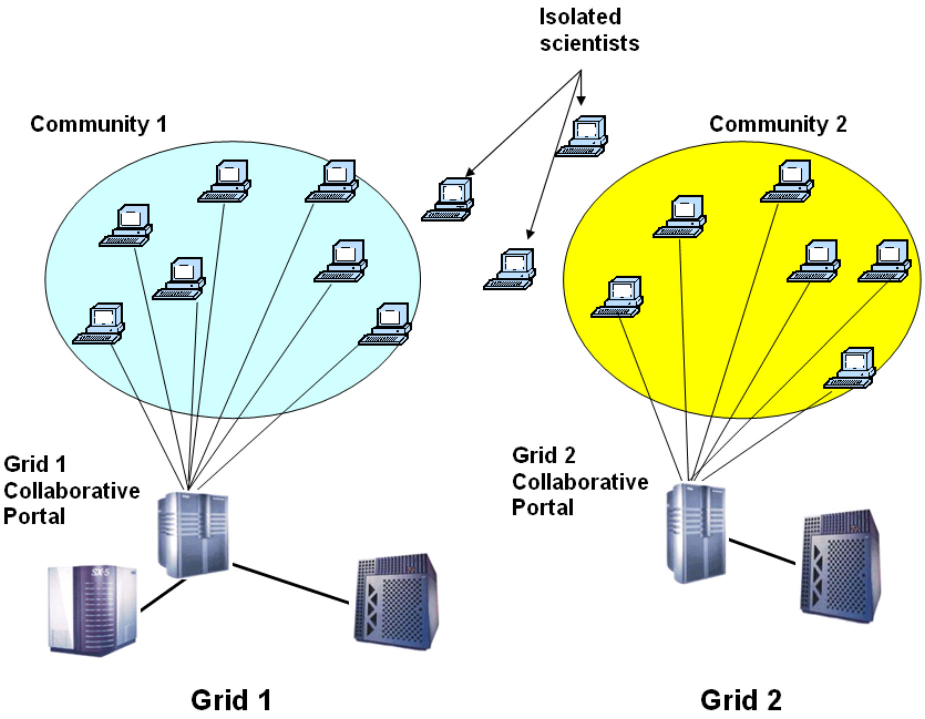


Figure 3.1: An illustration of using Web-based collaborative portals

Figure 3.1 illustrates the use of Web-based collaborative portals for two scientific communities, named as Community 1 and Community 2. Each community operates on a different collaborative portal. The two portals are parts of two disconnected grids, named

Grid 1 and Grid 2 respectively. There are also three other scientists who do not belong to any of the two communities.

As can be seen in the figure, interactions (e.g. sharing a file) amongst members of a community have to be done via the portal, on which the community operates. A member from one scientific community cannot collaborate with any member from the other community. Scientists who are not members of any community are isolated.

In addition, the centralised management of Web-based collaborative portals also discourages end users to collaborate for the following reasons, especially in terms of resource sharing:

- *Lack of control over shared resources*: As discussed in Section 2.1 of the previous chapter, competition is an characteristic of scientific research communities. New research data might keep a researcher to maintain his/her own competitive advantages. In a Web-based collaborative environment, when making a share, the data owner will need to upload his/her data to a central storage. Doing this, he/she implicitly transfers his/her control of data to a centralised administration. The data on the central storage might be exposed to his/her competitors. For this reason, researchers may not be willing to share.
- *Inconvenience*: In order to make a contribution to a central repository, a contributor needs to be proactive. He/she needs to choose a right datasets, then upload it to the central repository. In case of the data is frequently updated, the contributor may be deterred from making any contribution due to time and bandwidth limits.
- *Do not scale well with growing community*: As collaborations via Web-based collaborative environments are subject to a centralised management, the scalability will be an issue when the size of community is increased.

3.2 Potential of Peer-to-Peer Collaborative Environments

A P2P network is decentralised, made up by direct connections amongst individual peers. In contrast with Web-based communities, communities operating on P2P networks are highly autonomous and not subject to centralised controls. For these characteristics, P2P environments can potentially be used for supporting *ad hoc* collaboration activities with distributed scientific communities.

Figure 3.2 shows how a P2P environment can be used to support collaborations within the two communities, as shown in Figure 3.1. In a P2P environment, the boundaries of the two communities no longer exist. Members of different communities can freely interact without the need for a third party server, such as a portal. Therefore, collaborations can be carried out seamlessly across the two communities. Connections to the grids are only necessary when Grid resources are needed. Isolated scientists can now easily interact with other scientists.

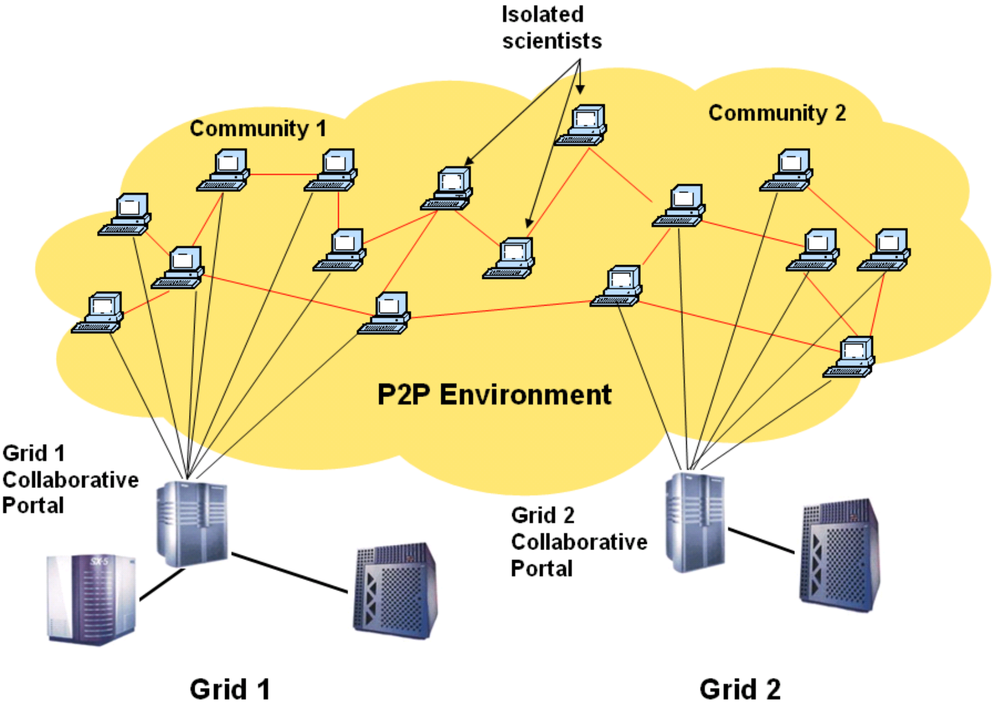


Figure 3.2: A P2P environment for end users' collaborations

The decentralised model and direct communication characteristics of a P2P environment can potentially bring in the following advantages for supporting distributed scientific collaborations. These advantages have been raised in a number of previous studies (Minar & Hedlund 2001, O'Reilly 2001, Shirky 2001, Parameswaran et al. 2001):

- *Information sharing is made easier*: In a P2P environment, information sharing amongst users is made easier. The sharing is not limited to files and chat. Peer users can also easily share ideas and viewpoint (Minar & Hedlund 2001, O'Reilly 2001, Parameswaran et al. 2001).

- *Utilisation of individuals' data*: One distinctive feature of P2P is that resources are available at edges of the network (Shirky 2001). An implication is that personal resources on users' desktops could be made accessible to other users. This allows other users to be aware of available resources in the community. It will lead to higher utilisation of desktop's resources (O'Reilly 2001).

- *Formation of ad hoc communities*: Communities based on a P2P network are highly autonomous and have no actual physical boundaries. A user can connect to any other users on the same network. This makes the formation of *ad hoc* communities, which is usually a barrier in centralised systems (O'Reilly 2001). This creates opportunities for a wider user community to work collaboratively together.

- *Fault tolerance*: The decentralised architecture of P2P is seen as the solution to bottleneck problem (O'Reilly 2001). No single point of failure exists in a P2P network (Parameswaran et al. 2001). This allows users to collaborate together across community boundaries. P2P network is also scalable as the number of compute nodes and network bandwidth is increased as the same rate as the population of the network communities.

- *Control over shared resources*: In addition, a user in a P2P can decide which data to share and who to share with. It is up to data owner to decide whether to share the data. As the shared data is located on the user's desktop, the user has absolute control over it.

In essence, in a P2P environment, services (e.g. collaborations, sharing and discovery of resources) can be provided to users in a door-to-door fashion, whereas, in a Web-based environment, all services are mediated by third party operators. The door-to-door approach provides end-users with greater control over their own resources and activities. Also in P2P, users have a high degree of autonomy. They can freely establish their own relationships and form *ad hoc* communities for their different purposes of collaborations. Hence, they can be more encouraged to actively get involved in it. Generally, "The P2P paradigm can be extended and enhanced to foster productivity in the workplace and support community activities" (Parameswaran et al. 2001).

3.3 The Collaborative e-Science Architecture

The goal of the CeSA is to address the current limitations of Web-based collaborative portals in order to efficiently support distributed scientific collaborations. In the CeSA, an integration of a P2P collaborative environment on top of Grid computing infrastructures is proposed to achieve this goal. This integration will provide a scalable collaborative environment, in which scientists can perform various day-to-day scientific collaborations. It will also be able to provide scientists with easy access to back-end computational resources and scientific data from Grid computing environments. The integration is made possible by using a service oriented architecture.

The CeSA should be viewed as a complement to the current Web-based collaborative portals, not a replacement. The use of a P2P collaborative environment will add in another dimension, based on direct communication channels between peers, for distributed scientific collaborations. It will open up the boundary of scientific communities and bring in the advantages of P2P computing model as discussed in Section 3.2.

3.3.1 High Level View of the CeSA

A high level view of the CeSA is shown in Figure 3.3. The CeSA consists of two layers: a P2P collaborative environment on top of a Grid environment. These two environments are loosely connected using services [1].

The Grid environment consists of a number of grids. They form the backbone of the CeSA. Grids are placeholders of computationally intensive resources (e.g. CPU cycles, memory and network bandwidth), storages and scientific data (e.g. data generated by experiments). Large-scale collaborations over these computationally intensive resources and scientific data are made possible by these supporting Grid infrastructures.

A P2P collaborative environment will be built on top of the Grid environment for individual scientists to perform distributed collaboration activities. The kind of collaborations supported by the P2P environment is more lightweight, such as a direct exchange of information about Grid resources, sharing a working dataset or forming an *ad hoc* working group. This P2P collaborative environment is also the place where scientists can get access to Grid back-end resources and scientific datasets.

This combination will be able to support a range of scientific collaboration activities, from sharing of back-end computational resources and large volume datasets to spontaneous *ad hoc* collaborations on lightweight resources.

The Grid and P2P environments are loosely integrated using a service oriented architecture. In this service oriented view, back-end Grid resources are enclosed and exposed as services. The interfaces of services and their associated information are published in the P2P collaborative environment. Scientists in the P2P collaborative environment can perform resource (in form of services) discovery and then gain access to Grid resources

[1] The term *services* used in the CeSA are abstraction of distributed services. They can be implemented as Web services, Grid services or WSRF.

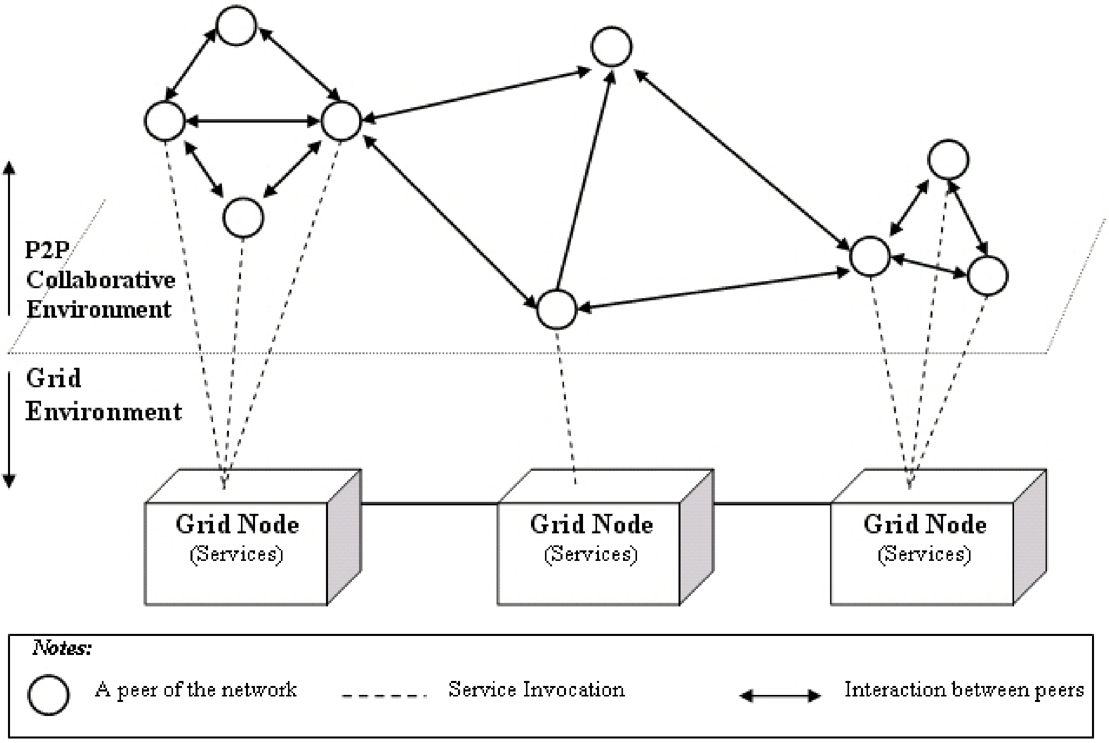


Figure 3.3: High level view of the Collaborative e-Science Architecture

via published service interfaces. This loose-coupling between the two environments will serve the following purposes:

- The P2P collaborative environment will be independent from the Grid to maintain its openness and autonomy to attract a wider range of user communities.
- The collaborative environment will be able to function without the requirement for the existence of any grid.
- To separate the complex and highly secured management within the Grid environment from day-to-day collaborations in the autonomous P2P environment.
- To benefit from Grid computing infrastructures currently available.

3.3.2 Specifications of CeSA Components

3.3.2.1 CeSA Service Oriented Architecture

As discussed in Section 2.2.1 of Chapter 2, a basic services oriented architecture consists of three main roles: service provider, service client and discovery agency. These three basic roles are mapped into the SOA of the CeSA as shown in Figure 3.4.

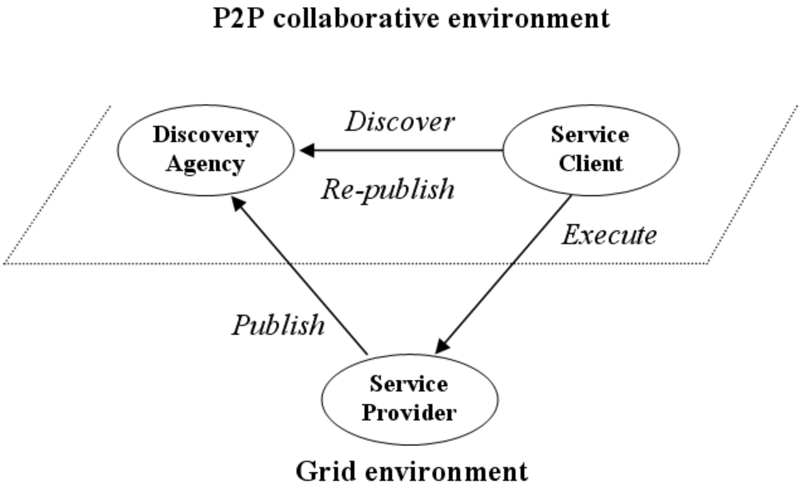


Figure 3.4: Service oriented architecture of the CeSA

Service providers are from Grid environment providing Grid resources to the P2P collaborative environment in form of services. The information about services is published to *discovery agency* in the P2P collaborative environment. *Service clients* which are P2P applications from the collaborative environment can discover information about available services through the *discovery agency*. A *service client* can execute a service by connecting to the provider of the service in the Grid environment. A *service client* may also want to re-publish information about useful services to other service clients in the P2P environment through *discovery agency*. In the CeSA, the discovery agency is an abstract concept. A P2P collaborative environment is decentralised, the actual publication and discovery of services is done collectively by service publication and discovery agents of P2P applications.

As services published in the P2P collaborative environment are directly operated by end users, who may have limited knowledge about information technology, they need to be easily executed, once discovered. Therefore, there is the need for a common standard interface for the services so that a simple service client program embedded in a P2P application can invoke any of the services.

3.3.2.2 Grid Environment

The Grid environment of the CeSA is made of one or many grids. These grids could be data grids and/or computational grids. In the CeSA's service oriented view, a grid provides its resources to the P2P collaborative environment in form of services.

Figure 3.5 illustrates an OGSA based Grid architecture for the CeSA. As shown in the figure, the grid provides its resources to the P2P collaborative environment as Application Services. Application Services are community and application specific services, such as an analysis service for analysing DNA sequence in biology or a chemical reaction simulation service in combustion chemistry. Output from an application service is directly usable by scientists. A common way to build application services in OGSA is composing lower level OGSA services, which directly operate on Grid resources, as shown in Figure 3.5. An application service can be built by wrapping directly a Grid resources, such as a simulation program, into a service or by composing other available application services. The application services from grids need to conform to the common standard service interface specified by the SOA of the CeSA.

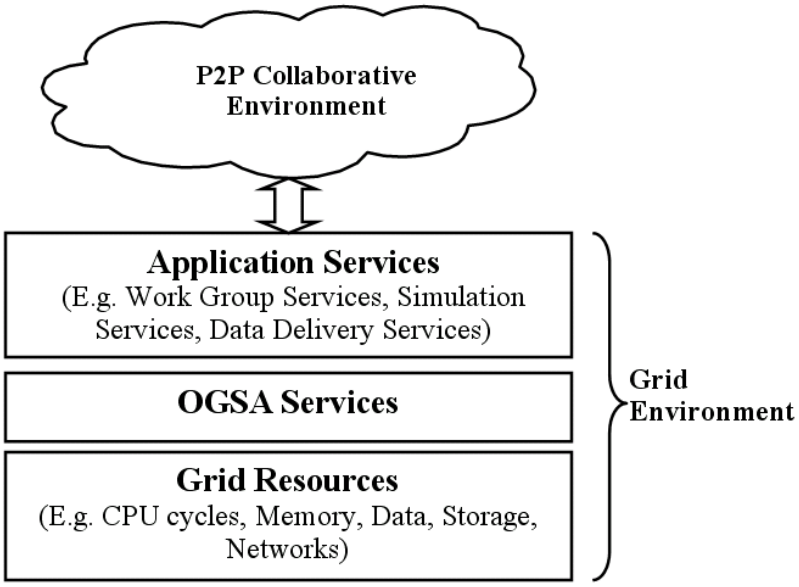

P2P Collaborative Environment
Application Services
(E.g. Work Group Services, Simulation Services, Data Delivery Services)
OGSA Services
Grid Resources
(E.g. CPU cycles, Memory, Data, Storage, Networks)
Grid Environment

Figure 3.5: An OGSA-based Grid architecture for the CeSA

In Figure 3.5, the OGSA is used as an illustration of Grid computing architecture. Indeed, the SOA of the CeSA only requires resources from the Grid environment to be exposed to the P2P collaborative environment as services and these services have to conform to a standard interface, regardless of internal architecture of the grids. Therefore, the P2P collaborative environment of the CeSA can interoperate with different Grid architectures as long as they can provide services and their services conform to the specified standard interface.

3.3.2.3 Peer-to-Peer Collaborative Environment

The P2P collaborative environment consists of a number of P2P applications, which may be run on personal workstations or hand held devices. A P2P application is an interface between a user and the environment. It is a place for users to perform collaboration activities. More specifically, functions of a P2P application include:

- Allowing users to perform basic communication, such as instant messaging, voice and video chat
- Supporting discovery of resources, which includes information about grid resources, expertise and potential collaborators
- Providing facilities for users to form working groups
- Enabling P2P resource sharing within the whole environment or within particular working groups
- Providing access to resources from the Grid environment via application services

These functions are provided by these components of a P2P application: User Interface, Grid Service Client, Service Publication and Discovery Agent, a set of Community Services and Peer Core Component. Figure 3.6 shows relative positions of these components in a P2P application and their interconnections (represented by bi-directional arrows).

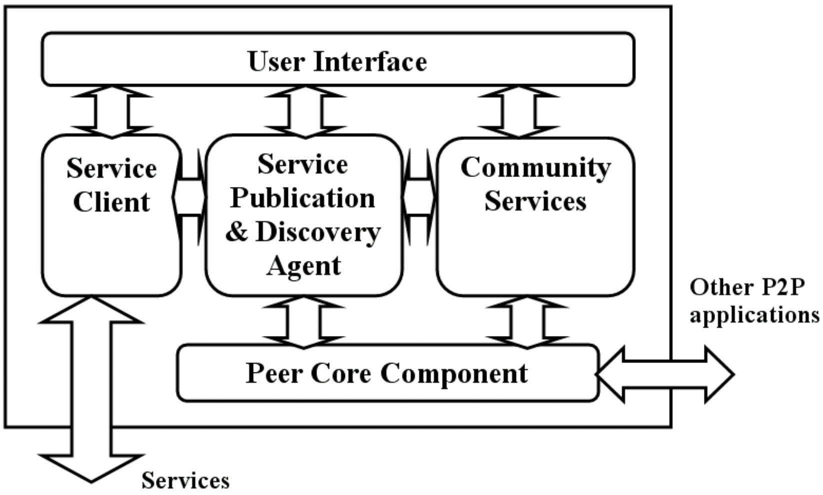


Figure 3.6: Components of a P2P application of the CeSA

User Interface

User Interface is the interaction point between users and the application. Through this interface, a user can manipulate functions provided by other components of the application.

Service Client

Service Client allows a user to browse and to run application services supplied from the Grid environment. As specified by the SOA of the CeSA, the Service Client should be able to execute virtually any service that conforms to the standard service interface.

Service Publication and Discovery Agent

Service Publication and Discovery Agent plays a very important role in the collaborative architecture. It provides two basic functions to the application: publishing information about resources, including information about application services available on the Grid environment, to the P2P environment, and discovering information about resources previously published by other users. In addition to these two basic functions, an agent is also in charge of processing search query sent to it from other peers.

For discovery, the agent formulates a search query and sends the query to service publication and discovery agents of other peers using the P2P communication channel set up by Peer Core. For publication, the agent can get information about the service to be published from the service client or from its search results. The information is then sent to other peers using P2P communication channel. The scope of publication and discovery can be narrowed to a particular community or an interest group using information from the Community Services component.

Community Services

Community Services consists of service components for the day-to-day collaboration within a community. Examples of community services include components for file sharing, community/group formation and instant messaging. Through these components, a user can set up a working group or a community. Then, the user can establish sharing relationships with other users in his working group or community. The Community Services component relies on the Peer Core for communication with other peers.

Peer Core Component

Peer Core Component makes it a peer in a P2P network. It provides mechanism for communication with other peers, peer identification and peer discovery within the network. Service Publication and Discovery and Community Services rely on this core component.

3.4 Summary

This chapter has discussed the limitations of the Web-based collaborative portals in supporting distributed scientific collaboration. Potential of P2P computing that can be exploited to overcome these limitations have also been identified. The CeSA has been introduced as an integration of Grid and P2P computing using a service oriented architecture to address the limitations of the Web-based collaborative portals. However, there are issues related to P2P computing that need to be addressed to turn the potential into reality as well as to have a successful implementation of the CeSA.

From a user point of view, usability and acceptability are important issues. Although P2P file sharing and instant messaging applications are popular, the use of a P2P collaborative environment in scientific domains has not been tested. Therefore, before any actual implementation of the CeSA, it is necessary to study the usability and acceptability of such an novel architecture in potential user communities. The next chapter of the thesis, Chapter 4, will deal with these issues by conducting a case study on the reaction kinetics research community.

From technical point of view, implementation of any P2P application also opens up a number of technical challenges. For example:

- How to enforce security and trust to protect personal computers in a decentralised network.
- How to maintain the connectivity of the P2P network, so that the collaboration can be carried smoothly and reliably.
- How to efficiently locate necessary resources in a decentralised P2P network, where resources are distributed at the edges.

In this thesis an adaptive resource discovery method will be proposed for the P2P collaborative environment in Chapter 5. Other technical challenges will be discussed in future work.

Chapter 4

A Case Study - The Reaction Kinetics Research Community

The focus of this chapter is on a case study, which was conducted for the following purposes:

- To demonstrate the applicability of the CeSA in a scientific research community. The applicability is demonstrated by showing how CeSA can be used to address the requirements for supporting distributed collaborations within the scientific research community selected for the case study.
- To evaluate the usability of the CeSA. The usability is assessed via potential end users' reaction to an implemented instance of the architecture, and their feedback on the comparisons between the functionalities provided by the architecture and their current working practices, particular with the functionalities provided by a Web-based collaborative environment.
- To confirm and collect further requirements which can be used for the next version of the CeSA. An iterative approach is chosen for this part of the research. After each iteration, the collected requirements and feedback are fed into the next iteration to improve the architecture.

The reaction kinetics research community was chosen for this case study to address the above objectives. The community was looking for an information technology infrastructure for their increasing demand for distributed scientific collaborations. The information and data for the case study were collected through collaborations with the reaction kinetics research group at the School of Chemistry, University of Leeds by a series of scheduled meetings with the group's members, discussion emails, observations and reports published by the community.

This chapter is organised into four sections. The first section reports on characteristics of the reaction kinetics research community, from which requirements for collaborations were identified. The second section demonstrates how the CeSA could be used to address the requirements of the reaction kinetics research community. This section also includes a description of an implementation of the CeSA. The third section focuses on a user evaluation of an implementation of the CeSA by end users from the community. The last section reflects on the case study and its implication for future work.

4.1 The Reaction Kinetics Research Community

The community consisted of reaction kinetics researchers all over the world. The scale of distribution could be demonstrated through the PrIMe (Process Informatics Model) consortium (PrIMe 2006). This initiative was set up by combustion experts to coordinate the development of predictive chemical reaction models, a branch of reaction kinetics, from UK, USA, Denmark, Germany and France. Another example was from atmospheric chemistry research community, which was another application area of reaction kinetics. The number of registered users of the British Atmospheric Data Centre (BADC 2006) as on the 24th of April 2006 was 3505, from 54 different countries.[1]

4.1.1 Research in Reaction Kinetics

Research in reaction kinetics and its related application areas, such as in combustion and atmospheric chemistry, was centred on reaction models (Pilling 1997). The basic component of a reaction model was a chemical reaction mechanism, which consisted of a series of steps called elementary reactions in which chemical species were inter-converted. Each elementary reaction was associated with involved species (reactants and products) and a rate coefficient, which determined the rate at which the reaction occurred. The elementary reactions and their associated rate coefficients were investigated in laboratory. It was also feasible to calculate some rate coefficients using quantum theory. The computing

[1] Although the information presented here and in the following subsections collected at an earlier stage of this research, the general picture remained the same as the time of reporting.

resource needed for this approach was substantial. This mechanism could then be used to construct a model that consisted of a set of ordinary differential equations that represent the rates at which the concentration of individual species in the mechanism would change with time.

There had been a wide range of applications of reaction models in combustion, atmospheric chemistry and environmental studies. Application of reaction models in combustion, for example, would involve the interaction between chemistry and fluid dynamics. This would add a further stage of complexity and would require an additional set of scientists with specific expertise. This stage was essential in applications to real systems, such as the design of engines. This stage would be mainly in the domain of engineering. However, continuing collaborations would ideally beneficial, so that feedback and cross-fertilisation of ideas were possible.

4.1.2 The Three Stage Modelling Process

The reaction modelling process, the central activity in reaction kinetics research, consisted of three stages: gathering and evaluating data, generating mechanisms and models, and publishing and archiving models, as summarised in Figure 4.1.

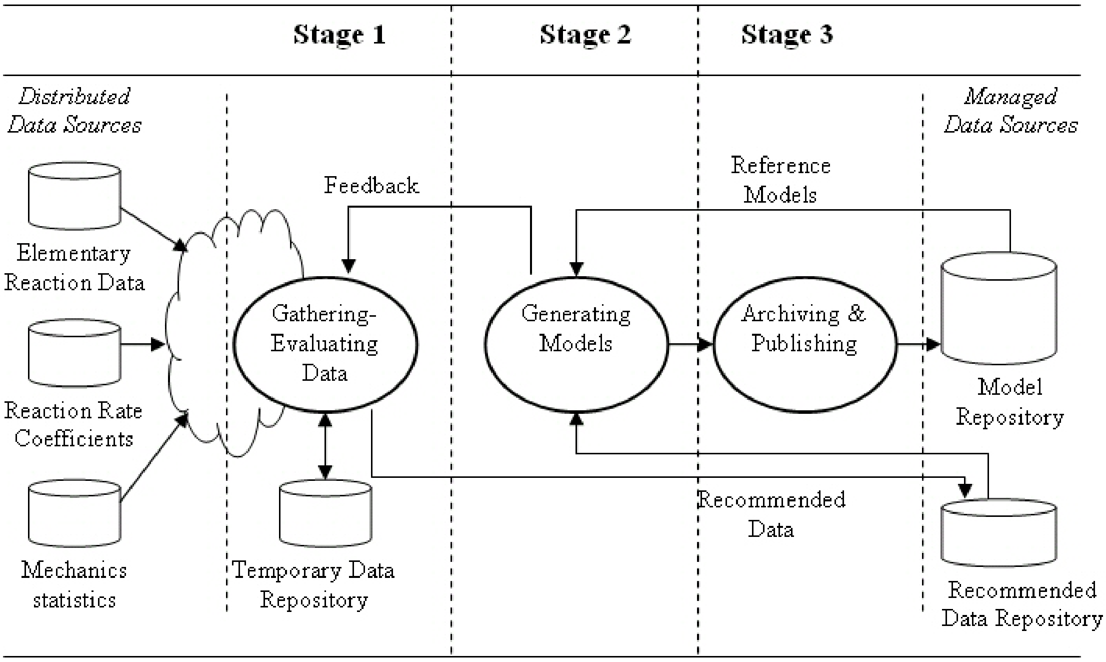


Figure 4.1: The three stage modelling process

i. *Stage 1 - Gathering and evaluating data*: Data gathering and evaluation was an essential stage of model construction. At this stage, elementary reactions would be identified together with their reaction rate coefficients and thermodynamic data required for the reaction mechanism of the model. These data could be produced by different research groups in the community and scattered over various sources. The data needed to be collected and evaluated. Validated datasets could then be archived as recommended data for use in later stages.

ii. *Stage 2 - Generating mechanisms and models*: At this stage, relevant elementary reactions and their associated parameters gathered and evaluated in the previous stage would be put together to build a mechanism. The resulting mechanism would then be put into a model. This usually comprised ordinary differential equations describing the chemistry and partial differential equations describing the fluid dynamics. The model could then be tested in a variety of ways, for example, through experiments on a flame, in which the concentrations of some of the species would be directly measured and checked against those simulated using the mechanism. A sensitivity analysis could also be conducted. This was to determine the sensitivity of an important observable to the mechanism components, e.g. the rate coefficients, to allow the experiments to be targeted at key features of the mechanism. The results of sensitivity analysis would provide essential feedback to the overall model development process.

iii. *Stage 3 - Publishing and archiving new models*: The resulting model from this process would be published so that other researchers and potential model consumers could be aware of its availability. The new model would also be archived for later use by application engineers or by other modellers as a referenced model. For archiving purposes, standards for data formats were essential, so that the archived models could be easily retrieved and used without the need for any conversion.

4.1.3 Limitations and Issues

The limitations and issues in terms of collaborations were identified through discussions with the reaction kinetics community members in Leeds, especially with a leading expert on reaction kinetics at the School of Chemistry - University of Leeds, who was also one of the founders of the PrIMe consortium. Some of these concerns also appeared on the white paper of PrIMe (PrIMe n.d.) and the Collaboratory Multi-Scale Chemical Science (CMCS) technical report (CMCS 2004).

One major problem that the reaction kinetics research community was facing was that the data required for the generation of reaction mechanisms and models were scattered in the community and often inadequately evaluated. There were different groups working on different reactions and aspects of data required for the modelling process. There might also be two or more groups working on the same reactions and datasets. However, there was little coordination across the groups. The organisation of research topics amongst these groups was unstructured. Therefore, it was often the case that datasets produced by different groups were different, even though they were all working on the same reaction. There was the need for evaluation to select the best datasets as recommendations for use in later stage of the modelling process.

From time to time, international panels consisting of experts in the field were set up to evaluate these kinds of data. Typically, they met once a year in a peer-review meeting. During the meeting, participants discussed submitted datasets and recommended the best datasets, which would then be deposited into a database and used as reference data. This process was time-consuming and costly.

There had also been efforts by some research groups in the community to collect, evaluate and archive available data to databases. Examples of collected databases were the Master Chemical Mechanism at the School of Chemistry of Leeds University (Rickard, Andrew and Pascoe, Stephen 2006) and the NIST (National Institute of Standards and Technology - USA) Chemistry Webbook (NIST 2005). However, the data collected and stored in these databases were not evaluated. There were also examples of evaluated databases, such as the “evaluated kinetic data for Combustion modeling” in Combustion (Baulch et al. 2005) and the database of “International Union of Pure and Applied Chemistry” (IUPAC) for Atmospheric Chemistry (Hynes & Cox 2006). The former was only available in hard copy, while the latter could be accessible from a website. However, such efforts were still patchy. In PrIMe, the scope of evaluation would be broken down to smaller sets of reaction. Each set would then be evaluated by a workgroup, operating remotely.

Another issue that concerned the community was the existence of many different formats for the same set of data. This was also a result of a lack of collaboration and coordination amongst research groups in combustion. Many different tools (e.g. for simulations, analyses or editing model data) were used in the community, and were often built by individual groups to meet their own needs. The formats of data were therefore customised to the habits and conventions of the groups that built the tools. Furthermore, different versions of the same tool might also be used at the same time. As a result, different researchers, or research groups might use and produce data using different formats and standards. That made the data transfer from one group to another group more difficult. Format conversion tools were often necessary.

Finally, the construction of a reaction model often involved hundreds to thousands of species and required a large set of ordinary differential equations. Solution of these equations would require a substantial amount of computational resources. The platforms on which these tools were running were usually personal computers or workstations in clusters. Therefore, it sometimes took hours to days or even weeks to complete.

4.1.4 Requirements for a Supporting Collaborative Infrastructure

In summary, this case study aimed to address two areas of concerns: the lack of coordination amongst distributed research groups and insufficient computational support for construction of reaction models. The lack of coordination made the gathering and evaluating data during Stage 1 and the publication of new data in Stage 2 of the modelling process become difficult. Insufficient support for computational resources limited the model construction capability required during Stage 2 of the modelling process.

The reaction kinetics research community expressed their interests in a collaborative infrastructure to support their distributed collaborations. In particular, the infrastructure should:

- Allow scientists who are working on the same or similar research activities to dynamically form working groups (small focused groups) to make the data transfer process from one research group to another easier and smoother.

- Provide efficient support for timely collaborations within and across working groups in the community for sharing expert knowledge, day-to-day working data, such as experimental data, chemical reaction mechanisms and related input data for reaction modelling to speed up the data collection and evaluation process.

- Provide easy access to computational intensive resources for time and resource consuming simulations and analyses and for storage of large amount of experimental data deal with large amount of calculation and storage required by the community.

4.2 An Application of the CeSA for the Reaction Kinetics Community

This section explains how different functions of the CeSA might help to address the limitations and issues related to distributed collaborations within reaction kinetics research community.

4.2.1 Mapping the CeSA

The Figure 4.2 shows an overview of a realisation of the CeSA on the reaction kinetics community and two of its closely related research communities: the combustion and atmospheric chemistry communities. In these communities, there may be members that are more data oriented (denoted as Data Nodes in the figure), while others are ordinary researchers (i.e. Community Nodes). As shown in the figure, different research communities, i.e. combustion, reaction kinetics and atmospheric chemistry, can jointly operate in one P2P collaborative environment. Working groups can be formed in the environment, even across community boundaries. Usually, each working group has a group coordinator (shown as Workgroup Coordinators in the figure). In the P2P collaborative environment, a member can seamlessly communicate with another member across working groups and communities (illustrated by straight thin lines).

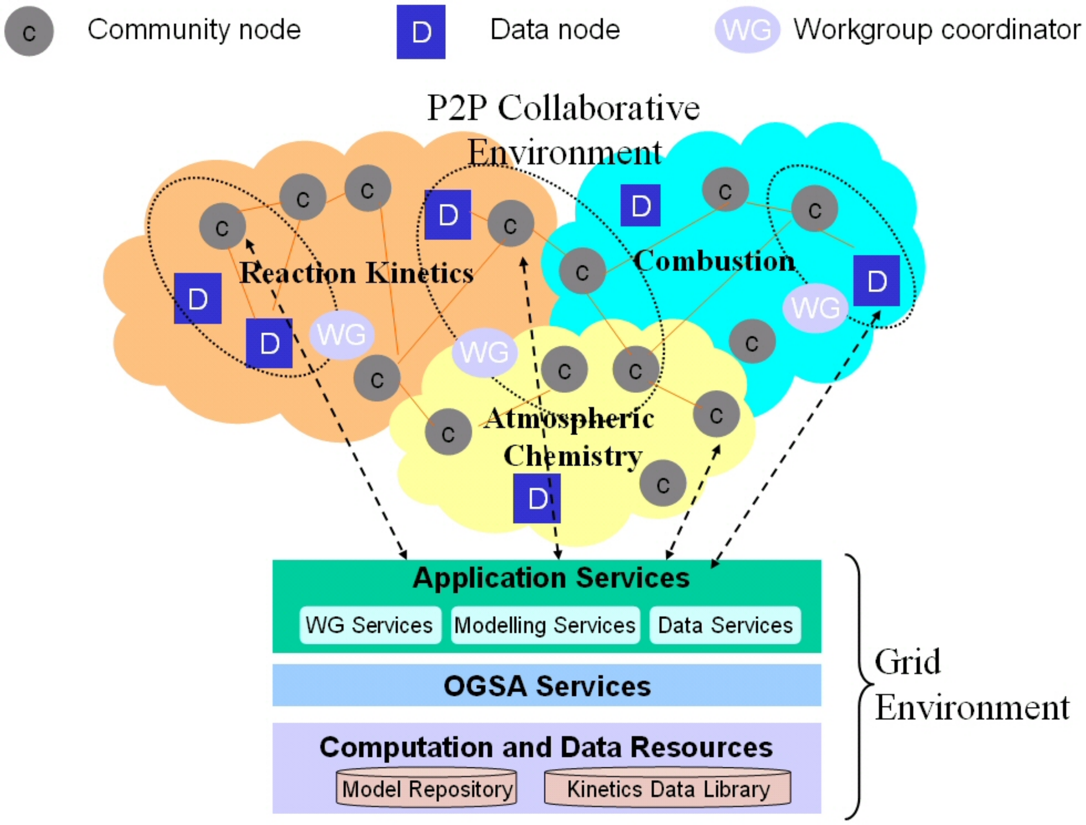


Figure 4.2: Application of the CeSA for reaction kinetics and its related research communities

The underlying Grid supports the community with back-end computational and data resources. Access to these Grid resources is made possible via services designed specifically for the research communities (illustrated by dashed bidirectional arrows).

4.2.2 Addressing the Limitations and Issues

Making scattered data easily accessible. Data scattering problem (as discussed in Section 4.1.3), can be addressed by providing a P2P file sharing and resource discovery functions in the implementation of the CeSA. Datasets produced by working groups or individuals usually reside on the groups' or individuals' personal storage. Usually, only the final, well-prepared versions of selected data are published. Majority of the earlier datasets are hidden from outsiders, although these datasets may be very important to some other groups. Through P2P file sharing and resource discovery process, these datasets can become visible to others. As a result, the data gathered for a modelling process is much richer. In a reversed direction, data publication is also easier in a P2P environment. Newly produced data can be made available to other members of the community.

Identifying expertise for potential collaborations. In addition, data held on a researcher's storage reflects his/her interests and expertise. Similarly, it reflects working areas of his/her working group(s). Knowing the interests of researchers and/or working groups before they actually publish their research data can bring about potential collaborations at an early stage of research. This speeds up the collaboration process, particularly in terms of data transferring and resource sharing across community boundaries.

For example, a group working on modelling of a chemical reaction model (e.g. process of burning methane), will need to know the research data about elementary reactions involved in the burning process, such as thermodynamic data, structural properties, the reaction rates. If they know that other working groups are working on the related data, a collaboration with these groups can be set up. The benefit for the modelling group is that they can produce the most up-to-date models. The groups working on the input data also benefit from the modelling group as their data will be validated and used at an earlier stage in the data creation process. Time required for review and validation is likely to be reduced.

Supporting the modelling process with computational and data resources. Computational, storage and data resources for the modelling process are provided to the communities in form of services from the Grid environment. As shown in Figure 4.2, potential services are Workgroup (WG) Services, Modelling Services and Data services:

WG Services are services that support collaborations amongst members of community in the P2P environment. Although *ad hoc* collaborations mainly happen in the P2P collaborative environment, occasional support from Grid may be necessary. Examples of WG Services are Shared Storage Services, WG Information Services and WG Authentication Services

Modelling Services provide computational capability for constructing reaction models. Examples of Modelling Services are Model Simulation Services, Model Optimisation Services, Model Reduction Services and Model Verification Services.

Data Services handle Grid-based data resources involved in the modelling process, such as experimental data, reaction rates, mechanics statistics and combustion models. Capability of Data Services is enhanced by metadata standards and ontology. Examples of Data Services are Data Publishing Services, Data Archiving Services and Data Validation Services.

The use of common sets of shared services together with standardised metadata and shared ontologies will also help to reduce the number of data formats. It will make the data transferred across platforms, working groups and communities easier and smoother. The time and efforts required for unnecessary conversion will be reduced.

In summary, the functions of the CeSA can potentially deal with various requirements of the reaction kinetics research community. Especially, with the P2P collaborative environment, the collaborations amongst reaction kinetics and its related research communities can easily be extended across the community boundaries.

4.2.3 A Prototype Implementation of the CeSA

This prototype was a scoped implementation of the CeSA as outlined in Figure 4.2. It facilitates the identification of possible technical challenges that need to be resolved for a successful implementation of the CeSA. This version of the prototype was also used in the user evaluation (to be reported in Section 4.3).

This prototype was developed using JXTA technology (Gong 2002, Project JXTA 2003) and Globus Toolkit version 3.0.2 (The Globus Alliance 2006), which was an implementation of OGSA for the P2P applications and computational services respectively. One of the main reasons for choosing JXTA as a platform for this prototype was its concepts of Peer and Peer Group. They matched with individuals and work groups, respectively, in scientific communities. Another reason was that Project JXTA was open source. It would be possible to modify the JXTA environment to meet the needs of the CeSA. At the time of prototyping, the Globus Toolkit version 3.0.2 was introduced and an implementation of the OGSA was available (Foster et al. 2002). For this reason, this version was chosen as the platform for developing services in the Grid environment for the prototype.

4.2.3.1 Application Services for Chemical Reaction Modelling

CHEMKIN had been a typical application package used in the community for simulations and analyses of chemical reaction mechanisms during a modelling process. A few programs in CHEMKIN package[2] and a related program, KINALC [3], were wrapped into Grid services. These applications commonly used files as input and output, and could also produce console output. When wrapping these programs into Grid services using Java, input and output (including console output) were mapped to the input and output parameters of Grid services.

The unified standard service interface, as required by the CeSA SOA specification, used for these Grid services were developed based on the common characteristics of input and output of simulation and analysis programs in CHEMKIN package. More specifically, the service interface consisted of the following operations:

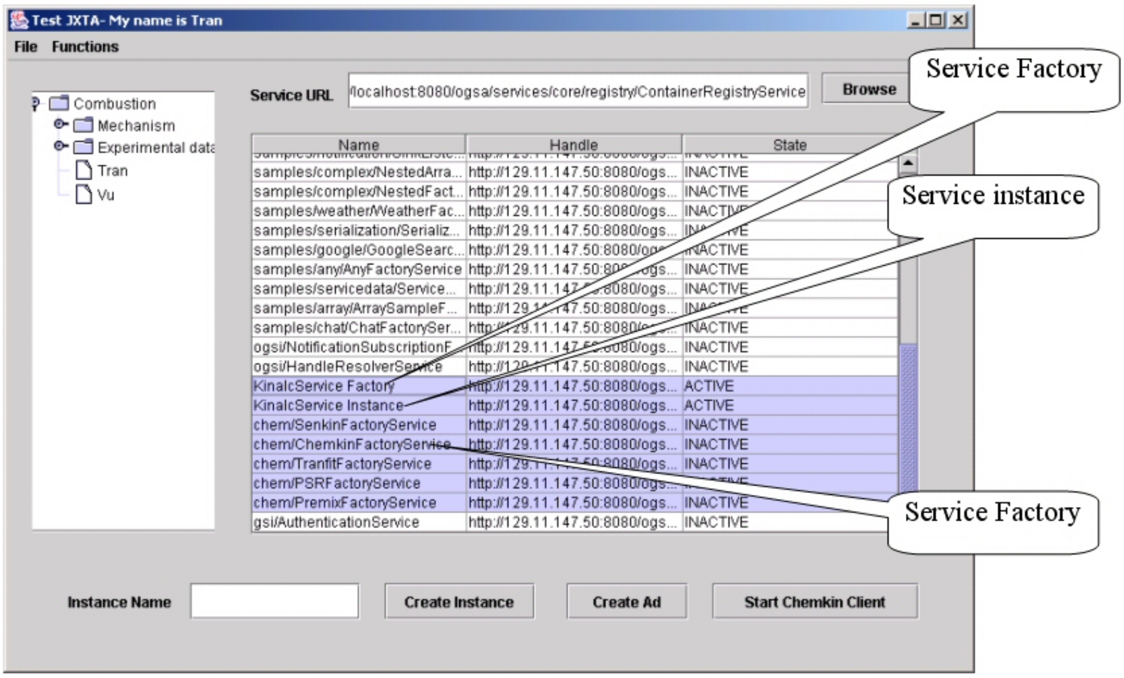
- *List input required:* allows service client to query in advanced input files required.
- *Load input:* to upload input files required to server prior to execution.
- *Execute:* to run the services after all required input uploaded.
- *List output:* to query number of outputs produced by the service.
- *Transfer output:* to send back outputs to users.

After being wrapped into Grid services, these new services were deployed into a Grid Service Container provided in Globus Toolkit version 3.0.2. This Grid Service Container ran on one machine, played the role of a computational grid providing simulation and analysis services for the P2P community.

The list of services for simulations and analyses in Reaction Kinetics research wrapped from CHEMKIN programs and KINALC are highlighted in Figure 4.3. As these services were OGSA based Grid services, there were service factories, services instances as shown in the figure.

[2] The version of CHEMKIN wrapped into Grid services was licensed to be used within the Reaction Kinetics research group in Leeds only. Therefore, the Grid services would only be available the Leeds group.

[3] Developed by the Reaction Kinetics group at School of Chemistry, University of Leeds



Service URL: /localhost:8080/ogsa/services/core/registry/ContainerRegistryService

Name	Handle	State
samples/complex/NestedArra...	http://129.11.147.50:8080/ogs...	INACTIVE
samples/complex/NestedFact...	http://129.11.147.50:8080/ogs...	INACTIVE
samples/weather/WeatherFac...	http://129.11.147.50:8080/ogs...	INACTIVE
samples/serialization/Serializ...	http://129.11.147.50:8080/ogs...	INACTIVE
samples/google/GoogleSearc...	http://129.11.147.50:8080/ogs...	INACTIVE
samples/any/AnyFactoryService	http://129.11.147.50:8080/ogs...	INACTIVE
samples/servicedata/Service...	http://129.11.147.50:8080/ogs...	INACTIVE
samples/array/ArraySampleF...	http://129.11.147.50:8080/ogs...	INACTIVE
samples/chat/ChatFactorySer...	http://129.11.147.50:8080/ogs...	INACTIVE
ogsi/NotificationSubscriptionF...	http://129.11.147.50:8080/ogs...	INACTIVE
ogsi/HandleResolverService	http://129.11.147.50:8080/ogs...	INACTIVE
KinalcService Factory	http://129.11.147.50:8080/ogs...	ACTIVE
KinalcService Instance	http://129.11.147.50:8080/ogs...	ACTIVE
chem/SenkinFactoryService	http://129.11.147.50:8080/ogs...	INACTIVE
chem/ChemkinFactoryService	http://129.11.147.50:8080/ogs...	INACTIVE
chem/TranfitFactoryService	http://129.11.147.50:8080/ogs...	INACTIVE
chem/PSRFactoryService	http://129.11.147.50:8080/ogs...	INACTIVE
chem/PremixFactoryService	http://129.11.147.50:8080/ogs...	INACTIVE
gsi/AuthenticationService	http://129.11.147.50:8080/ogs...	INACTIVE

Figure 4.3: A list of Grid services for simulations and analyses in Reaction Kinetics research

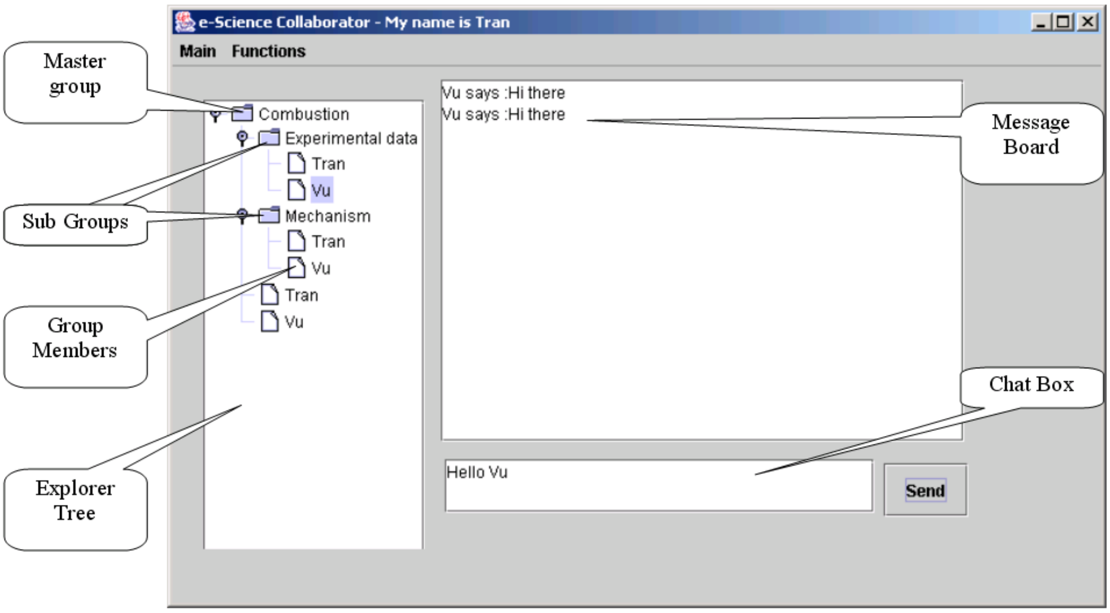
4.2.3.2 The e-Science Collaborator: A Peer-to-Peer Application

The P2P application of the CeSA was named as "e-Science Collaborator". The following functional components of the CeSA were implemented:

User Interface. The user interface was developed for the e-Science Collaborator using purely Java. The main window of the user interface of the e-Science Collaborator is shown in Figure 4.4. The left hand side of the main window (and also in other windows) showed a hierarchy of user groups (or communities). Content displayed on the right hand side was specific to the function being selected. In this figure, the message board, text box and send button of instant messaging function were displayed. Other screen shots of the user interface will be presented together with the functions they were designed to support.

Peer Core. This core component was based mainly on JXTA's core middleware. Small development was done to make use of JXTA communication pipes for instant messaging and file sharing services.

Grid Service Client. The Grid Service Client consisted of two parts: a service browser and a service executor.



Vu says :Hi there
Vu says :Hi there

Hello Vu

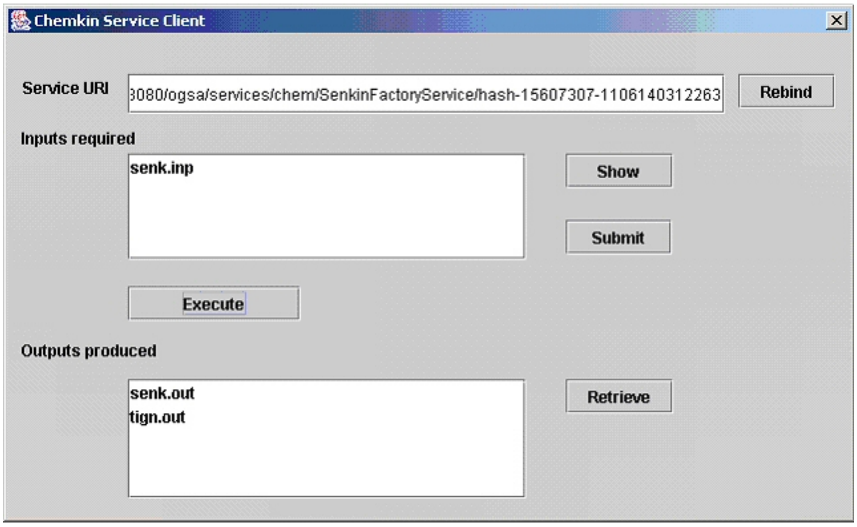
Figure 4.4: The main user interface window of the e-Science Collaborator

The service browser (shown in Figure 4.3) had an interface that enabled end users to browse services available from a particular Grid Service Container. The end users would need to provide the URI (Universal Resource Indicator) of the container's Registry Service. Through interaction with the Registry Service, the browser would display a list of service handlers of the services provided by the container. The service browser also had functions for creating new instances of a service from a service factory handler and for starting service execution interface to execute an instance of a service (using Start Chemkin Client button as shown in the Figure 4.3).

The service executor could invoke a service from a service handler. The version of Globus Toolkit used for prototyping had two types of service handler for a Grid service. One type of handler was for the Factory Service and the other type was for the Grid service instance. The service executor could be used to generate new instances of a service from a factory service handler or to execute the service using the handler of a service instance. The service executor could interact with any the services developed previously for chemical reaction simulations as they all conformed to the unified standard service interface specified previously. The unified execution interface is shown in Figure E.8.

Service Publication and Discovery Agent. The method used for publication and discovery of service information was based on JXTA protocols (Project JXTA 2003).

All information about a service, such as service name, service provider, input and



Chemkin Service Client

Service URI 3080/ogsa/services/chem/SenkinFactoryService/hash-15607307-1106140312263

Inputs required

senk.inp

Outputs produced

senk.out
tign.out

Figure 4.5: Service execution interface of the service client

output was enclosed in a JXTA advertisement. The advertisement about the service was then published in JXTA P2P network using JXTA discovery protocol.

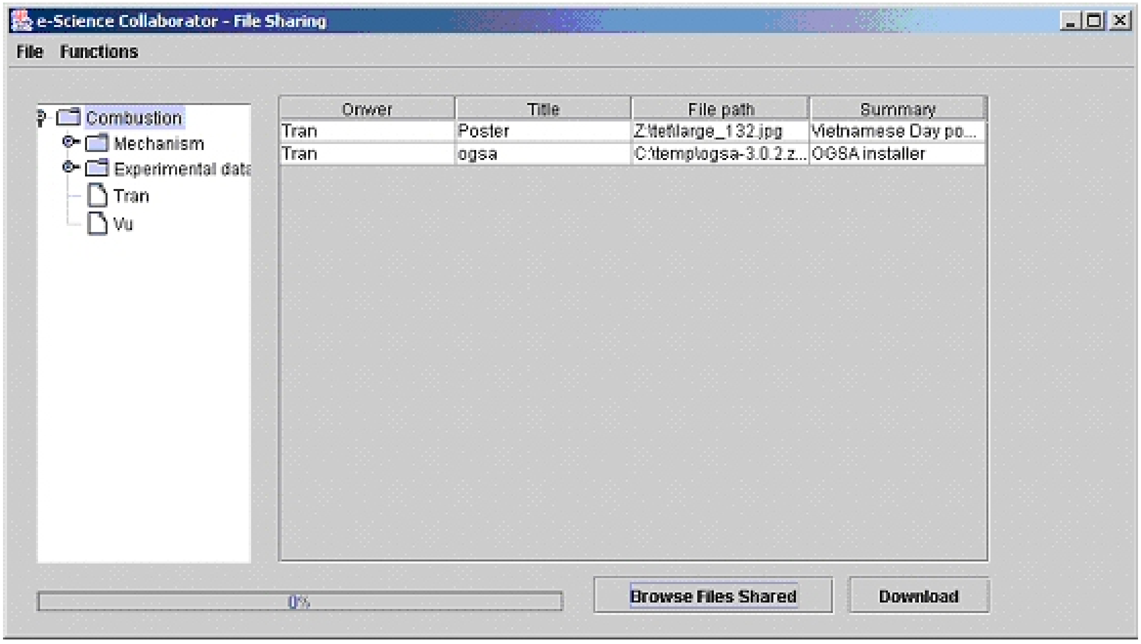
The discovery of information about services, however, was not based on JXTA discovery, but using JXTA resolver protocol, because the default discovery mechanism provided with JXTA discovery protocol (Traversat et al. 2003) was not flexible enough to deal with complex query requirements. In the prototype, service discovery was required to allow end users to make query using any information about services or a combination of them, whereas, with the default mechanism provided by JXTA discovery protocol, only a few indexed attributes could be searched. With JXTA resolver protocol, a query could be distributed to the other peers in the environment. On receiving the query, a peer would flexibly search through its cache for service advertisements that matched the criteria specified in the query, such as service name, service provider, etc. The results would be sent back directly to the query issuer.

Community Services. The following functions were implemented for Community Services:

- Tools for managing work groups and communities
- File sharing
- Instant messaging
- Resource discovery

Tools for managing working groups and communities were developed upon JXTA Peer and Peer Group concepts. An individual peer user was mapped to a peer in the JXTA P2P network. Similarly, a working group or community is corresponding to a JXTA Peer Group. Example displays of organisation of peers and peer groups are positioned on the left side of Figure 4.4. In these figures, the folder icons represented groups or communities (Peer Groups). The file icons stood for individuals (Peers).

Tools for file sharing and instant messaging were built on communication infrastructure provided by the Peer Core. To send a message or a file from one peer to another peer, a connection channel between the two peers would be set up first. Then, the message or file would be sent over this channel. Figure 4.6 presents a snap shot of an interface of file sharing component of the e-Science Collaborator.



Onwer	Title	File path	Summary
Tran	Poster	Z:\tet\large_132.jpg	Vietnamese Day po...
Tran	ogsa	C:\temp\ogsa-3.0.2.z...	OGSA installer

Figure 4.6: A snap shot of a file sharing interface. The table in this figure shows a list of files shared by Combustion group selected on the left.

There were also additional components for managing share relationships amongst peer users and working groups or communities and for searching for shared resources. Shared resources available on a peer, such as a working data file, were more dynamically managed, not as static as a shared service. A file could be set to share to a group at a particular time, but not at the other time. Therefore, the approach to resource discovery, more specifically file search, was different from the method used for publication and discovery information about services. There was no publication. As only the owner of the resources could say whether it had resources being shared for the query issuer, the query message had to be distributed to every potential resource owner. The current version of the prototype was using broadcasting method to distribute query messages. Scope of queries could be limited within particular working groups.

4.3 User Evaluation

The evaluation was focused on two major sets of functions provided by the CeSA: (i) the use of P2P file sharing for collaborations and (ii) the use of remote services for simulations and analyses. P2P file sharing was chosen to evaluate as it had strong influence on resolving the data scattering problem of the reaction kinetics research community. This is part of Stage 1 of the modelling process. For a similar reason, remote services are essential in supporting the generation of reaction models at Stage 2 of the process, as described in Section 4.1.2.

4.3.1 Objectives

The aim of this evaluation was to conduct qualitative assessment on the usability and acceptability of the CeSA by a sample of potential users. As this was a very early evaluation on the CeSA, this study was focused only on how potential users react to the two sets of collaborative functionalities provided by the CeSA, as discussed above, and their opinions on the capability of the CeSA to support their work. More specifically, the objectives of the evaluation were:

- To evaluate the usability of P2P file sharing function to support collaborations within the user community. The kind of collaborations was the sharing and exchanging of day-to-day working data.
- To assess how users can benefit from the access to remote simulations and analyses in Reaction Kinetics using Grid services. This includes the method for publication and discovery of information about services in P2P environment.
- To assess general attitudes of potential users to the P2P collaborative environment provided by the CeSA. This would be the basis for further analysis on the acceptability of the CeSA.

Findings from this user evaluation would be used to form further requirements for the next design iteration on the CeSA.

4.3.2 Evaluation Criteria and Data Collection Method

The following are criteria for evaluating the CeSA using the prototype. These criteria were based on characteristics that have been claimed to be advantages of the CeSA. The criteria were organised into two groups, corresponding to two sets of functions provided by the CeSA to be evaluated. No specific criterion was used for general feedback on the prototype system, as the questions were open and designed for getting participants' overall impression to the system.

For evaluating P2P file sharing function for collaborations, criteria used were:

- Ease of making a share: in terms of time and steps required
- Control over shared resources: the control was measured based on end users' ability and flexibility to share a file or not to share a file
- Duplication of shared data
- Overall comparison with the current way of working to carry out the same task
- Overall impression on the P2P file sharing functionality

The evaluation of using remote services to support the modelling process used the following criteria:

- Suitability of using remote services for simulations and analyses
- User preference on using remote services
- Potential benefits
- Protection of intellectual properties
- Willingness to share tools for simulations and analyses as remote services
- Convenience of discovery and publishing information about shared services

To evaluate the above criteria, questionnaire (attached in Appendix C) was used to capture the participants' feedback during the evaluation process. It consisted of a mix of 30 open and closed questions, organised into four sections. The first section of the questionnaire consisted of questions related to the use of P2P file sharing function as a means of collaborations. The Section 2 and 3 were about using remote services for simulations and analyses. The fourth, also the final, section was to collect participants' general feedback on the system. Each of the first three sections, which were directly related to the two sets of functionalities to be evaluated, consisted of 3 types of questions:

- Questions to capture participants' current way of working to perform the (equivalent) work specified in the questions,
- Questions about potential benefits (or any drawback) of the function(s) being evaluated, in comparison with the current working practice, and
- Questions to capture participants' recommendations for improving the functions being evaluated.

Contents of these questions were focused on the evaluation criteria specified above. As open questions were used, for each criteria, there could be more than one related question. On the other hand, responses to one question could be an implication to more than one criterion. Table 4.1 shows a mapping between criteria and questions in the questionnaire.

Functions	**Criteria**	**Questions**
P2P File Sharing	*Ease of making a share*	Q3, Q4
	Control over shared resources	Q3, Q5, Q6
	Avoiding duplication of data being shared	Q3,Q7
	Comparison with current way of working	Q1, Q2, Q3
	Overall impression	Q8
Using remote services	*Suitability of running simulations and analyses as remote services*	Q11, Q12, Q13, Q14, Q15, Q16
	Preferences on using remote services	Q17, Q18, Q19, Q20
	Recognition of potential benefits	Q17, Q18, Q19, Q20
	Importance of the protection of intellectual property with regard to sharing computation resources as services	Q19
	Willingness to share tools as services	Q19, Q21
	Convenience of discovery and publishing information about shared services	Q23, Q24, Q25, Q26, Q27

Table 4.1: The mapping between evaluation criteria and questions in the questionnaire

The fourth section included only questions (Q28, Q29 and Q30) about the participants' overall impression on the prototype system and their recommendations for the next version of the CeSA.

Each of the above criteria was graded using the following three point scale, based on the analysis of participants' free-text responses collected from the questionnaire:

- *positive*: for responses that supported the functions of the CeSA being evaluated in relation to the criteria.
- *neutral*: for responses that did not say either support or not support the functions of the CeSA in relation to the criteria, or for responses that contained a mix of supporting and non-supporting statements.
- *negative*: for responses that did not support the functions of the CeSA in relation to the criteria

4.3.3 The Evaluation Process

The evaluation was conducted at a Reaction Kinetics research laboratory at The University of Leeds. The researchers in this laboratory could be potential users of the new system.

Three chemists participated in the evaluation. At the time of evaluation, they were working on atmospheric chemistry, an application area of reaction kinetics. Two of the participants were research fellows. The third person was a PhD student. The three participants are referred to anonymously as participant 1, participant 2 and participant 3 in subsequent discussions.

Three copies of the e-Science Collaborator were installed on three different computers in the laboratory for the evaluation. Each participant was provided with a documentation (attached in Appendix E) that guided him/her through the functions of the system to be evaluated. Firstly, the participants were guided to walk through the P2P file sharing functions of the e-Science Collaborator to perform the following file sharing activities:

- Share a file
- Browse files shared by all users of a group
- Revoke a file from sharing
- Search for a required file shared by other users

Secondly, in order to evaluate the use of remote services, a number of Grid services for simulations were provided to the P2P environment. The Grid Service Container was running on one machine at another department in the university. The participants were guided to get access to remote computational resources on a Grid via Grid services using the Grid Service Client of the e-Science Collaborator.

All the participants were using the system at the same time, with the presence of the author. They filled in the questionnaire when they walked through the system. There were also discussions with the author during the evaluation to clarify the questions in the questionnaire.

4.3.4 Results and Analysis

The whole evaluation session took approximately 3 hours. Answers to the questionnaire are attached in Appendix D. They are summarised and organised according to the two main functions of the CeSA being evaluated and the general feedback on the e-Science Collaborator.

4.3.4.1 P2P Collaborations Using File Sharing Function

The results collected showed that all the participants had a need to share working data with their colleagues. The most popular methods were using email, via a shared area on the laboratory's computer network, using collaborative workspaces hosted by an external website. The workspace on British Atmospheric Data Centre (BADC) website was used by all participants. One of the participant also used FTP sites for exchange of data.

All of the participants identified the inconvenience of using a centralised web site for sharing or exchanging working. When using a centralised website, a file being shared needed to be uploaded to the server (answers to Q3). Then, any user, who needed to use the file, had to download the file to his/her local machine. When sharing a file this way, a file being shared might not be the most up-to-date as the provider might be deterred from uploading new versions frequently if the demand is unclear. This was seen as inconvenient by all the participants.

In terms of usability of file sharing function of the prototype, all participants recognised the benefits of using the CeSA prototype system for sharing working data because of the following reasons (answers to Q3, Q4 and Q5):

- A file could be shared directly from the user machine. Therefore, there was no need to move a file around for sharing.
- Spontaneous sharing of file-in-progress was possible. This allowed other users to be able to copy the latest version of the file.
- There was no need to maintain multiple copies of the data for sharing

- Users had control over shared data. They could choose to share with a group or with a specific person. They could also easily revoke a file from sharing.

Following are feedback from the three participants, respectively, when they were asked about their overall impressions on P2P file sharing functionality of the CeSA (Q8):

> *"I like it, I would certainly use it for certain applications. The new share function would save time and space and be more convenient. However, security would be an important issue."*

> *"Useful to be able to share files from own computer without having to copy to shared directory, disks etc."*

> *"It is quite useful - It is much quicker to share data especially if data needs to be worked on by several people. It is easy to exchange copies of the updated work. On the BADC it is just the original raw data that is shared and then everything else is worked on separately. This way of sharing does seem more useful. Would save a lot of time - both in the sharing process and working on data."*

A summary of the feedback collected is shown in Table 4.2. The criteria and the evaluation scale used in the table are presented in Section 4.3.2. The results are graded based on an analysis on feedback to corresponding questions of the questionnaire shown in Table 4.1. The results summarised in this table shows some of the advantages of using P2P computing model for sharing day-to-day working data in the CeSA.

Criteria	**Par. 1**	**Par. 2**	**Par. 3**
Ease of making a share	positive	positive	positive
Control over shared resources	neutral	positive	positive
Avoiding duplication of data being shared	positive	positive	positive
Comparison with current way of working	positive	positive	positive
Overall impression	positive	positive	positive

Table 4.2: Summary of participants' feedback on P2P file sharing functionality of the CeSA. (Par. is used to refer participant for short.)

The recommendation for next design iteration on the CeSA included security of P2P environment, which was expressed by one participant. When the access controls are decentralised to individual machines, it will be important to provide a security mechanism to protect user's own computing resources and personal data. This participant also identified the need for a mechanism for tracking changes in datasets being shared. Another participant expressed the need for better information about shared resources, such as who had downloaded the data. Version control of shared data was also mentioned. The other participant suggested an idea for improving sharing mechanism of the user interface of the prototype.

4.3.4.2 Using Remote Services for Simulations and Analyses

All participants had a need to run some kind of simulations or analyses for their research. FACSIMILE[4] was an analysis program that was commonly used by all three participants. One of the participants also used some other programs, such as MECHGEN[5], for analyses and simulations. These programs (i.e. Facsimile program) ran on desktop machines. The input and output of these programs were usually in form of text files, though they could be in different formats. Some programs (e.g. MECHGEN) could be run remotely via Telnet. Running time of Facsimile program for an analysis could be minutes to weeks.

In the evaluation, after using sample Grid services provided by the prototype, all the participants agreed that the way of running simulations and analyses as remote services as in the prototype was suitable for them. Here is a feedback from one of the participants to question Q16 of the questionnaire:

> *"Running simulations in the way would work for both MCM*[6] *and TUV*[7] *as long as you had control of the input files. Sometimes you would require many input files for each simulation therefore would require some kind of indexing system. Also it would be desirable to run multiple simulations simultaneously."*

A common benefit of using remote services for simulations and analyses recognised by all participants was that it would free up their desktop computers' resources for other purposes, especially for large simulations, that required hours to days to complete (answers to Q17).

In terms of sharing remote services, all the participants agreed that intellectual property protection was an important factor of sharing remote services, although one participant responded that it might not be useful in some cases (answers to Q19). They would

[4] A computer software for modelling processes and chemical reactions

[5] Computer Aided Generation and Reduction of Reaction Mechanisms

[6] Master Chemical Mechanism

[7] Tropospheric Ultraviolet and Visible

be encouraged to share their self-built tools as remote services if this issue was well addressed. When asked about preference on sharing resources in general, the participants recognised the importance of sharing resources in their working environment and agreed that they would be happy to share more of their own resources if they received more (Q21). However, when referred specifically to sharing research tools (Q20), such as a simulation program, as remote services, there were different feedbacks from the three participants:

- The first participant was worried about the amount of computing resources required for sharing services and man power to maintain the system, although recognised the benefit of sharing. Software licensing was also one of his concern.
- The second participant preferred to share a copy of the program, as it would need less computing resources required for sharing. The reason for this preference could be because the participant thought that she would need to use her own computing resource on her desktop machine to make the share.
- The third participant had preference on sharing tools as remote services as the tools could be improved over time.

There was only feedback from one participant on recommendation for improving next version of the e-Science Collaborator (Q22). The recommendation included support for submitting multiple jobs simultaneously, better indexing and documentation of input and output files and visualisation of job queues.

When asked about the comparison between the use of the prototype to discover/use remote services and their current way of working, generally, all the participants said that the prototype was more useful. The advantages recognised were that it allowed information about services to be published directly to particular groups and that once the information about a useful service was found, it could be used directly as the advertisement linked directly to the service it advertised (answers to questions Q25, Q26 and Q27).

Feedback on the use of remote services was summarised in Table 4.3, using criteria described in section 4.3.2.

4.3.4.3 General Feedback

On the general feedback, all three participants said that P2P file sharing function was the most useful in the prototype. Here was what they said:

> *"Good, file sharing would be most useful."*

Criteria	Par. 1	Par. 2	Par. 3
Suitability of running simulations and analyses as remote services	positive	positive	positive
Preferences on using remote services	positive	positive	positive
Recognition of potential benefits	positive	positive	positive
Importance of the protection of intellectual property with regard to sharing computation resources as services	positive	positive	positive
Willingness to share tools as services	neutral	negative	positive
Convenience of discovery and publishing information about shared services	positive	positive	positive

Table 4.3: Summary of participants' feedback on using remote services for simulations and analyses. (Par. is used to refer participant for short.)

> *"The sharing file function would be most useful in day to day use, but also function have possibilities and potential in my work."*

Generally, the participants' feedbacks on the functionality of the CeSA were positive. They all saw the potential of the CeSA for an application to support their research community. The following were their sayings about the potential:

> *"As a first basic prototype the potential of such a system is clear to see. All of the facilities added so far show promise. I would like to see the remote service operated using multiple simulations simultaneously and get a better feel as to how easy it would be to use."*
>
> *"I think that our group would certainly use such a system if it proved to be the way forward in e-science (which I feel it is) and the scientific community embraced the use of such a system."*
>
> *"A fully working system would benefit the atmospheric chemistry group, provided it was widely accepted by the whole community."*

4.4 Summary and Reflections

This chapter has discussed the requirements for a collaborative architecture through the study of reaction kinetics research community. These requirements included the need for collaborations at the user end as well as the need for access to large computing resources. The case study has also illustrated how the CeSA can be applied to address these requirements. In this chapter, a prototype implementation of the CeSA has also been described. A user evaluation on the CeSA using the e-Science Collaborator, a prototype of the CeSA, has also reported.

The result of the evaluation has provided positive feedback to the potential of the CeSA, especially on the use of a P2P collaborative environment for collaborations within the user community and the access to remote simulations and analyses using Grid services. As a result of the evaluation, a number of new issues emerged: security in the P2P environment, user interface and tools for documenting and tracking changes were raised. These issues will be considered in the design of the next version of the architecture.

There are also limitations in the user evaluation reported. Firstly, it is limited because the number of participants involved in the evaluation was not big enough to obtain a representative outcome. Secondly, the version of the e-Science Collaborator used in the evaluation was implemented with very basic functions of the CeSA. These functions were evaluated relatively independent. Therefore, for a richer evaluation, where different functions of the CeSA can be evaluated collectively in a more realistic collaborative environment, there is the need for further user evaluation using a richer implementation of the CeSA.

Chapter 5

Adaptive Method for Resource Discovery in Peer-to-Peer Environment

This chapter focuses on a technical issue that needs to be resolved in order to successfully implement the CeSA. As specified in Chapter 3, the P2P part of the CeSA provides a collaborative environment, in which scientists can publish and share resources within other scientists in the community. The challenging issue is how to provide an efficient and scalable mechanism for resource publication and discovery in a decentralised P2P environment.

The chapter begins with an explanation on the importance of resource discovery in distributed environments in general. It then describes typical resource discovery requirements and characteristics of scientific communities. A review of a number of current popular approaches to resource discovery for P2P environments follows. There is also assessment on the applicability of the popular resource discovery approaches to the scientific domain during the review. The main body of this chapter describes an adaptive resource discovery method for the P2P collaborative environment of the CeSA. Results of simulations to evaluate the efficiency of this proposed approach are also reported.

5.1 The Importance of Resource Discovery in Distributed Environments

The importance of resource discovery can be seen through the history of the Internet. The development of search engines happened alongside the development of the Internet to deal with the very fast growing of resources available. In the early days, after the Internet was open up for educational and commercial purposes, the number of sites on the Internet was booming. It was not easy to manually keep track of resources of interest available on the Internet. The first effort to help locate relevant resources was Archie search engine, created in 1990 (Sonnenreich 1997). It collected resource available on anonymous FTP (File Transfer Protocol) sites and built a searchable index. After the invention of the World Wide Web (WWW) by Tim Berners-Lee in 1989 (Berners-Lee 2006), a number of Web search engines were launched. Amongst the first were Aliweb (1993), WebCrawler (1994), Lycos (1994) and Alta Vista (1995) (Sonnenreich 1997, Chu & Rosenthal 1996). The success of Yahoo and Google has shown how important resource discovery is in a distributed world. In 2005, Gulli & Signorini (2005) estimated that by the end of January 2005, there were about 11.5 billion Web pages index-able by major such engines such as Google, Yahoo! and MSN. The actual number of documents available on the Web, including those not indexed, would be much larger. Efficient resource discovery methods are really important in order to exploit such a huge resource in a distributed environment such as the Internet.

5.2 Typical Resource Discovery Requirements in Scientific Research Communities

Scientific communities have characteristics that make their requirements for discovery of scientific resources different from general P2P file sharing applications, which is currently the most common form of P2P applications. The two important characteristics that most influence on the requirements are the scientists' interests in those resources and the type of resources to be discovered.

5.2.1 Interests in Resources of Scientists

One of the characteristics of scientific communities discussed in Chapter 2 is the increasing demand for tackling multidisciplinary research problems. Resolving these problems requires a pool of resources from different research disciplines. Scientists involved in multidisciplinary research also need to be aware of the availability of these resources. Therefore, there is an increasing interest in resources from different disciplines in multidisciplinary scientists. At a particular time, a scientist who is undertaking multidisciplinary research may have many different interests in resources. The number of interests that the scientist has might depend on the number of research disciplines that he/she is involved.

e-Science community is an example. There is a wide range of interests held by its members, spanning across multiple disciplines. For example, scientists who are addressing a complex problem in biology may be interested in researching for a computing infrastructure to better support their work. Scientists who are doing research on the application of reaction mechanisms in atmospheric chemistry may have interests in reaction kinetics literature.

Furthermore, a scientist's interest in resource may also be changed over time, usually depending on the projects that the scientist is working on. A research project often lasts for a few years. When undertaking new projects, a scientist may be in need for resources that are relevant to the new research problems. As a nature of scientific research, the research problems addressed by different projects are different. Therefore, the kinds of resource that are necessary for different projects are usually different.

5.2.2 Types of Scientific Resources

The types of scientific resources are also different from one to another. For example, observed from the case study reported in Chapter 4, the types of resources to be discovered in scientific research communities are usually experimental datasets, working documents, journal papers or information about available tools and services. Each of these categories requires a different method for query matching. For example, to search for a research dataset, the information required could be authors and time of experiment. For journal papers, a search could involve a set of indexed keywords, authors or the paper's title. In the prototype application of the CeSA, to discover information about Grid resources (i.e. a Grid Service) a query might look for information about service providers, input and output parameters.

5.2.3 Implications on Resource Discovery

These two typical characteristics have a strong influence on designing of an efficient resource discovery method for the communities:

Firstly, a scientist's interests will have an influence on the way the scientist searches for resources and also on the types of resources that the scientist shares with his/her communities. The scientist may join to one interest group for resources on a particular interest, but another group for resources on another interest. The changes of interests of a scientist over time will also have an effect on his/her memberships at these groups. At a particular time, the scientist may be most interested in a particular interest group. However, when changing his/her interests, another group may be more important.

Secondly, different types of scientific resources might require different ways of matching the search queries and the resources. For example, when searching for journal papers, using a set of keywords to match against available journal contents or indexed keywords using full-text search might be enough. However, when search for a dataset stored in a binary format, full-text search might not be applicable. In such a case, a combination of methods, such as annotating binary data using metadata and building a complex search query with different search constraints for matching the semantics of the data content, might be necessary.

In summary, in order to efficiently support the need for resource discovery in distributed scientific communities, an efficient discovery method needs to:

- be scalable, a general requirement that any method of discovery in a distributed environment needs to address,
- support different types of queries and query matching techniques
- support the different research interests of scientists, and
- adapt to changes of scientists' interests over time.

5.3 Resource Discovery in Peer-to-Peer Environments

As with the WWW, resource discovery in P2P remains a challenging issue. There have been many search engines used in the WWW. However, because of the differences in architecture and requirements, available search methods used for the WWW cannot efficiently be applied in P2P environments. In a decentralised environment, a resource discovery approach in a distributed environment often involves two major operations: routing of search queries and matching the search queries against available resources. Depending on the approaches used, one operation could be more important than the other. The challenge is to design efficient methods of routing queries within a decentralised environment, while still able to support different requirements for matching of complex queries required for different types of resources. The current most popular resource discovery techniques can only partially meet the requirements.

The basic flooding technique used in Gnutella-like systems is routing based approach (Gnutella 2001, Kan 2001). The complexity in this approach is in the routing of search queries to relevant resource holders. Hence, it is able to support various kind of queries but not scalable as the population of peers of the network grows (Chawathe et al. 2003). On the other hand, matching based approaches are scalable in terms of query routing. In matching based approaches, matching of search queries and available resources are done first, usually by indexing, then the routing to location of resources. Example of this type is the indexing method using distributed hash table (Ratnasamy et al. 2001, Rowstron & Drusche 2001, Stoica et al. 2001). However, this type of approach can only support keyword matching. It cannot process complex queries, such as those which require matching the semantic of document contents.

Resource discovery methods which are based on the users' interests have recently emerged to improve the routing of complex queries (Iamnitchi & Foster 2005, Schlosser, Decker & Nejdl 2002, Schlosser, Sintek, Decker & Nejdl 2002, Sripanidkulchai et al. 2003). Instead of sending queries blindly to every peer in the network, these methods try to forward the queries to peers that are most likely to have the answers. The number of fruitless attempts can then be reduced. These approaches often require a complex network topology (Schlosser, Decker & Nejdl 2002, Schlosser, Sintek, Decker & Nejdl 2002) and/or clustering of peers into groups of common interest (Iamnitchi & Foster 2005). With clustering methods, a peer is assigned to only one particular group of common interest. Queries about other interests will not be efficiently routed.

The rest of this section reviews in detail these approaches to P2P resource discovery.

5.3.1 Centralised Indexing

The centralised indexing approach to P2P resource discovery was used in the Napster (Shirky 2001). This discovery model is matching based and similar to the search model used by search engines in the WWW. In this model, a centralised index server or a federation of centralised index servers keep a index of resources available on connected peers. When a peer issues a search query, the query will be sent to the index server(s) for processing. Results will be sent back to the issuing peer. As resources in a P2P network are dynamic, the index stored on the central server(s) has to be frequently updated. One of the disadvantages of this approach is that the network is not totally a P2P network. There is a certain level of centralised control. The index servers are the bottlenecks of the network. Therefore, if an index server is down, the whole or part of network will be down. Secondly, the indexing approach is not suitable for complex query matching.

5.3.2 Flooding Query

Flooding is a popular method for resource discovery in P2P environment. This method is well-known with Gnutella-like systems (Gnutella 2001, Kan 2001). The basic idea behind this technique is that, in order to locate a particular resource, a peer firstly asks all its neighbours about that particular resource. Then in turn, the neighbours ask their neighbours about that resource, and so on. Eventually, the request will be distributed throughout the network. If a peer finds the resource being asked in its local storage, it will send the answer to the original peer who made the request. In order to prevent the search request from wandering in the network, there is a Time-To-Live (TTL) associated with each query, which limits the number of hops a request can be forwarded to. The flooding technique is simple and easy to implement, however, it has some major limitations. Firstly, if every peer in the network keeps forwarding the query message, there will be a large number of duplicate messages as a peer may receive different copies of the same query. These messages are clearly redundant, and become a burden to the network. Secondly, the success rate of a search query is dependent on the value of the TTL. If the TTL is small, the success rate will be small. However, if the value of TTL is large, then there will be a growth, exponentially, in the network traffic. Determining a good TTL value is not an easy task. If the population of the network is large, then the TTL value must be large in order to have a good coverage to achieve good results. Consequently, the network traffic will grow considerably. This makes this pure flooding approach not scalable.

There have been a number of techniques used to address the problem of pure flooding algorithm. Kazaa P2P network (Kazaa 2006) is using a number of supernodes, which are peers having high bandwidth and processing power. These supernodes have index databases of shared content on other normal peers. The search queries are forwarded to supernodes, not to all peers in the network. Hence, the network traffic caused by passing queries is reduced.

Other approaches have also been introduced, but based on underlying physical network topology (Adamic et al. 2001, Chawathe et al. 2003). The basic principle of this approach is based on power-law distribution characteristics of a P2P network. In a network, there are nodes with higher degree (connectivity) and nodes with lower degree. The search queries should be forwarded to high connectivity nodes and then from high connectivity nodes to lower connectivity nodes in order to reduce network traffic. These approaches require a complex mechanism to recognise and select higher degree nodes.

Another method uses replication and random walker to improve success rate and reduce traffic in the network (Lv et al. 2002). With this method, the resources are replicated on multiple peers to enhance the success rate. A query, instead of broadcasting to all neighbours, is forwarded to a fix number of peers from the originating peer. It is then forwarded to other peers until the TTL is reached. This method reduces the number of redundant messages. However, it requires duplication of contents. The success rate depends on the content duplication rate and the TTL of search queries.

5.3.3 Indexing Using Distributed Hash Tables

Indexing approach using Distributed Hash Table (DHT) has been introduced as a scalable solution to resource discovery in distributed system, the problem that the flooding technique is suffering. This approach is popular with Chord (Stoica et al. 2001), CAN (Ratnasamy et al. 2001) and Pastry (Rowstron & Drusche 2001). The basic principle of DHT is based on the building of a structured network. Each peer in the network holds a partition of the whole network key space. A key lookup, hashed from a file name by a uniform hash function, is done by routing the key logically in the structured network to the peer whose key space contains the key being looked up. The looked up value, the location of the search file, is then returned to the requester. The cost of DHT method is composed of the cost of construction and maintenance of the DHTs and the cost of routing the key through the network (Rowstron & Drusche 2001). As the construction of the network is done only once and the maintenance happens only in part of the network where new peers enters or drop, this approach is scalable when the network population increases.

The main disadvantage of DHT approach is that it can only support the exact matching of search keys. Therefore, it is not suitable for application that requires complex query matching. There were attempts to modify DHT to deal with partial matching and multiple keywords (Felber et al. 2004, Schmidt & Parashar 2004), but they are still from unable to deal with complex query, for example, for those that require not only the matching of the file names but also the contents of inside the files.

5.3.4 Exploiting User Interests

A number of discovery methods that exploit user interests have also been introduced. The underlying principle of these approaches is to route search queries to peers that most likely have the answers to improve query hit rate and to reduce network traffic caused by unsuccessful queries.

The HyperCuP system used ontology to organise peers into groups of similar interests using a hypercube topology network (Schlosser, Decker & Nejdl 2002, Schlosser, Sintek, Decker & Nejdl 2002). With this the method used in HyperCuP, search queries are forwarded to interest groups to produce a better hit rate and reduce redundant query messages. This approach requires complex construction of the structured hypercube topology network. When joining the network, a peer declares its interest so that the network will place the peer into the cluster of its interest. Similarly, the METEOR-S system used ontology to classify peers in the network on their registration, so that search queries and publication messages could be routed directly to relevant peers (Verma et al. 2003). As P2P is a dynamic environment, a peer might change its interest over time. Constantly updating the network would result in high cost. Furthermore, it would be more complicated if peers had more than one interest.

Along this line, Cohen et al. (2003) proposed an algorithm to build associative overlays based on guide rules to route search queries. One guide rule proposed was possession rule, which grouped together peers that shared a common data item. This approach required a traced index of peers that participated on the rule. The use of document names for possession rules made it unable to deal with the semantic similarity of document contents. Alternatively, Sripanidkulchai et al. (2003) introduced an architecture in which a peer's view of the semantic overlay was a list of peers that had previously had answers to its queries. The future queries would be forwarded directly to peers in the cache list, as shortcuts. This method is simple to implement. However, the size of the list had a strong effect on the search results. If the users have many different interests, the hit rate may also decrease.

The small world pattern introduced by Newman (Newman 2001*a*, Newman 2001*b*, Newman 2001*c*) was used to develop an information dissemination algorithm (Iamnitchi & Foster 2005). The basic principle of this algorithm was to build clusters of peers of similar interests. The similarity of interests was calculated by the number of common file requests the peers had made. Each peer kept a list of indices of other peers who had downloaded its files. At a regular time interval, the peer exchanged its list with other peers in the network. The cluster of interest that the peer belonged to was calculated by the peer's list and lists it received from others. A search query (or a piece of information to be disseminated) was then targeted to its relevant cluster, where the query was most likely to be answered, to reduce network traffic and to increase query hit rate. There were some issues need to be resolved with this approach. Firstly, the formation of clusters was complicated and the size of the cluster should also be carefully calculated to achieve optimal performance. If the cluster size was very large, the quality of results would be decreased. However, if the cluster size was small, some relevant information would not be retrieved. Secondly, there was overhead caused by messages exchanged at regular interval for calculation of clusters. Thirdly, if a peer had more than one interest, queries of interests other than the common interest of cluster, to which the peer was assigned, would be less efficiently distributed.

A very similar approach to exploiting similarity in user interests has been introduced recently in TRIBLER P2P system (Pouwelse et al. 2006). A similar mechanism for identifying peers with similar interests in a social network is used in TRIBLER. Therefore, in terms of resource discovery, the TRIBLER has similar characteristics as the system introduced by Iamnitchi & Foster (2005).

5.3.5 Summary of Peer-to-Peer Resource Discovery Methods

Table 5.1 summarises the capability of different P2P resource discovery approaches in terms of scalability and ability to support different types of query matching. These are the two most important requirements for resources discovery in distributed scientific communities. Centralised indexing, flooding query and indexing using DHTs methods have not sufficiently addressed both of these issues. Centralised indexing and flooding methods are not scalable. The indexing techniques, which are more scalable, but not suitable for complex query matching. Only interest based methods reviewed can satisfy both of these requirements.

Methods\Requirements	**Scalability**	**Query supported**
Centralised indexing	No	Simple
Flooding queries	No	Complex
DHT indexing	Yes	Simple
Interest based	Yes	Complex

Table 5.1: Summary of capabilities of different P2P discovery methods in terms of scalability and supporting complex query matching

The current resource discovery methods based on users' interests tend to group peers having common interests into interest groups. However, with these models, a peer is assigned to only one interest group that it is most closely related to. All peers in a group either have the same set of interests or have their strongest interests similar. In the first case, the method will be unsatisfactory for users with more than one interest. In second case, only search queries for resources on the strongest interests are efficiently routed. Queries on other weaker interests are not. A good resource discovery method needs to efficiently route queries of any interests.

5.4 The Adaptive Approach to Peer-to-Peer Resource Discovery

This adaptive approach to P2P resource discovery is interest based. The goal is to provide an efficient mechanism for routing search queries in a pure P2P environment. As in other interest based approaches, this is a routing based method. The routing mechanism is separated from query matching so that it can be used with any types of queries and query matching techniques.

5.4.1 Terminologies

The following are definitions of terminologies used in subsequent sections of this chapter:

Query and Query message. A query message is a message that carries an actual query in P2P network. For example, a simple query could be a list of keywords to look for resources that contain the keywords. A complex query could be a SQL query that requires a database management system to process. In additional to the actual query to be matched against resources, a query message may also contain *routing information* that helps to route the query in a P2P network. For simplicity, a query message is generally referred to as query in the following discussions.

Query routing. Query routing is the process of sending/forwarding a query message from one peer to another peer to look for resources.

Query matching. Query matching is the process of comparing a query with resources to find a match.

Peer. A peer is a P2P application in a P2P network.

Peer's interests. A peer's interests refer to interests of a user who uses a P2P application.

5.4.2 The Principle

The basic principle that guides the routing of query in the adaptive method is that a peer should try to send a search query to peers that are most likely to have the answers. This will improve quality of results (e.g. number of relevant results found) and will also reduce unnecessary network traffic. A peer makes its decision on which ones are most likely to have the answers by learning from its past query results. This learning process happens continuously during a peer's life time, so that it can adapt to changes in its environment.

5.4.3 Underlying Properties

The guiding principle for this adaptive method exploits three properties emerging from the characteristics of scientific research communities. Property 1 provides a conceptual model for the grouping of peers. Property 2 is the basis to develop a learning mechanism and Property 3 underpins the routing algorithm for queries.

Property 1: There is the existence of groups of common interests within a research community.

In a large scientific community, collaborations usually take place amongst groups of scientists who are working on similar or the same topics. This is similar to the small world concept (Iamnitchi et al. 2004, Newman 2001*a*, Newman 2001*b*, Newman 2001*c*). However, a scientist may work on a number of related or overlapping research topics. Hence, he/she can participate in different groups.

The following 2 properties are drawn from Property 1.

Property 2: Scientists who have a common interest often need and share a common set of resources for that particular interest.

For example, in reaction kinetics, a chemist who is building a reaction model, such as burning of methane, may need to access available data related datasets and previously developed reaction models about methane. In turn, he will make his new models about methane available to others in need.

Property 3: Transitive relationships about 'interest in resources' exist in scientific research communities.

If two people (for instance, A and B) are interested in a particular type of resources (R) and one of them (B) knows that another third person (C) is also interested in that type of resources(R). Then, there will be an implicit thread of common interest between person A and person C on R. These two people are likely to have the need and to share resources on that topic. People connected by these transitive relationships eventually form a common interest group.

5.4.4 The Operations

The adaptive resource discovery method consists of three operations:

- *Describing peer interests using ontology*: The purpose of this operation is to create entries for a peer's diary (technically named as *Query History Tree*), which keeps information about other peers with similar interests. Each diary entry corresponds to an interest of the peer.

- *Recording peers with similar interests*: This operation helps to update a peer's diary about other peers with similar interests using query results. This operation is carried out whenever the peer receives query results.

- *Routing a search query*: This operation is carried out whenever a peer needs to forward a search query. The peer uses its knowledge stored in its personal diary to intelligently route the search query.

These operations exploit Property 2 and 3 of scientific research communities to implicitly realise the natural grouping of scientists as described in Property 1. These operations are discussed in detail in following subsections.

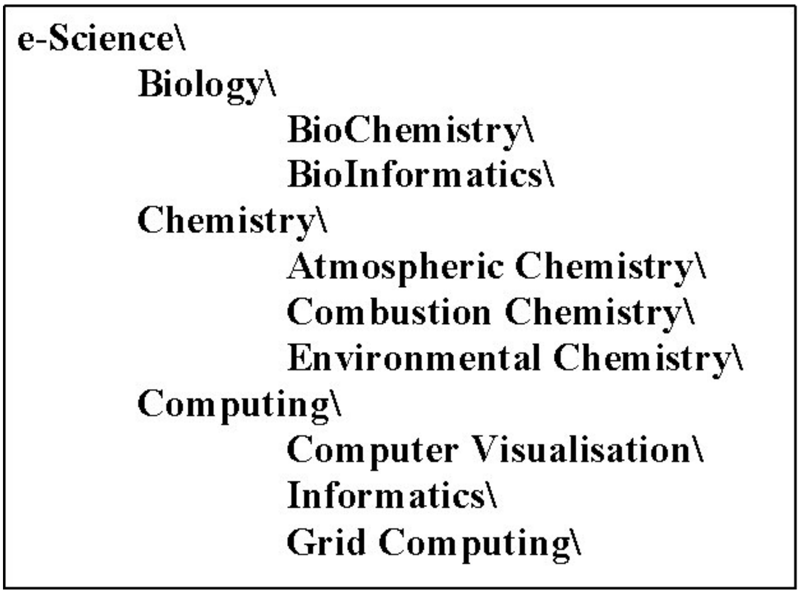
5.4.4.1 Describing Peer Interests

Initially, each peer in the network is provided with an initial classification ontology called *global ontology*. This global ontology adopts a hierarchical structure and defines a set of terms that are globally recognised for classification of a wide range of interests in the target user community. It is similar to eBay or Yahoo directory but the interests are from a scientific domain. Using this global ontology, a user can describe his/her peer interests.

As an example, Figure 5.1 shows a fraction of the global ontology that might be used to describe the e-Science domain.

The ontology in Figure 5.1 starts with 'e-Science' as the general domain. In the 'e-Science' domain, there are three sub domains: 'Biology', 'Chemistry' and 'Computing'. Similarly, the 'Biology' domain can further be classified as 'Biochemistry' and 'Bioinformatics' and so on. The ontology provided in Figure 5.1 is only a very simple ontology for illustration purpose. In reality, the initial ontology should contain much more detail.

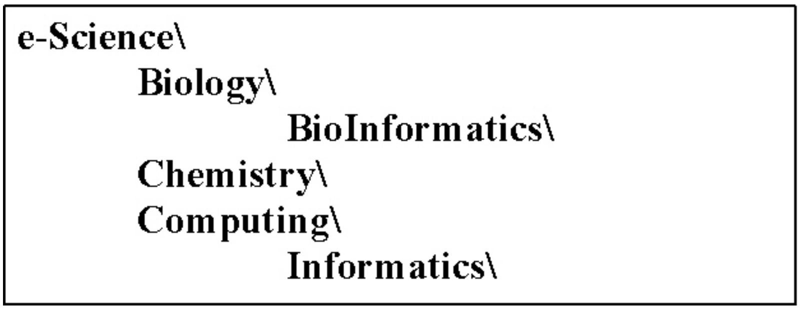
Using the global ontology provided, individual scientists start to describe their interests. If a scientist has only a general interest in 'Biology', the classification can just be 'e-Science\Biology'. However, if the scientist has more specific interests within 'Biology', for instance 'BioInformatics', the classification associated with the peer should be



```
e-Science\
        Biology\
                BioChemistry\
                BioInformatics\
        Chemistry\
                Atmospheric Chemistry\
                Combustion Chemistry\
                Environmental Chemistry\
        Computing\
                Computer Visualisation\
                Informatics\
                Grid Computing\
```

Figure 5.1: A fraction of an initial global ontology for e-Science community

‘e-Science\Biology\BioInformatics’. ‘Informatics’ within ‘Computing’ might also be of interest. Hence, ‘e-Science\Computing\Informatics’ is added to the peer’s description of interest. The final description of the peer interest for our example will be a subset of the initial ontology, as illustrated in Figure 5.2.



```
e-Science\
        Biology\
                BioInformatics\
        Chemistry\
        Computing\
                Informatics\
```

Figure 5.2: Description of a peer’s interests

5.4.4.2 Recording Peers with Similar Interests

This is a learning operation which takes place through out the life time of a peer. The learning process helps the peer to update its knowledge and to adapt itself to the environment. The learning process is based on Property 2 described above.

For each peer, a *Query History Tree* (QHT) is constructed. The backbone of the QHT is the classification ontology used to describe the peer's interests. Each node (branch or leaf) of the QHT represents a classification of a peer's interest. Attached to each node of the tree is a *peer list* which records the peers that have previously answered queries on the interest represented by the node. Each entry to a 'peer list' must contain enough information to identify a peer in the network. Depending on the P2P application used, an entry could be a peer ID or a pair of IP address and port number. Initially, these lists are empty and will be updated during the life time of the peer. As a node in the QHT is a sub-classification its parent node, a 'peer list' of a node may also inherit information contained in the 'peer list' of its parent node. Figure 5.3 gives an example of the QHT for the peer used in Figure 5.2.

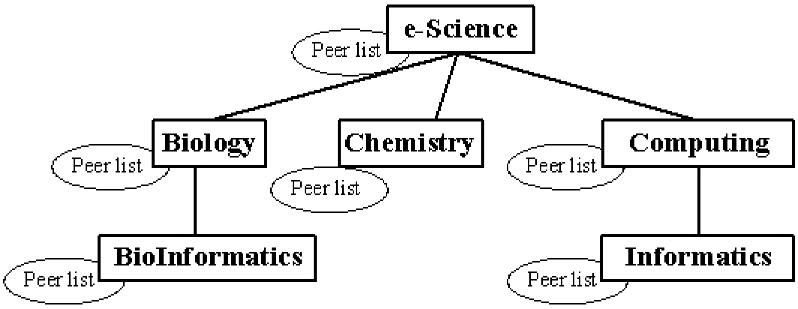


Figure 5.3: A query history tree of a peer

When a query is issued by a peer, a *classification tag*, which defines interest area that the query is looking for, will be attached to the query. This classification tag is constructed as a tree path from the root of the peer's QHT to the node that represents the interest. Following the previous example, if a query is looking for resources about 'BioInformatics', a classification tag 'e-Science\Biology\BioInformatics' will be assigned to it. When receiving query results, the peers with valid responses will be added into the peer list attached to the node.

The decision on whether a response is valid or not depends on the implementation strategy. If full automation is chosen, any peer answering will be added to the list. However, this method might not produce a very accurate list, as it commonly happens that a response to a search query is not necessarily a relevant answer to the query. If accuracy is preferred, the validation will be left to the user. With this approach, only peers with valid answers will be added to the peer list of the node. For complex queries, the second approach is preferable, in order to improve the quality of the peer lists.

A peer list attached to each node of a QHT can also be a priority list. The use of prioritisation strategy decides how a peer should adapt to changes in the environment.

Depending on the characteristics of the community, priority can be given to peers that have previously provided the largest number of valid answers or the most recent valid answer. If 'most recent valid answer' is prioritised, the peer will respond quicker to changes in the environment. Hence, it is more appropriate for a fluid or newly set up group. However, in a more static environment, the use of 'largest number of valid answers' will provide more reliable query results.

5.4.4.3 Routing of Queries

The routing mechanism aims to utilise the existence of common interest groups for more effective routing. By using QHTs constructed in the previous step together with the transitive relationship of 'interest in resources' amongst the peers (Property 3), a 'network' of peers with a common interest can be traced (Property 1). This overlay network can be used for routing the query to the peers most likely to provide an answer. The following will explain the routing in detail.

Each search query is associated with a *Time-To-Live* (TTL), a *fan-out* value (f) and a classification tag:

- *TTL*: the maximum number of hops that the query can travel within the network. This value is defined by the application.
- *Fan-out value (f)*: the number of peers to which a peer will forward a query message. This value is defined by the application.
- *Classification tag*: the construction of classification tag is explained in the previous subsection. It is used for routing the query and for recording query results.

The routing process is carried out when a peer issues a search query or when it receives a search query from another peer. The information about peers with similar interests contained in the QHT of the peer is used to guide the routing of query messages to next appropriate peers within the environment.

When issuing a query message at a peer:

- The user specifies the topic that the query message is looking for (e.g. 'BioInformatics').
- A classification tag will be constructed and attached to the query (e.g. 'e-Science\Biology\Bioinformatics').

- The peer will then look up at the node of its QHT that is pointed to by the classification tag and pick up from the peer list of the node first f peers to forward the query to.

On receiving a search query, a peer will:

- Attempt to answer the query by searching in its local storage for relevant resources.
- If there is an answer, reply directly to the requesting peer.
- If the query has not reached its TTL, then use the classification tag attached to the query to look up in its QHT for the first f peers (similar to the previous case), and forward the query to the selected peers.

A few possibilities might happen when routing search queries:

(i) If the peer list being pointed to by the classification tag is empty (for example, in the initial state, when peer lists in the tree have not been populated) or the number of peers in the peer list is fewer than f, then the following steps will be done:

  - Traverse up the tree and pick up the peers in peer lists held by parent nodes until the requested number is met, with the priority given to closest parents.
  - If the request is still not met, forward the queries to the peers selected by the previous step and some randomly selected neighbouring peers to have enough f peers or as closest as possible.

(ii) If the classification tag carried by the query does not match any node of the QHT of the processing peer (which is processing the query), then partial mapping between the classification tag and the peers QHT will be used. This situation may happen when the current processing peer and the peer that originally issues the query have different interests or different description of interests. The partial mapping will start from the root of the tree and the root of the classification tag. Only the matching part of the classification tag with the QHT will be used by the processing peer as if it was the classification tag. The procedure used to select peers to forward the query to will be exactly the same as in the previous case. If no match is found, the query will be forwarded to random neighbours.

(iii) To avoid forwarding a query to a peer more than once, a loop detection technique is used. This technique requires each query to keep a record of peers it has visited. Before forwarding a query to other peers, a peer will check the path that the query has taken so far, and will only forward the query to peers not in the record. However, with the current routing mechanism, at every hop on its way, a query message is cloned into f copies before being forwarded to the next set of peers. It is possible for the ‘same’ query to arrive at a peer via more than one route taken by different ‘cloned queries’. This kind of duplication cannot be eliminated by the loop detection technique.

5.5 Experiments

5.5.1 Objectives

In order to evaluate the efficiency and also to analyse the behaviours of the adaptive approach, three experiments were conducted with the following objectives:

- The first experiment was to evaluate the efficiency of the adaptive resource discovery method by comparing its performance with the basic blind (random) flooding method.
- The second experiment was to analyse the relationship between resource distribution of the network and the efficiency of the adaptive method.
- The third experiment was to analyse the sensitivity of the adaptive method in response to the increases of network population. This sensitivity analysis will also be used an indication of the scalability of the adaptive method.

All the experiments were conducted in a simulated condition. This was the only feasible way to have a controlled experiment on a P2P environment with potentially thousands of peers involved with hundreds of thousand of queries messages.

5.5.2 The Simulation Engine

The simulation engine was built entirely in Java. There were two simulation programs - one for simulations on the adaptive method and the other for flooding methods. Each peer was represented by a Java object in these programs. In each simulation, connections amongst peer objects formed a simulated P2P network. A message passed from one peer to another was simulated by a method call between the two Java objects. Details of the simulation engine are discussed in the following subsections:

5.5.2.1 Network Peers

There were two classes of peers for the two simulation programs. The SmartPeer is for simulations on the adaptive method. The BlindPeer is for the flooding method. Basic properties of these two classes of peers are the same: identification, number of connected neighbours and number of resources available. The difference between these two classes was the way their instances handled search queries and search results. An instance of the BlindPeer class randomly forwarded queries to its neighbours, whereas, an instance of the SmartPeer class needed to keep a record of its previous search results in a QHT and used this knowledge to forward search queries as specified by the adaptive approach.

The QHT of SmartPeer was implemented using Java hashtable. Each entry of the hashtable represented an interest of a peer. Associated with each entry (interest) was a Java vector (array) that held a record of peers which potentially had answers to queries about the interest. For simplicity of implementation, the QHT implemented in this simulation engine had only one level in depth. The number of interests a peer could have was controlled by an input parameter.

5.5.2.2 Network Topology

The procedure and parameters used to create the P2P network topology were the same for simulations on both discovery methods. At the creation of the network for each simulation, the peers were generated by instantiating SmartPeer or BlindPeer classes. Each peer was assigned with an identification, which was the order of its creation. Each peer was also randomly assigned with a number of its connected neighbours, which were kept in a one dimension array. The number of neighbours that a peer could have was also randomly generated using input parameters. In all simulations conducted, the maximum number of neighbours a peer could have was 7 and the minimum was 3. After their creation, all peers were kept in a one dimensional array. Index of a peer in the array was its identification number for easier lookup.

5.5.2.3 Resources and Peer Interests

Resources used in the simulations were enumerated as discrete positive integers for simplification of query matching. As this adaptive method focuses only on the routing of search queries, it would be sufficient to use enumerated integers for comparison without the loss of generality.

The whole resource domain was organised into a number of smaller categories, based on their values. Each category represented an area of interest, or a classification of resources. For example, if the resources were in range from 1 to 100 and organised into 10 categories, the resources in range 1 to 10 would be grouped the first category. Resources whose values were from 11 to 20 would be grouped to the second category and so on. Each of these resource categories could be thought of as an interest area in reality, such as computing or biology.

On initialisation for each run of simulation, each peer was assigned with a number of resource categories as its interests. A number of resources, with values ranged within the assigned categories, were randomly generated for each peer.

The total numbers of resources, their range of values, the number of categories, numbers of resource categories assigned for each peer and the number of resources per peer could be configured for each simulation through input parameters. In most cases, the resource range was set from 0 to 4999. The number of resources assigned to each peer was 5. Other parameters could vary, depending on the objectives of the specific experiments.

5.5.2.4 Query and Query Forwarding

As each piece of resources was represented as an integer, a query would be in form of finding peers that held a particular integer. In the specification of the adaptive method, each query also needed to be tagged with a classification of resources. However, in the simulation, as a query key was an integer, the category that the integer belonged to could be easily inferred from its value. Therefore, no classification tag was actually used.

Each search query was also associated with a fan-out factor and TTL value. In this simulation, they were fixed at 3 and 6 respectively for all simulations .

A peer forwarded a query to another peer in form of a procedure call. A calling object (a peer) invoked a procedure of a target object (another peer). The parameter used for this procedure call was the query. The matching between a query and a peer's resources was basically comparisons between the integer of the query and five integers, which represents 5 different resources of the peer.

In each simulation, peers were selected randomly to issue search query until the maximum number of queries required for the simulation was reached. The integer values of queries issued by a peer were also randomly generated in range of the peer's assigned categories, to reflect its interests. As the number of queries generated for each simulation was quite large (in order of hundreds thousands to millions) in comparison with the network population (in order of tens of thousands), it was expected that every peer would generate a similar number of queries.

5.5.2.5 Configuration Parameters and Logging

Input parameters for configuration of simulations were kept in a separate file, which was also written in Java. These parameters could be adjusted and needed to be recompiled for every run.

Parameters used for the experiments were chosen in a way that the effect of the adaptive method could be clearly identified. The span of a search query, which was the number peers that a search query would be forwarded to, and the density of resources needed to be relatively small in comparison with the whole network population. Otherwise, there would be no need for an intelligent routing. The selections were made after a number of test runs on the simulation engine with various parameters. For a particular experiment, these parameters could be changed for different analysis purposes. However, the network topology, fan-out and TTL of search queries, resource range and resource density were the same for all experiments, as these were usually static properties of a P2P network.

Output of each simulation was basically a log file, in text format, which recorded the network configuration for the simulation, number of hits, hit rates (in percentage) and number of query messages passed around the simulated network for each query interval.

5.5.3 Experiment 1 - Evaluating the Adaptive Approach

Two simulations were set up - one for the blind flooding method and the other for the adaptive method. The same network configuration and pattern of resource distribution were used in both simulations.

Network configuration: The simulated network was set up with 10,000 peers. The network topology was randomly generated so that every peer would be connected to at least 3 and maximum 6 neighbours.

Resource distribution: Each piece of resource was randomly enumerated as an integer in range 0 to 4,999 (inclusive) and was assigned to one of 500 categories, based on its value. Each of these categories represents a topic of interest. Four consecutive categories were assigned to each peer to represent the interests of the associated scientist. In the simulation of the adaptive method, the categories (areas of interests) form the ontology.

Each peer was assigned with randomly five pieces of resources, ranged within its assigned categories.

Measurement: The following two measurements were taken in each of the simulations:

- *Query hit rate*: calculated by the ratio of 'the number of queries that have answers' and 'the total number of queries issued' by all peers in a specified period. Hit rates were represented in percentages.

- *Network traffic*: measured by 'the total number of query messages' passed around in the network during a specified period.

The period used to calculate query hit rates was defined in terms of number of queries issued. Particularly in this experiment, a period was set as a total of 5,000 queries being issued by all peers.

Process: A total of 400,000 queries were generated by all peers in the network for each of the simulations. Queries produced by a peer were restricted within its assigned categories. After every 5000 queries, a hit rate was calculated and recorded. Network traffic after every 5000 queries was also recorded. In this experiment, for both methods, when a peer found an answer for a query in its local storage, it would stop forwarding the query.

Results and analysis: The graph in Figure 5.4 shows the hit rate comparison between the blind flooding method and the proposed adaptive method. As seen from the graph, the hit rate of the blind flooding method, calculated after each 5000 queries, fluctuated below 30 percent, while, the hit rate of the adaptive method grew gradually when the number of queries increased. After about 325,000 queries, the hit rate of the adaptive method reached 90 percent. It became stable at 93 percent, after 360,000 queries were issued.

The hit rate of the adaptive method improved dramatically when the number of queries increased. This is because it took into account the characteristics of resource distribution within the environment. At the beginning, the hit rate of this method was roughly the same as the blind flooding method. However, when the learning progressed, peers accumulated more knowledge about the environment. Therefore, search queries were forwarded to more appropriate destinations. The hit rate levelled when it had learned quite enough about its environment. In the blind flooding method, search queries were always forwarded randomly to other peers, hence the hit rate was almost the same, no matter how many queries had been issued.

Similarly, the graph on Figure 5.5 shows the number of messages passed in the network after every 5000 queries for both cases. As expected, the number of messages needed for every 5000 queries by the blind flooding remained roughly the same (just over 1,800,000). In the case of using adaptive method, the number of messages required for every 5000 queries decreased when the total number of queries increased. This is easy to explain. As the hit rate increases, fewer number of query messages would be passed on from peer to peer.

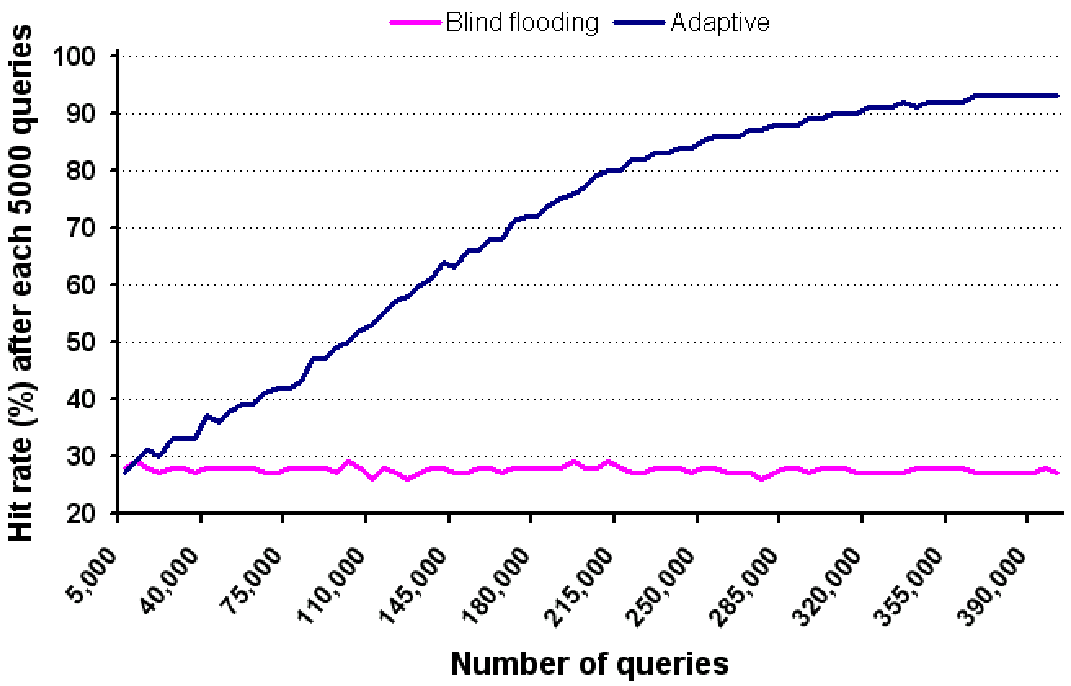


Figure 5.4: Hit rate comparison between the blind flooding method and the adaptive method

In conclusion, this experiment has shown that the adaptive method is more efficient than the blind flooding method. After a certain number of queries are issued, the learning process will help peers to adapt to their environment. As a result, the query hit rate will increase.

5.5.4 Experiment 2 - Effect of Resource Distribution

In order to analyse the effect of resource distribution on the proposed adaptive approach, a number of simulations using this method were run using different patterns of distribution.

Network configuration: This experiment used the same network configuration as in the previous experiment.

Resource distribution: Resources and categories were enumerated in the same way as in the previous experiment. The only difference was the number of resource categories assigned to individual peers. In this experiment, several runs of the adaptive method were performed. In each run, peers were assigned with a different number of resource categories (2, 6 and 10).

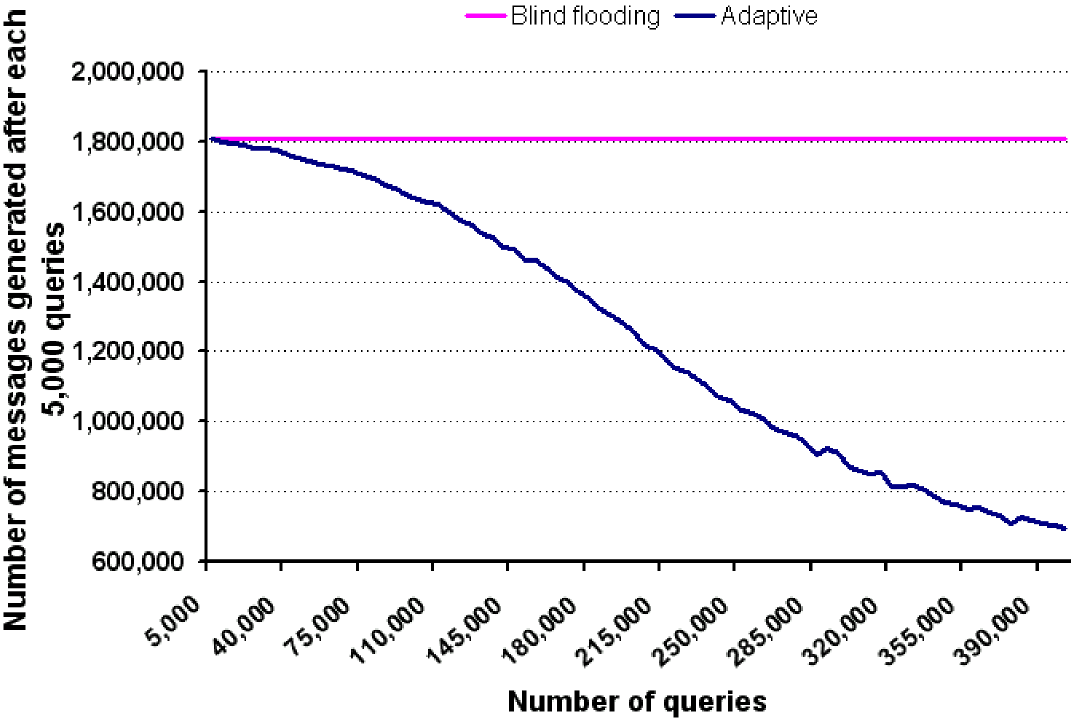


Figure 5.5: Messages passed in the network when using the flooding method and the adaptive method

Random distribution of resources was also experimented. Simulations were run on the following two implementations of random distribution using the adaptive method:

- The whole resource domain was classified into 500 categories as the previous simulations. In this simulation, a peer could have any resource within these 500 categories.
- Resources in the network were treated as in one category. All peers could have any number of resources within resource range of 0 to 4,999.

Measurement: For each simulation, after every 5000 queries were generated by all peers in the network, a hit rate was calculated for comparison.

Process: The measurement process for each simulation was done exactly in the same way as in the previous experiment.

Results and analysis: The graph in Figure 5.6 shows different hit rates returned by simulations using different patterns of resource distribution.

As shown on the graph, as the number of categories assigned to each peer increased, it took longer for the hit rate to rise. This result concurred that the learning outcome

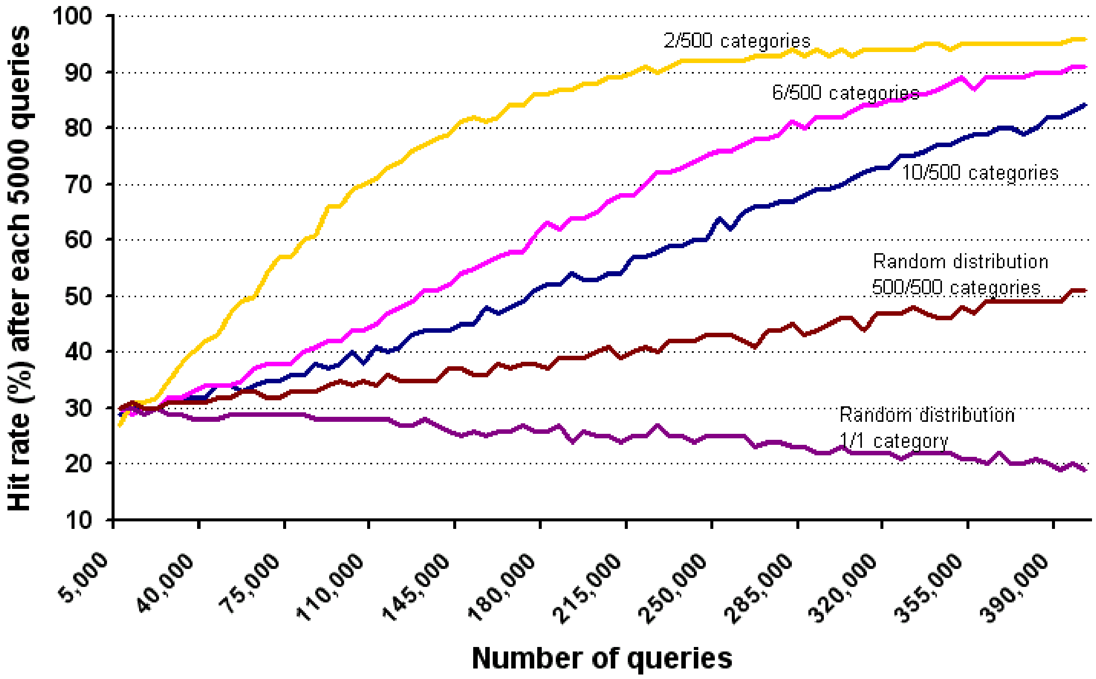


Figure 5.6: Query hit rates of simulations on different resource distribution configurations

would be better when each peer had resources in fewer categories. When each peer has limited amount of resources (as in this experiment), and if the resources on each peer were spread over so many of categories (interests), it would be harder to learn accurately a peers interests. Therefore the learning outcome would be less accurate. It would take longer for the network to produce optimal query results.

The two ways of applying the adaptive approach to a random distribution of network resources produced two contradicting results. By classifying all resources to one category, the query hit rate kept decreasing when the number of queries increased. Whereas in the other case, the hit rate produced increased over time, despite slowly. This is because when treating the whole resource domain as one category, the use of ‘peer-list’ for learning not only did not help, but also encouraged a ‘group-think’ scenario. This means a smaller group of peers seemed to satisfy each other’s query, hence having very little opportunity to explore peers outside the group. As a result, the coverage of query messages (for a query) was reduced. The scope was even smaller than the coverage of a query routed by the blind flooding method. Hence, the hit rate was lower.

In summary, this experiment had two important outcomes. Firstly, it has shown that resource distribution of the network has an effect on the performance of the adaptive method. Secondly, the adaptive method can also be used in a random distribution network if resources are categorised.

5.5.5 Experiment 3 - Sensitivity and Scalability in Response to Network Population

In this experiment, there were a number of simulations on networks with different populations to analyse the sensitivity of the adaptive methods. In order to analyse the trend of query hit rates on these networks in long run, large numbers of queries were issued in these experiments.

Network configuration: The simulated networks were set up with different populations. These were 10,000, 20,000 and 30,000 peers. Network topologies were randomly generated in the same way for all the simulations as in the first experiment, where every peer would be connected to at least 3 and maximum 6 neighbours.

Resource Distributed: Resources in all of the simulations in this experiment were generated in the same way as in Experiment 1.

Measurement: Query hit rates were measured for analyses as in previous experiments. It was ratio of the sum of queries that had answers and the total number of queries issued (by all peers) in a period. However, in this experiment, two different sets of hit rates were calculated using two different periods.

- A hit rate was calculated after 5,000 queries were issued by all peers in the network as in the previous experiments
- A hit rate was calculated after every peer in the network, on average, had issued a search query

The first measurement period was used to analyse the sensitivity from an overall network perspective. The second measurement period was for the analysis of the adaptive method from a peer's point of view. Two different simulation processes were used for calculating two set of hit rates.

5.5.5.1 Sensitivity from an overall View

Process: Three simulations were run on these three simulated networks. A total of 2 million search queries were issued in each simulation. In each simulation, after each period of 5000 queries had been issued, a hit rate was calculated. The total number of messages generated in the networks for each of these periods was also calculated to analyse the load of the networks.

Results and Analysis:

The hit rates computed from the three simulations using the period of 5,000 queries were summarised in Figure 5.7. The figure shows that, from the overall network point of view, the hit rates of the three networks with different populations increased quickly after the simulations started and levelled at an optimal condition, approximately above 95 percents. This general trend of the adaptive method had been confirmed by the outcomes of the previous experiments. The figure also shows that hit rates of networks with smaller populations increased faster than those with larger populations from this perspective. However, the networks with larger populations eventually returned higher query hit rates.

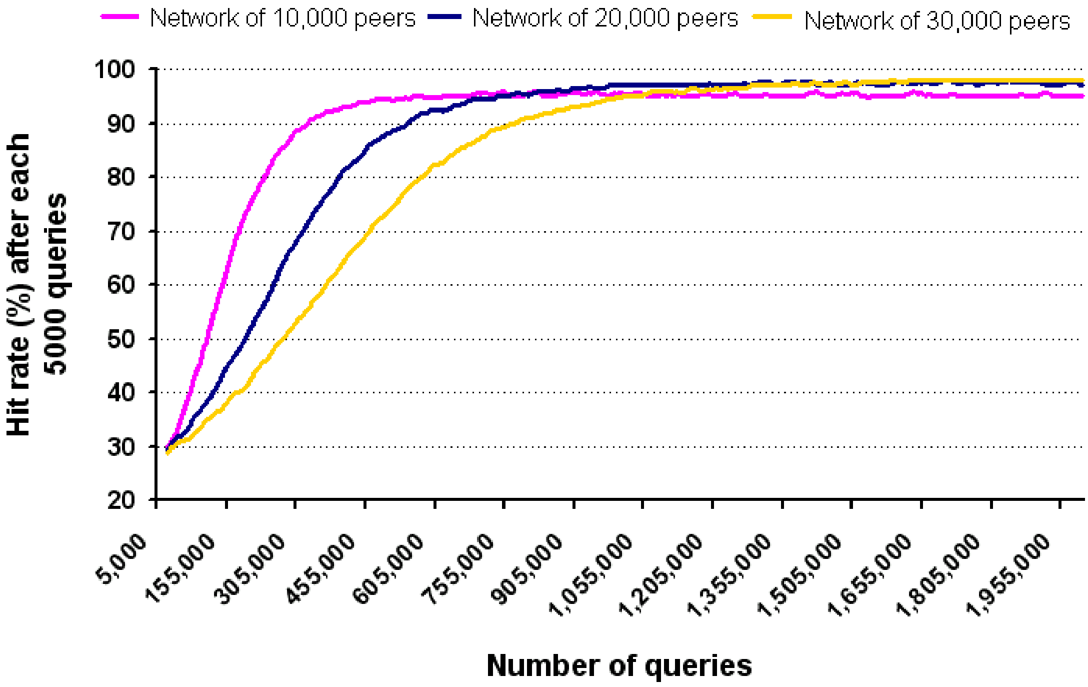


Figure 5.7: Hit rate comparisons amongst three networks with populations of 10,000, 20,000 and 30,000 peers from overall perspective. Hit rates were calculated after each 5,000 queries were issued by all peers.

The difference in learning speed was because in the larger networks, the chance that a peer could send a query message, out of the total of 5,000 queries, was smaller than in smaller networks. This reduced the chance of a peer, and collectively all peers, to learn initially. However, as the number of copies of a piece of available resource in larger networks was bigger than that number in the smaller networks, with the same resource density and all the resources were in the same range, therefore, in the long term, when the knowledge of individual peers reached an optimal level, the networks with larger populations returned higher hit rates. This implies that in larger networks, the adaptive methods can be more efficient.

In a reverse direction, the total number of messages generated within the networks with smaller populations decreased at a higher rate in the initial period, immediately after the simulations started. This is shown in Figure 5.8. However, when the number of queries issued by all peers in the network increased to a certain level, the number of messages generated by networks with larger populations became lower. As can be seen in the graph of Figure 5.8, when the total number of queries issued reached 1,500,000, the number of messages generated by the network with 30,000 peers became the lowest.

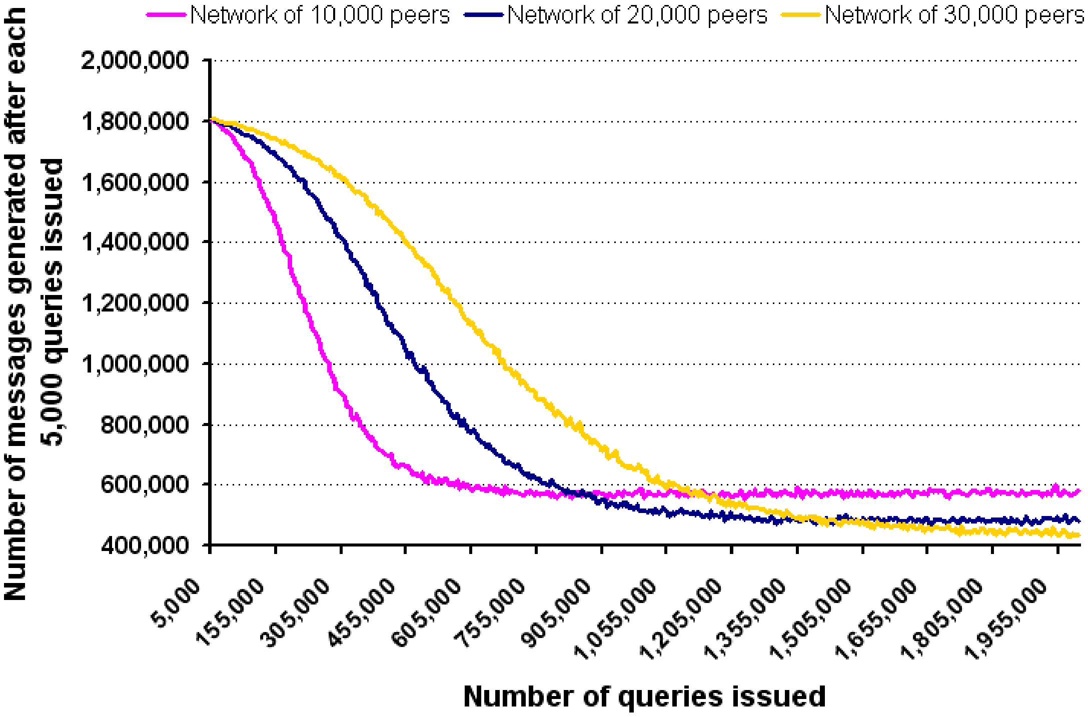


Figure 5.8: Message passing comparisons amongst three networks with populations of 10,000, 20,000 and 30,000 peers from overall perspective. Numbers of messages passed in a network were calculated after each 5,000 queries were issued (by all peers).

5.5.5.2 Sensitivity from a Peer's Point of View

Process: There was also one simulation on each of the networks. In each of the simulations, each peer sent totally 100 search queries on average. As peers were randomly selected to issue search queries, this average was deduced from the total number of queries sent out by all peers in the network. For example, if the network consisted of 10,000 peers, in order to get 100 queries each peer issued on average, the total number of queries issued by all peers would be 1,000,000 queries. As the purpose of this set of simulations was to analyse the hit rate from an individual's peer view point, the number of messages passed in the systems from this view point was not relevant.

Results and analysis:

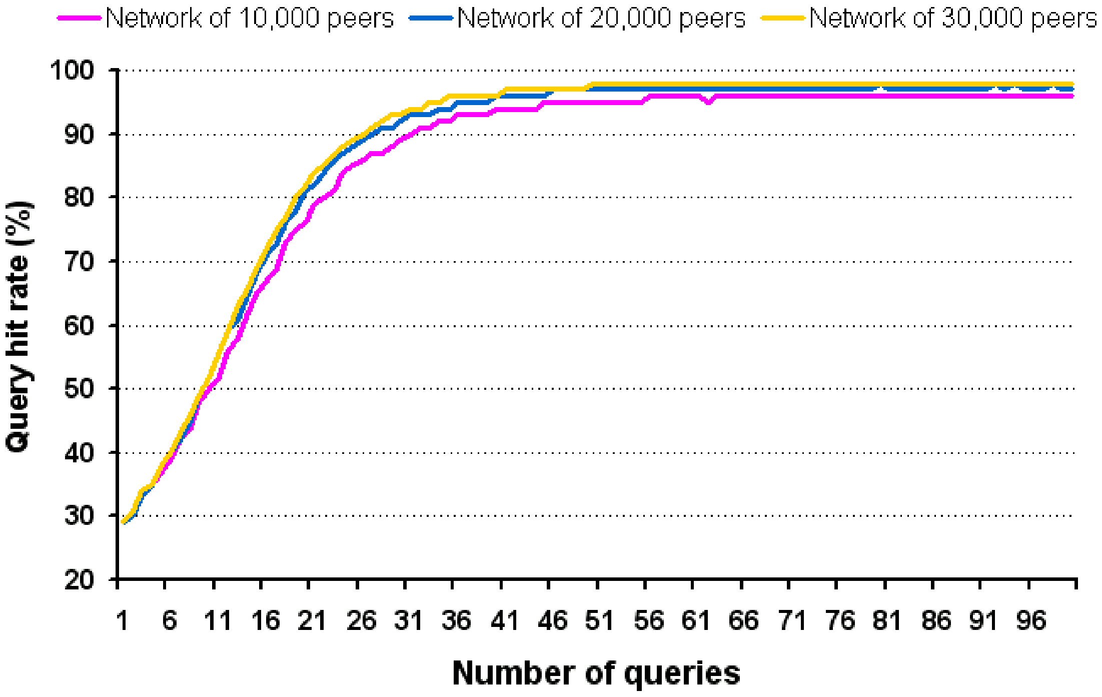


Figure 5.9: Hit rate comparisons amongst three networks with populations of 10,000, 20,000 and 30,000 peers from individual perspective. Hit rates were calculated in periods that each peer, on average, had sent a query.

The hit rate comparison from an individual's perspective is demonstrated in Figure 5.9. The graph shows that hit rates of networks with larger populations increased slightly faster and were always higher than networks with smaller populations. In these three simulations, every peer in all three networks issued the same number of queries. Hence peers in the three networks had the same chance of learning. The difference in learning speed was due to the effect of collective learning feature of the adaptive method. When issuing a search query, a peer initially used its knowledge to forward the query to potentially best destinations. In turn, on receiving the query, other peers also helped to forward it to next destinations. Therefore, not only did a peer use its knowledge to route its search queries, but other peers also contributed their knowledge to the routing.

The results shown in Figure 5.7 and Figure 5.8 confirmed the scalability of the adaptive discovery method. When network population was increased, the search did not only return higher hit rate, but it also reduced the number of messages generated by a search query. It clearly scaled well with the increases of the network population. This was due to the effect of continuously learning of individual peers.

In a summary, this experiment has confirmed two important characteristics of the adaptive discovery method: (i) it can return a higher hit rate when the network is richer in resources and (ii) it is scalable in networks with large populations. This is a result of the efficiency of the adaptive and collective learning features of the method.

5.6 Issues about Management of Classification Ontology

The experiments reported have shown the efficiency and scalability of the adaptive approach. This is a result of the collective learning process, in which the classification ontology for describing users' interests is the heart. Using ontology in a P2P environment, however, poses a new challenge. As ontologies are commonly agreed shared concepts (De Roure et al. 2005, Aberer et al. 2003), they require certain level of centralised management and time to maintain its consistency. Consequently, pre-agreed ontologies are often not sufficient in ad-hoc and dynamic situations, especially in P2P environments where new contents being produced regularly (Aberer et al. 2003, Mathes 2004). However, if the management of ontologies are pushed down to individual users and local user communities, semantic interoperability of ontologies at a global scale will be difficult to reach.

The query routing mechanism of the adaptive method uses a technique to temporarily resolve conflict between two different concepts generated locally by using parent concepts up in the hierarchical structure of the classification ontology. This technique can only temporarily resolve the conflict to get a search query going to its next destinations. It however reduces the accuracy of the routing. In the long run, the conflicts need to be resolved at local ontology level to improve the consistency.

An outline of a possible approach to address the above challenges is provided in Section 6.3.1 of the next chapter.

5.7 Summary

The chapter has proposed an adaptive approach to resource discovery in a P2P environment. This adaptive approach takes into account the characteristics of scientific research communities in order to provide an efficient way of routing search queries. As the routing is separated from query matching, this adaptive approach can be used with any types of queries and query matching techniques.

In comparison with other interest-based approaches have been discussed, the proposed adaptive method provides users with more flexibility. Although also exploiting similarity of user interests, it does not require complex construction of the network. Only general classification ontology for the domain is required at the beginning. It does also not limit peer users to any particular cluster. The users can participate (implicitly) in any group by declaring their interests. The algorithm will adaptively locate the group. In case there are groups with too big or small size (that might affect quality of search results) the users can use the query history tree to further classify the big groups or to merge the small groups to a larger one.

The experiments showed that this approach can significantly improve query hit rate in comparison with blind flooding method. If individual peers have many different interests on average, the learning speed will be slower, though, an optimal speed will be reached in the end when the number of queries issued gets to certain level. Most importantly, experiments have confirmed that the adaptive approach is scalable in large population networks.

In these experiments, as discrete resources were used in simulations, the number of discrete resources in a given range (e.g. a category) is limited. Therefore, the hit rates returned by the adaptive approach were very high (over 90 percents) when it got into an optimal condition. This was because peers in the network could accurately and exhaustively learn the number of discrete resources available. In a realistic environment, as the number of resource in a give range is unlimited, query hit rate returned by a search query may be lower.

Chapter 6

Conclusions

This chapter concludes the whole thesis. It starts by summarising research findings that have been reported. The chapter continues with the contributions of this work. The discussions on potential future work follow at the end. Two potential areas for future work to discuss are the management of classification of ontology in the P2P collaborative environment of the CeSA and a revision on the CeSA based on results of the user evaluation and introduction of the adaptive resource discovery and discussion on potential further work.

6.1 Research Findings

The problem this research addresses is finding an efficient collaborative architecture to support end-to-end distributed scientific collaborations. An integration of a P2P collaborative environment and Grid computing has been identified as a candidate solution. In this research, an investigation into an integration of these two computing models using Web services and a usability of the new architecture have been done. The following is a summary of research findings:

- The kind of scientific collaborations that needs to be supported includes the sharing of large scale computational resources, huge volume of datasets as well as day-to-day collaborative activities for gathering knowledge and expertise across disciplines such as sharing a piece of working data, passing information about available resources or giving advice on a particular research problem. The collaborations happen not only within the boundary of a particular institution but at a global scale. Protection of personal resources during the collaboration process in order to help scientists maintain their competitive edge is important to encourage them to collaborate.

- An integration between a P2P collaborative environment and Grid computing, using Web services has been formulated to support the kind of scientific collaborations required. The P2P environment is designed to support lightweight day-to-day collaborative activities such as share a piece of research data, whereas the Grid infrastructure is dedicated to sharing large-scale computational resources and large datasets. Access from the P2P environment to Grid computing resources is made possible via Web services

- The result of a user evaluation on usability of the proposed architecture has confirmed the potential of P2P collaborative environments in supporting day-to-day collaborations amongst distributed scientists. Resource sharing function of the P2P application was identified as the most useful feature of the P2P environment. The ability to provide access to Grid resources using Web services for scientific simulations, those require heavy computational capability was also recognised as an useful feature of the architecture.

- An adaptive resource discovery for the P2P collaborative environment of the proposed collaborative architecture has been developed, to efficiently support the sharing of resources. Simulation results have shown that this adaptive approach can greatly improve the query hit rate in comparison with the basic flooding method. This adaptive method can also flexibly adapt to changes in users' interests to efficiently route search query in order to produce better query results.

6.2 Contributions of This Work

To conclude, original contributions of this work are summarised below:

- The first major contribution is the introduction of the CeSA for collaborations within distributed scientific communities. It is the idea of having the integration of a P2P environment and Grids to support scientific collaborations at two different levels of granularity. The results of the early user experiment have initially confirmed the approach and motivated further research on the integration.

- The second major contribution is the adaptive resource discovery approach for decentralised P2P environments using classification ontology. This approach was designed for the CeSA based on the context and characteristics of scientific communities. However, it can also be applicable to other types of communities. Experimental results have shown that the approach significantly improves query hit rates in comparison with the random flooding method. The results have also shown that this discovery approach is scalable with respect to the network population. Together with the classification, this approach has a lot of other potential in supporting decentralised communities, in addition to resource discovery, such as location of expertise for possible collaborations and formation of groups of common interests.

In addition to the two major original contributions above, the followings can also be considered as contributions:

- The study of a typical e-Science community through the case study of the Reaction Kinetic research community. This community is typical because it is both data and computation oriented and requires tight collaborations amongst related disciplines.
- The prototype version of the CeSA, which consists of a P2P collaborative application and a number of GT3 services for simulations and analysis in Reaction Kinetics.
- The method for simulations of resource discovery in decentralised P2P environments.

6.3 Future Work

This future work section discusses on two potential areas - the management of ontology in the P2P collaborative environment of the CeSA and the work on its revision.

6.3.1 Evolutionary Approach to Classification Ontology

This is a proposed method for the management of classification ontology for the adaptive discovery approach used in the P2P collaborative environment of the CeSA. It is based on a similar principle as emergent semantic principle proposed by Aberer et al. (2003). The users of an ontology will eventually contribute to the evolution of the ontology.

The classification ontology consists of two layers: global and local ontologies, as previously mentioned in the specification of the adaptive discovery method (Section 5.4.4.1).

6.3.1.1 Global Ontology

Global ontology, centrally managed, consists of commonly agreed concepts. It is used as references and provides semantic interoperability at a global level for resolving conflicts that may occur amongst local ontologies. The global ontology provides a skeleton for the user community to develop local ontologies.

6.3.1.2 Local Ontology

As the global ontology may not be sufficient for describing user interests, users can develop their new and customised concepts for their uses. The new local concepts are extensions and rooted from globally agreed concepts. They are specific to individual's needs. As different users may have different views on the same phenomenon, there may be inconsistency amongst local ontologies. However, local ontologies only have their values when there a certain level of consensus amongst them. If a local ontology is too different from others, it will hardly have any effect in the collective learning process. The inconsistency need to be resolved in order to improve the quality of the ontologies.

6.3.1.3 Resolving Inconsistency

At resource point of view, concepts described by an ontology are also resources themselves. Hence, they can be shared with other users in a community. When further classifying an interest, which is not sufficient described by the global ontology, a user may want to look for what other users in the community have done to further specify a global concept by using search capability of the P2P application. The user can then define his/her classification if not happy with the findings or import a good classification shared by other in the community. A good classification can be justified by its frequency of use in the community. The user then may want to share the classification with others in the community. The process of create, share and reuse will eventually will bring consensus to different local ontologies.

6.3.1.4 The Evolution of the Global Ontology

Through the conflict resolving process, some of the locally defined concepts will become well recognised and receive a high level of agreement amongst users in the local community. At some point, the usage (frequency of use) of a locally defined concepts will reach a certain threshold. It can be promoted to a global concept. The global ontology needs to be updated accordingly with the new concept and its attributes. At this point, the new

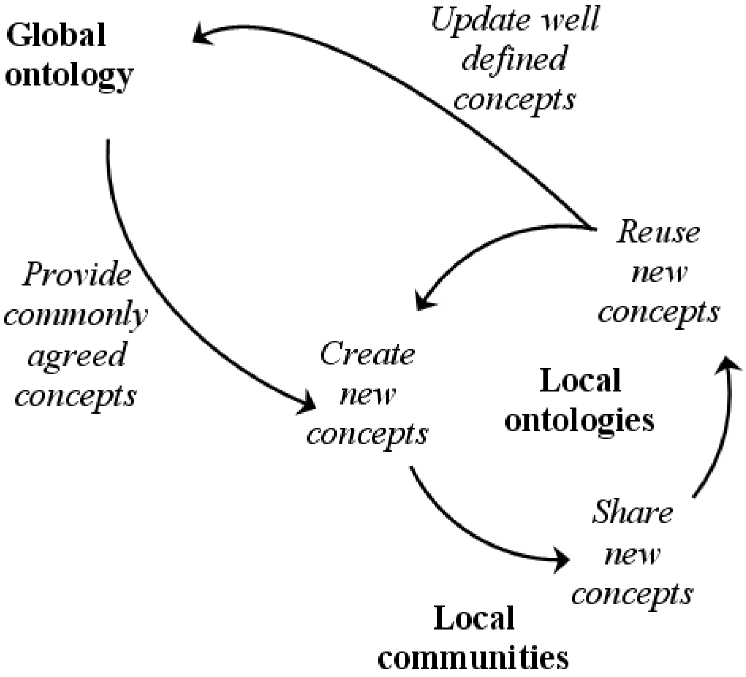


Figure 6.1: Evolution of the global ontology

concept has evolved itself from a user defined one to a globally agreed concept. Through this evolution process the global ontology will be richer and up-to-date with changes from the dynamic user environments.

The whole evolution process is illustrated in Figure 6.1. The global ontology provides commonly agreed concepts for uses in various local communities. With a local community, users can create new concepts that are not covered by the global ontology for their needs. The new concepts can be shared and then reused by others in the local community. Through the create/share/reuse process, a well defined concept can be recognised by the community. It can then be recommended for updating the global ontology.

Generally, the evolution process consists of two cycles. The larger cycle is when the global ontology is provided as reference concepts in local user communities, then the user communities provide back input to update the global ontology. The inner cycle exists within each local community, where new concepts will be defined and recognised as recommendations for global ontology.

6.3.2 Revising the Collaborative e-Science Architecture

A number of requirements have been recommended for the CeSA after the user experiment by potential user community such as security, annotation and provenance of shared documents. The proposal of the adaptive resource discovery for the CeSA also poses new requirements for the architecture. In this section, the discussion focuses on newly collected requirements and a revision on the CeSA to accommodate new requirements.

6.3.2.1 Requirements Revisited

Newly collected requirements for the CeSA came from three different sources: results of the user experiment, discussions with an expert from Reaction Kinetics after the experiment and the proposal of the adaptive resource discovery. These new requirements are more specific to particular functionalities of the CeSA than those collected before the design of the CeSA. This is because the potential users have experienced on the prototype of the CeSA. They have a clearer picture on what aspect of the CeSA that can help to improve their work.

During the user experiment, reported in Chapter 4, the participants expressed their interests in using file sharing function of the CeSA. They raised a number of issues that need to be addressed in order to better support their work. These issues are about tracking changes of shared documents and datasets and the security of P2P system to protect their personal resources.

Discussions with the expert from Reaction Kinetics community came up with two new requirements. Firstly, it is the ability to locate scattered resources and expertise for potential collaborations within the community using P2P resource discovery function of the CeSA. Secondly, there was also suggestion on a way to support coordinated file sharing at user end. This is a kind of lightweight workflow or pipeline for transferring files from one to another for carrying coordinated work in the community.

The CeSA is also required to have a new capability to accommodate the adaptive discovery method. As this discovery uses classification ontology to facilitate its learning process, the CeSA needs to provide a mechanism for manage this kind of ontology in decentralised P2P environments.

6.3.2.2 The Revised Architecture

All the changes required to accommodate the above new requirements are made in the P2P layer of the CeSA, particularly on the resource sharing function of the P2P application architecture. In the revised architecture, no changes are made to the specification of the Grid layer.

Figure 6.2 provides an overall description of the revised architecture of P2P applications of the CeSA.

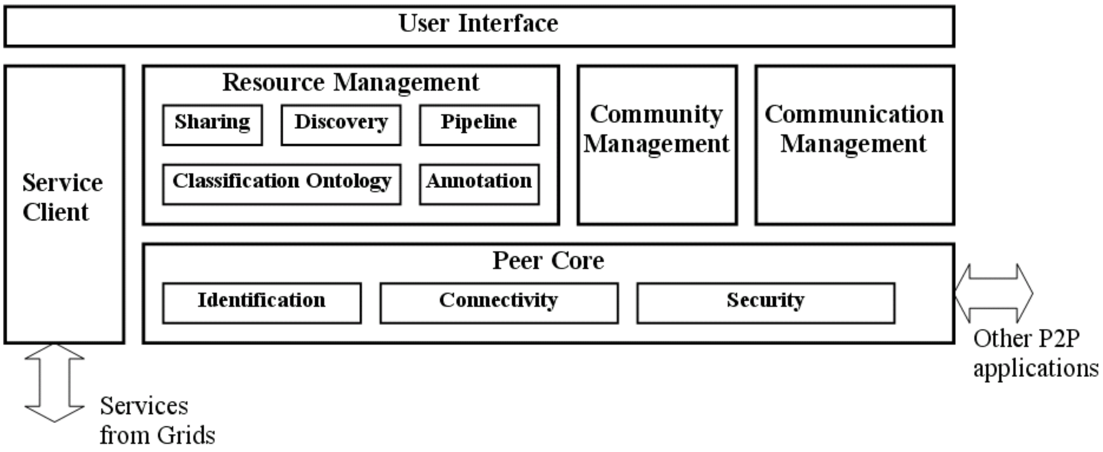


Figure 6.2: The revised architecture for P2P applications of the Collaborative e-Science Architecture

In the original version of the CeSA specified in Chapter 3, Service Publication and Discovery Agent and the File Sharing component of the P2P application are both related to resource sharing and discovery. Indeed, service information and files are resource themselves. Therefore, in the revised architecture, these two different kinds of resources will be uniformly managed by Resource Management component. The same sharing and discovery mechanism, which is the proposed adaptive method, will be used.

Under the Resource Management, a Pipeline subcomponent is proposed for managing flows of shared resources within working groups. Pipeline is a similar concept as workflow in Grid environments. However, a pipeline in a P2P environment will only be for transferring of lightweight resources.

The Classification Ontology subcomponent of Resource Management is for management of classification ontology, in order to support the adaptive resource discovery method. The classification ontology will also be used for annotation of shared resources, which will be carried out by Annotation subcomponent. The management of classification ontology will follow the method proposed in Chapter 5.

The Community Services in the original version of the CeSA is divided into two components in the revised architecture: Community Management and Communication Management. The Community Management will focus on formation of work groups, joining and leaving work groups. The Communication Management is separated from Community Management so that more advanced P2P communication tools, such as P2P voice chat and video conference, can be incorporated into P2P applications. These tools have a very important role in distributed collaborations.

Peer Core component is decomposed into three subcomponents. Identification subcomponent will deal with setting up the peer identity in a P2P network. Connectivity subcomponent will ensure that a peer is well connected to the network. Security subcomponent is introduced in the Peer Core for two purposes: firstly, to ensure that contents that are transferred through P2P connection are securely protected; secondly, for authentication of other peers, those request for shared resources.

Service Client and User Interface components will have the same functionalities as they do in the original version.

In a summary, the following are a number of potential areas for further work:

- Further work needs to be done on the underlying ontology infrastructure for the adaptive resource discovery. Evaluation is required for the evolutionary approach to ontology management in a P2P environment.

With regard to the CeSA in general, the following work are necessary to realise its revised architecture.

- Research also needs to be carried out on the security issue of the CeSA. This issue was not on scope of the work reported.

- A number of implementations also need to be done to support the above research activities. Firstly, the prototype needs to be upgraded for usability evaluation. The focus should be on user interface. This is important to attract attention from a wider user communities. Secondly, the adaptive discovery method and management of ontology also need be implemented in a working prototype. At the current stage, these two functions have not been incorporated in the P2P application prototype. Thirdly, it would be desirable to have the current prototyped services implemented on an operational Grid for usability evaluation and demonstration purposes.

- Further work on usability evaluation of the CeSA in a wider user community is also needed. Because of the limitation of resources, especially the difficulty in getting potential user communities to get involved in the evaluation process, the reported user evaluation was on a very small user group. For better evaluation results, there is the need for involvement of distributed user groups.

Appendix A

List of Abbreviations

Abbreviations	Full names
BADC	British Atmospheric Data Centre
CERN	European Particle Physics Laboratory
CeSA	Collaborative e-Science Architecture
CMCS	Collaboratory Multi-Scale Chemical Science
DHT	Distributed Hash Table
DOE	Department of Energy (US)
EGEE	Enabling Grids for e-Science
JISC	Joint Information Systems Committee
LHC	Large Hadron Collider
FTP	File Transfer Protocol
NERC	Natural Environment Research Council
NESC	National e-Science Centre
OGSA	Open Grid Services Architecture
OWL	Web Ontology Language
P2P	Peer-to-Peer
PrIMe	Process Informatics Model
QHT	Query History Tree
RDF	Resource Description Framework
SOA	Service Oriented Architecture

TTL	Time To Live
URI	Universal Resource Indication
VKP	Virtual Knowledge Park
VRE	Virtual Research Environment
WSDL	Web Services Description Language
WSRF	Web Services Resource Framework
WWW	World Wide Web
XML	eXtensible Markup Language

Appendix B

Glossary of Terms

Back-end Resources. Back-end resources are computationally intensive resources provided by computers with powerful computational capability such as CPU cycles, memory, huge-storage.

Complex Query. Queries that require complex processing techniques for matching with resources.

e-Science. e-Science is about global collaboration in key areas of science, and the next generation of infrastructure that will enable it.

Fan-out. The number of peers to which a peer will forward a query message.

Global ontology. A global ontology defines a set of terms that are globally recognised for classification of a wide range of interests in a target user community.

Grid. Grids are referred to as hardware and software infrastructures that provide consistent, pervasive, dependable, transparent access to high-end computing resources in a seamless, integrated computational and collaborative environment. High end computing resources provided by Grids can be CPU cycles, memory, storage and huge volume datasets.

Groupware. Groupware is software that supports and augments group work.

Ontology. An ontology is a published, more or less agreed conceptualisation of an area of content.

Portal. A portal is network service that brings together content from diverse distributed resources using technologies such as cross searching, harvesting, and alerting, and collate this into an amalgamated form for presentation to the user.

Peer. A peer is referred to a P2P application in a P2P network.

Peer's interests. A peer's interests are referred to interests of a user who uses a P2P application.

Peer-to-Peer (P2P). Peer-to-Peer is a network-based computing model for applications where computers share resources via direct exchanges between the participating computers.

Query. A query message is a message that carries an actual search query in P2P network. For example, a simple search query could be a list of keywords to look for resources that contain the keywords. A complex query could be a SQL query that requires a database management system to process. For simplicity, a query message is generally referred to as query or search query, unless explicitly stated.

Query matching. Query matching is the process of comparing a query with resources to find a match.

Query message. *See definition for Query*

Query routing. Query routing is the process of sending/forwarding a query message from one peer to another peer to look for resources.

Research Method. A research method is a particular activity such as analysing a survey or conducting a controlled experiment to do research.

Research Methodology. A research methodology is a combination of the process, methods, and tools which are used in conducting research.

Scientific Community A scientific community is a community consisting of members who have common interest on doing scientific research.

Semantic Grid. Semantic Grid is an application of the Semantic Web into Grid computing. The relationship of the Semantic Grid and the Grid connotes a similar relationship that exists between the Semantic Web and the Web.

Semantic Web. Semantic Web is an extension to the Web, in which information is given well defined meanings, better enabling computers and people to work in cooperation.

Semantic Web Services. Semantic Web services are Web services marked up with semantics.

Service. A software component that can be accessed via a network to provide functionality to a service requester.

Service Oriented Architecture (SOA). A style of building reliable distributed systems that deliver functionality as services, with the additional emphasis on loose coupling between interactive services.

Time-to-Live (TTL). The maximum number of hops that the query can travel within the network. This value is defined by the application.

Work Group. A work group consists of members who are working together to achieve a common goal.

Web Service. A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.

Appendix C

Questionnaire for Evaluation on the CeSA

Section 1: Peer-to-Peer Collaboration to Share/Exchange Working Data

Q1 For your research, do you need to share or exchange your working data (i.e. experimental results, output from a simulation, working papers, etc.) with other researchers?

Q2 If the above answer is YES, how do you share/exchange your working data currently?

Q3 The prototype provides a way of sharing data (files). How is this different from your current way of sharing?

Q4 Using the prototype to make a share, the shared file does not needs be copied to anywhere (for example, when sharing a file through a web server, the shared file needs to be uploaded to the server). Some amount of time required for file transfer can be saved. Does the time saved have any value to you? Does it have any benefit to your work?

Q5 Using the prototype for file sharing, the shared file on your computer can be accessed directly from other users. Do you feel comfortable with this behaviour?

Q6 If the answer for the previous answer is NO, could you please explain why?

Q7 As only one copy of the data is needed to make the share (there is no need for duplication of data to enable a share), some storage space can be saved. In term of storage, does this have any benefit for you and your group?

Q8 What is your overall impression on the sharing function provided by the prototype? In comparison with your current way of sharing, does this have any advantage? Which way of sharing would you prefer? Why?

Q9 What would you recommend us to improve this function of the prototype to better support your research?

Section 2: Using remote services for simulations and analyses

Q10 For your research, do you need to run any kind of simulation or analysis programs, such as programs in CHEMKIN packages?

Q11 If the previous answer is YES, how are you currently running these programs? (i.e. login to a remote computer, run on local desktop machine, use command lines, batch jobs, etc). On average, how long does it take to complete a simulation?

Q12 How did you have these programs? (build them yourselves, download from other research groups or buy, etc.)

Q13 If you develop your own tools, will you be happy to give them to other people in need? Currently, are you sharing any tools with other? Please justify your answer.

Q14 Do these programs often have graphical user interfaces?

Q15 What are special characteristics of input and output data? (Format, type, size, etc.)

Q16 Is the way of running a simulation or analysis provided by the prototype suitable for the kind of simulations or analyses required for your research? If NOT, please explain why?

Q17 From a service consumer point of view, would you prefer to run a simulation (an analysis or any kind of tools) using remote service, as in the prototype, or to download the code and run it on your machine? Please explain why?

Q18 From a service consumer point of view, running a remote service instead of downloading the code and running it locally would save some amount of storage space and CPU cycle on your desktop computer. Do these amounts have any value to you and your research? How valuable are these?

Q19 From a service provider point of view, by sharing a simulation (analysis or any kind of tools) in form of services, you will not give away the program (and/or source code) for the simulation to other users. On other words, you still have total control on your intellectual property. You can make the service available to other users or take it back. Would this be an important characteristic to encourage you to share more tools in form of services?

Q20 As a tool (i.e. simulation programs) provider, would you prefer to share a copy of a program to others users, then they will run it on their local machine? Or to make them available in form of service, so that you still have control over the service? Please explain further your choice?

Q21 In a collaborative sharing environment, the more people share their resources, the more resources you will receive. The more you give, the more others will receive. Consequently, the more you give the more you will receive. Do you agree with this statement? Would you be happy to give more so that you will get back more in return?

Q22 What would you recommend us to improve this feature of the prototype? Is there anything that should have done differently to better suit your research?

Section 3: Service publication and discovery

Q23 Currently, how do you search for tools (i.e. simulation programs) to support your research? (Search on the Internet, on a research forum, from colleagues, etc.)

Q24 Are you satisfied with your current way of searching for tools? What should be done to improve the quality of this kind of search?

Q25 If you need to share a tool with other researchers, how will you let other people know that you are sharing that tool? Do you think that your current method is effective? Please explain.

Q26 How do you compare your current way of publishing information about a shared tool to the way that the information about a service is published in the prototype? Which one is more preferable?

Q27 How do you compare your current way of discovering information about a shared tool to the way that the information about a service is discovered in the prototype? Which one is more preferable?

Section 4: General Feedback

Q28 What is your overall impression on the system? Which functionality is most likely to bring benefit to your work? Which one is the least?

Q29 If the functionalities provided in the prototype are implemented in a fully working system, do you think you and your research group will achieve some benefit from the system? Will you use the system after all?

Q30 Any further comments?

Appendix D

Responses Collected from the User Evaluation

The responses are presented in the table on the following pages.

Table D.1: Responses collected from the user evaluation using the questionnaire in Appendix C

Quest.	Participant 1	Participant 2	Participant 3
Q1	Yes	Yes	Yes - In the past I've shared experimental results and working papers
Q2	For fieldwork: Preliminary data and working papers are put onto the CAST (http://badc.nerc.ac.uk/community/) database on the BADC (http://badc.nerc.ac.uk/home). Final data is put onto the BADC in the appropriate fieldwork section. For modelling: Complete MCM box models can be downloaded from a dedicated MCM website (http://mcm.leeds.ac.uk/MCM/). Subsets of the model can also be searched for, extracted and download using a variety of specially designed web based internet tools. Others: Exchange of data via FTP sites and through shared directories (through the chemistry intranet).	Email, on disks, collaborative workspaces such as BADC, shared directories on networked computers	Experimental results ware shared on BADC website, other things I share via email

Q3	At the moment we have to upload/download files to/from separate computers/web servers in order to share them. The prototype would bypass this. Also you would get more security control using the prototype	More flexible once set up as it is easier to change permissions and a file does not have to be copied or moved. Currently, shared files within the group by shared directories which have to be set up by an administrator. Also, latest version of a file is available for download.	No need for an external website(& additional personal to move file from uploaded place to where people can view) No need for specific file format
Q4	This aspect is obviously useful however in the research I carry out sharing of large files is rare (at the moment). It could however be useful if, for example, various groups are updating different sections of the MCM. We could all share a complete MCM database file. Indexing and documenting changes would be an issue	Yes. Particularly useful if the file will be modified several times.	Yes - I makes the whole process easier
Q5	This aspect is obviously useful however in the research I carry out sharing of large files is rare (at the moment). It could however be useful if, for example, various groups are updating different sections of the MCM. We could all share a complete MCM database file. Indexing and documenting changes would be an issue.	Yes, provided there is control over which users can have access.	Yes, as it is down to me what I wish to share
Q6	N/A	N/A	N/A

Q7	Yes, this would create less files and less clutter on the computer	Yes. Some data files are large so only having one copy would be beneficial. Also useful when the file is updated as there is no need to remember to update a copy.	Yes
Q8	I like it, I would certainly use it for certain applications. The new share function would save time and space and be more convenient. However, security would be an important issue.	Useful to be able to share files from own computer without having to copy to shared directory, disks etc.	It is quite useful - Its much quicker to share data especially if data needs to be worked on by several people. It is easy to exchange copies of the updated work. On the BADC it is just the original raw data that is shared & then everything else is worked on separately. This way of sharing does seem more useful. Would save a lot of time - both in the sharing process and working on data
Q9	Again, flexible security controls on your own files.	Include the date on which the file was last modified so it is easy to check if it has been updated since it was downloaded. Quick way to see who you are currently sharing with. A record of who has downloaded your files.	When sharing files to specific groups, it would be beneficial to select node after the file is chosen just to ensure the file is shared to the correct groups of people.
Q10	Yes	Yes. E.g. Facsimile	Facsimile analysis program

Q11	For TUV: Need to download program and input files and run using a suitable Fortran compiler. Run time c.a. 2-3 minutes For MECHGEN: Telnet to the server and run through a command line. Also use PUTTY to add/retrieve files to/from the server. Run time c.a. a few seconds MCM: Download latest mechanism form the website and add to a model file which is run on the windows version of FACSIMILE. Run time: anything from minutes to days.	Local desktop machines. Hours	Local desktop machine Simulations can take minutes to weeks
Q12	Downloaded most up to date versions from a dedicated website. Developed certain tools ourselves and placed on the web for access by others	Bought	Download and buy
Q13	Yes, the MCM. The MCM has been developed as a publicly accessible chemical tool.	Yes	If I developed new tools I would be happy to share but at the moment, I am not.
Q14	No		Yes

Q15	Currently the input files from the MCM are specifically designed for use with the FACSIMILE integrator. However, a tool on the website can convert the FACSIMILE format into Fortran and XML versions. The output files are in the form of text files (ASCII) as are the input files for TUV. Sizes of files vary depending on the size of the mechanism used, typically no bigger that a couple of Mega Bytes.		Input and output are in text files
Q16	Running simulations in the way would work for both MCM and TUV as long as you had control of the input files. Sometimes you would require many input files for each simulation therefore would require some kind of indexing system. Also it would be desirable to run multiple simulations simultaneously.	Yes, but simulations take a long time to run	Yes, in terms of the input and output files

Q17	I think it would be most useful if you could submit your input files to a server and run your simulations remotely, freeing up valuable computer space and time. However, sometimes the source code of the program needs to be altered. If both options were available (remote access and download) that would be useful.	Using remote service, as would leave my machine free for other things	Using a remote server would free up my computer for other uses. So, for this reason it would be preferable.
Q18	If you want to run a full MCM box model and the PTM model this would be extremely valuable, especially if you can run multiple large scale simulations at the same time (utilising the power of the Grid).	Would be very useful when running large simulations on a remote service as these can tie up my computer for many hours	It has some value - although I think freeing up computer time is the most beneficial aspect

Q19	This would not be useful in terms of the MCM as we make the general MCM available to the public anyway. However some of our specialised box models we would want security control of. In terms of making a general European MCM box model available to run using FACSIMILE, we could create a general input file so that the user can vary the initial conditions. This would be something similar to the 'quick' version TUV which is available online (www.acd.ucar.edu/TUV/) along with the downloadable version.	Yes, although would need enough computing power for the services	Yes
Q20	Running a service would require a lot of computing resources (supplied through the Grid?) and possibly man power. However supplying downloadable models for people to adapt and perhaps facilitating the running of a simple box model with a basic initial input file (c.f. TUV) could provide a useful service. Again could run into licensing problems if let people run simulations using a commercial program.	Share a copy of the program, would need less computing resource	Make it available in the form of a service - then would be able to improve the service at regular time.

Q21	Yes, however I think this type of approach will require a different type of think with respect to how data is shared and processed in science.	Would be happy to share provided that I am receiving useful resources. However, I would not be happy sharing resources if other people were not	I agree with this statement and feel that this would be the ease in my working environment. And yes, I would be happy to give more, therefore.
Q22	Multiple job submission to run simultaneously Better indexing and documentation of the input and output files (can you see how calculation is progressing? Visualisation of job queues). Flexible control of who can and cannot see your shared files		
Q23	e-mail, internet search, colleagues, conferences/talks.	Colleagues, internet	Search on internet and colleagues' recommendations
Q24	Should be better advertisement of tool on the internet. Could subscribe to a mailing list with regular updates on the latest tool via e-mail. It is also difficult to know if you are using the most up to date version of the tool or not.	Can be difficult finding specific tools on internet. It would be better to search in related forum.	It has been sufficient, although some type of database would be useful.
Q25	Advertise tools and updates on a dedicated website (e.g. MCM). Advertise site/tool at conferences and talks Advertise via mailing list, message boards and e-mail.	Tell colleagues, publish on web. Not very effective as only limited people would be aware of it.	Group emails are usually sent to inform people of required tools. It is effective, but people still need to figure out how to use the tools - so a link to a remote server would be beneficial.

Q26	If the wide ranging user group is included in the share lists then the prototype system would be direct. Again can let only let certain members and user groups know about updates and new tools as well (security issue).	Don't really share tools at the moment	This prototype is more useful as there are full links to remote server.
Q27	Same as above!	Prototype would be better if enough people in a related filed were using	The prototype as it is linked directly to remote server
Q28	As a first basic prototype the potential of such a system is clear to see. All of the facilities added so far show promise. I would like to see the remote service operated using multiple simulations simultaneously and get a better feel as to how easy it would be to use.	Good, file sharing would be most useful	The sharing file function would be most useful in day to day use., but also function have possibilities and potential in my work
Q29	I think that our group would certainly use such a system if it proved to be the way forward in e-science (which I feel it is) and the scientific community embraced the use of such a system.	Yes	A fully working system would benefit the atmospheric chemistry group, provided it was widely accepted by the whole community.

Q30	I think that you would benefit in testing this prototype (and subsequent versions) within a variety of scientific situations and groups as some of the aspects of this system may be of more interest to other groups. Two such groups I can think of that you should approach are the global and regional modelling groups in the Environment centre (Dr Matt Evans and Dr Steve Arnold) and the high-level computational group in the Chemistry department (Dr Liming Wang). A meeting with Professor Pilling should bring up other groups who it would be useful for you to approach to go through this testing/questionnaire process.		

Appendix E

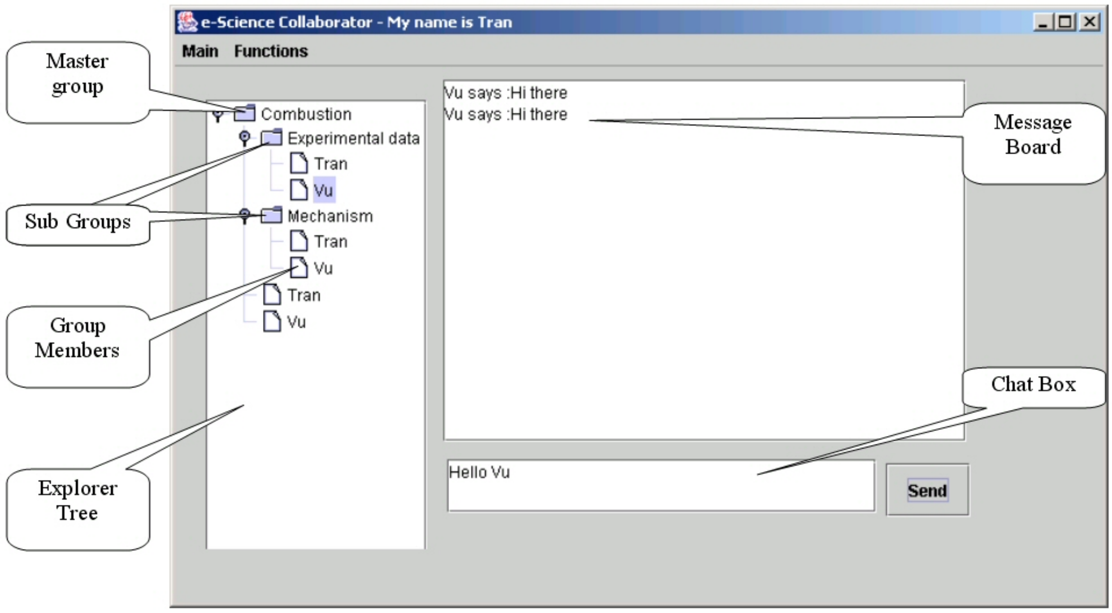
Guides for Using the e-Science Collaborator during the User Evaluation

This guide is designed for experiment on e-Science Collaborator prototype. It should be used in conjunction with the questionnaire provided.

The aim of this experiment is to collect your reflection on the functionalities of e-Science Collaborator provided with the prototype in comparison with your current working practice. As the prototype is designed mainly for experiments on the functional aspects of the system, there is still limitation in the system user interface. Please ensure that the e-Science Collaborator prototype is properly started before using the guide described in next sections.

The main application window, after being started up and initialised, should be similar to the following picture (Figure E.1).

The box on the left of Figure E.1 describes a view of the community. In this example, the master group, or community, is named as Combustion, which has two sub groups: Experimental data and Mechanism. “Tran” and “Vu” are members of all of these groups. The big box on the top right is a message board, which displays system message as well as chat message received from other members. The chat box below the message board is where to type in the message to send to other members. To send a message to a group or



e-Science Collaborator - My name is Tran

Vu says :Hi there
Vu says :Hi there

Hello Vu

Figure E.1: e-Science Collaborator main window

a specific member, select the target audience on the left and type in the message in chat box, then click send button.

1. P2P collaboration to share/exchange working data

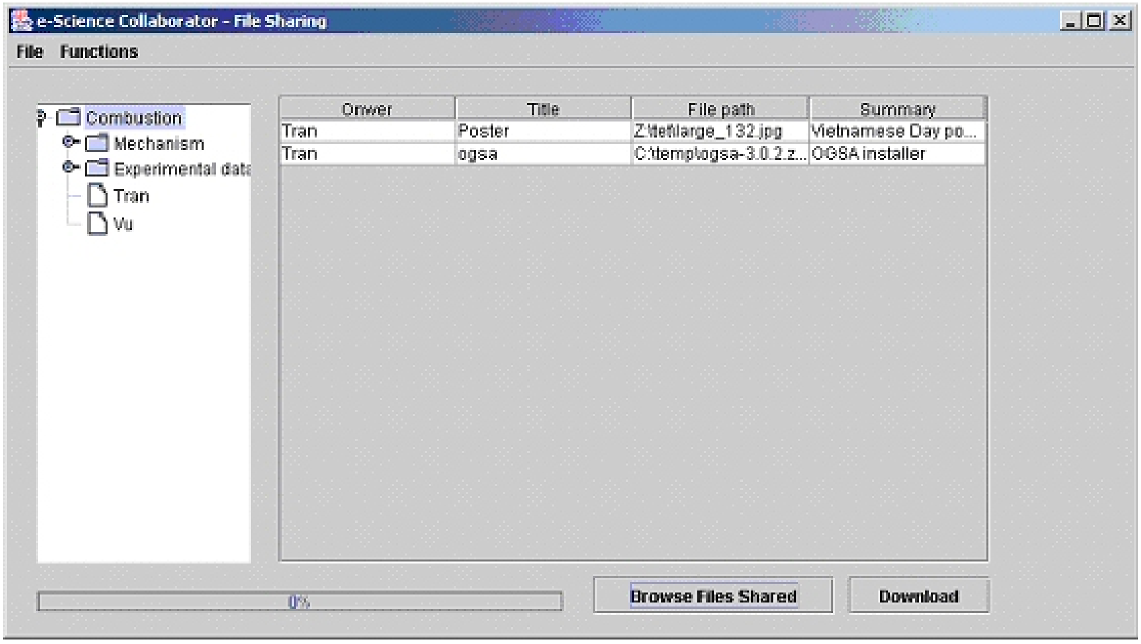
This function of e-Science Collaborator allows users to share/exchange files with each other. Using this function, a user can share a file to a particular working group or to any one in public. The users can also search or browse files shared by other users in the network.

To experimenters: please answer question 1 and 2 of the questionnaire before proceeding further.

Start using this function by going to "**Functions ->File Sharing**" on the menu. A window similar to the picture in Figure E.2 should appear.

a. Sharing files

Files need to be added to **Share Database** before it can be shared to other users or working groups. In order to add a file to the share database, select "**Add File to Share Database**" on **Functions** menu. The following window (Figure E.3) will be displayed.



Onwer	Title	File path	Summary
Tran	Poster	Z:\tet\large_132.jpg	Vietnamese Day po...
Tran	ogsa	C:\temp\ogsa-3.0.2.z...	OGSA installer

Figure E.2: File sharing window

This window allows users to enter details of new files into the database. The information about a file includes File name, its title, keywords and a short abstract about the file. If more than one keywords need to be entered for the file, they should be separated by semi-colons.

A file can be shared to a specific working group and to anyone in public. To share a file to a working group, select a working group on the explorer tree on the left, the go to "**Functions->Share a File**" menu from File Sharing widow. The dialog as in Figure E.4 will appear.

This dialog displays a list of files have been added to Share Database. Select a file in the list then click on **SelectedNode** button to enable a share. Sharing a file to public is similar to sharing to a specific group. However, users do not have to select a group before starting Share file dialog. At the last step, the **Public** button should be selected instead of **SelectedNode**.

You can view all files you are currently sharing to a working group by selecting an interest group, then "**View Files Being Shared**". The list of files being shared to the selected group will be displayed on the table on the right of File Sharing window.

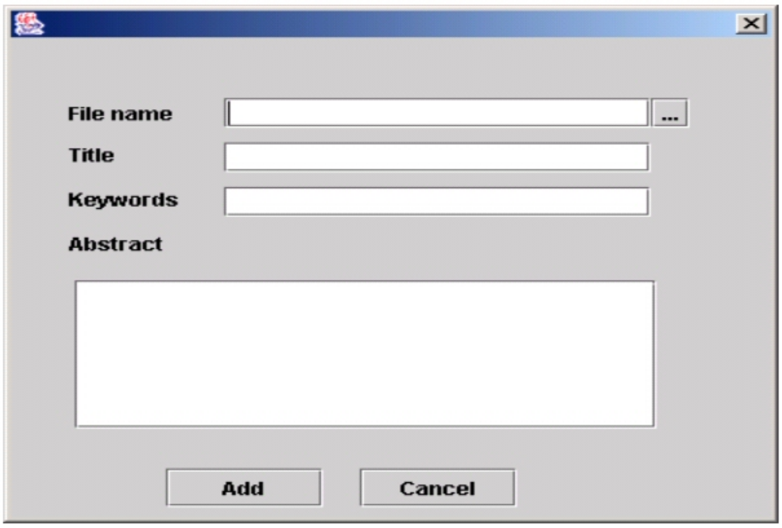


Figure E.3: File details

b. Browsing shared files

In order to view files shared by all users to a specific group, select the group on the explorer tree, then click on "**Browse Files Shared**" button at the bottom of File Sharing window. The application will send query to other peers (applications) on the network and then display results on the table on the right.

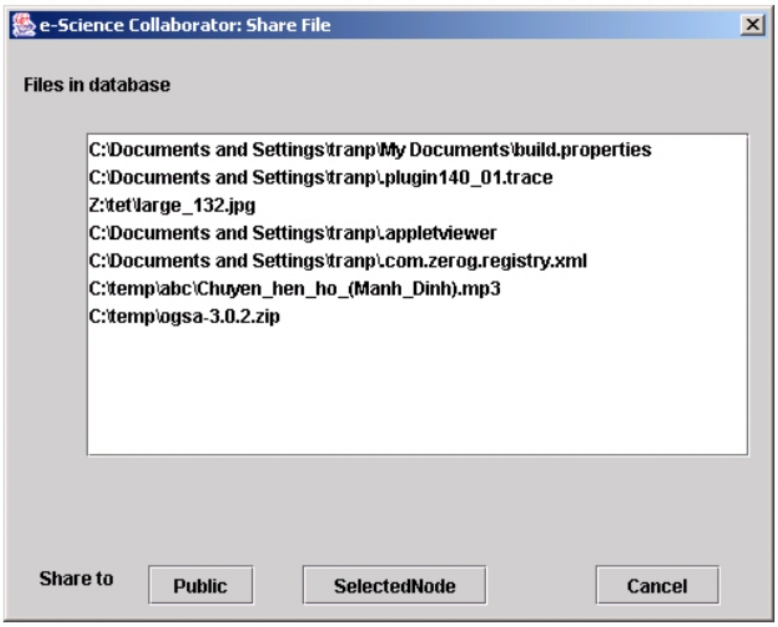
To download a file of interest, what you need to do is to select a file on the table, and the click on the "**Download**" button. The application will ask for file name and location for the file to be saved, then, start downloading.

c. Revoking a share

In case you do not want to share a file (to public or to a particular group) any more, you can revoke the share by selecting option "**Revoke a File from Share**" on **Functions** menu (shown in Figure E.5). The procedure is similar to sharing a file. To revoke a share from a group, select a group on the explorer tree, then, start this option on the **Functions** menu, select the file you want to put away from sharing, finally, click on **SelectedNode** button. Revoking a share from public can be done similarly.

d. Search for shared files

Instead of browsing, you can also looking for files shared by other users by using search function of the application. To start this function, close the File Sharing window to return



C:\Documents and Settings\tranp\My Documents\build.properties
C:\Documents and Settings\tranp\.plugin140_01.trace
Z:\tet\large_132.jpg
C:\Documents and Settings\tranp\.appletviewer
C:\Documents and Settings\tranp\.com.zerog.registry.xml
C:\temp\abc\Chuyen_hen_ho_(Manh_Dinh).mp3
C:\temp\ogsa-3.0.2.zip

Figure E.4: File details

to the main, then, select “**Search**” option of the **Functions** menu on the main window. The window as in Figure E.6 should appear.

The search function uses keyword search. It will search for any shared files (by all users) that you have access to and match the keywords entered.

To experimenters, please follow the above guide to share your files with other experimenters. Please also use browse, search and download functions to get files being shared by other experiments to your computer. After you have done all these, please answer questions from 3 to 9 on the questionnaire.

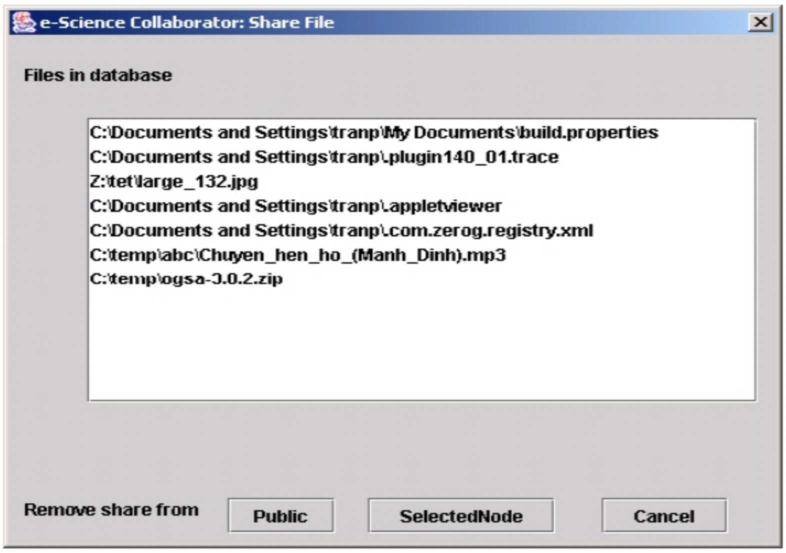
2. Using remote services for simulations and analyses

This function of e-Science Collaborator allows you to run simulations and analyses from CHEMKIN package, such as Senkin and PSR, and Kinalc on a remote computer via Grid Services.

To experimenters: Please answer questions from 10 to 15 before proceeding further.

To start this function, select “**Grid Services**” option on the **Functions** menu of main window. A window similar to the Figure E.7 should appear.

If nothing appears on the table, please replace the string “**localhost**” of Service URL by this IP address: **129.11.147.50**. This is the address of the server which provides ser-



C:\Documents and Settings\tranp\My Documents\build.properties
C:\Documents and Settings\tranp\.plugin140_01.trace
Z:\tet\large_132.jpg
C:\Documents and Settings\tranp\.appletviewer
C:\Documents and Settings\tranp\.com.zerog.registry.xml
C:\temp\abc\Chuyen_hen_ho_(Manh_Dinh).mp3
C:\temp\ogsa-3.0.2.zip

Figure E.5: Revoke a file from sharing

vices.

Alternatively, you can type this full URL in Service URL field: "http://129.11.147.50:8080/ogsa/services/core/registry/ContainerRegistryService".

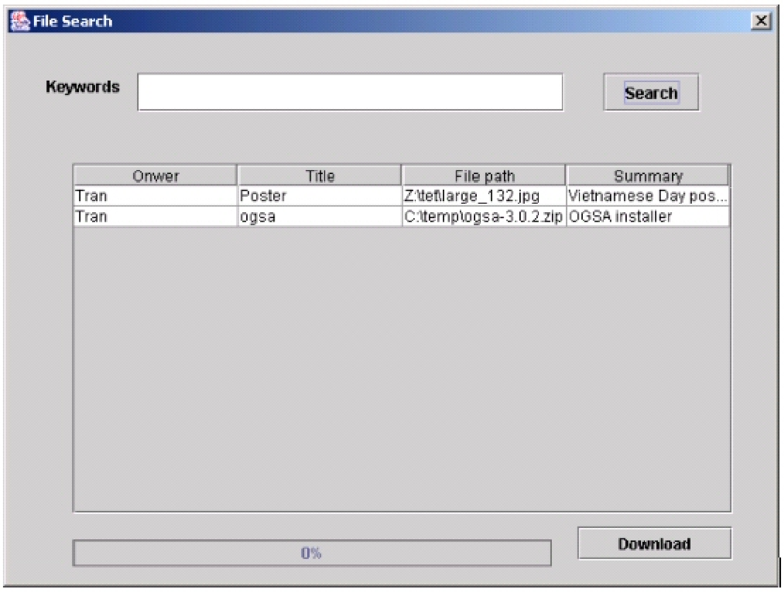
Grid services built from CHEMKIN package are highlighted in the table. There are two type of services displayed in the table: Service Factory and Service Instance.

A service factory is run to generate service instances of a service. These service instances will deal with users' work.

For example, in order to run Senkin service, first, we you generate a Senkin Service Instance, if there is none, by selecting SenkinService Factory on the table, then clicking on **CreateInstance** button. If the new instance does not appear automatically, press Browse button on top-right corner to refresh the table. Running the service by selecting the newly created instance, and then clicking on "**Start Chemkin Client**" button. A new dialog as in Figure E.8 will appear to deal with your request.

The dialog shown in Figure E.8 is a common service client used for all service in CHEMKIN package. Show button is used to tell the service display input files required to run the services. These input files need to be submitted to the server, by selecting the file, and then, clicking on **Submit** button. After all required files have been sent to the server, you need to click on **Execute** button to run the service. The processing may take a while. The results will be displayed in **Outputs produced** box. Select output files in this box and click on **Retrieve** button to download these results to your local computer.

To experimenters: please follow the above step to run one or two service, then an-



Onwer	Title	File path	Summary
Tran	Poster	Z:\tef\large_132.jpg	Vietnamese Day pos...
Tran	ogsa	C:\temp\ogsa-3.0.2.zip	OGSA installer

Figure E.6: File search window

swer questions from 16 to 22 of the questionnaire. Sample input files are provided in the installation directory.

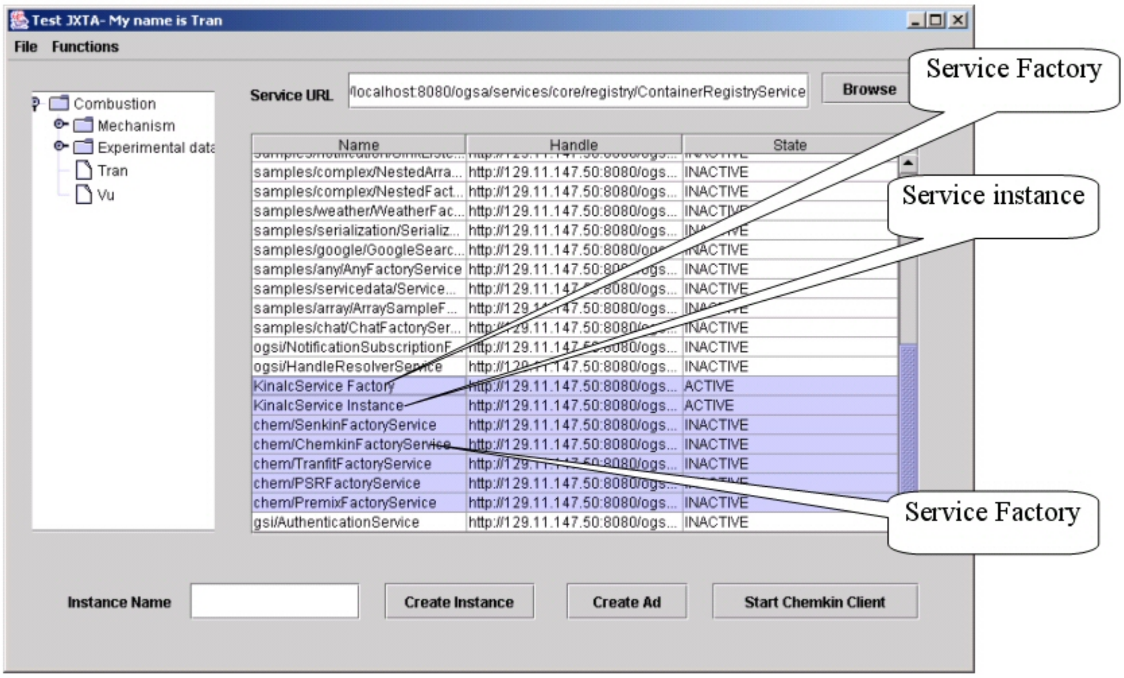
3. Service publication and discovery

This functions of the system allow users to publish information about services they know to others users in the community. It also allows users to search for services that are published by other users. This is an important characteristic of e-Science Collaborator, as in reality, there will be many service providers, and you might not be aware of the existence of services you are in need.

a. Publishing service information

For example, if you want to tell other people in your community about Senkin service that you have found and use. What you need to do is select the Senkin service (factory or instance) on the table, then, click on “**Create Ad**” button. A window as in Figure E.9 will appear for you to enter information about your publication.

There are a lot of details for you to enter, but they are all not mandatory except for Service URI and Service Name. After entering all the details you want, select on a working group on the left and click on **Publish** button to send the information to that group. You



Service URL: /localhost:8080/ogsa/services/core/registry/ContainerRegistryService

Name	Handle	State
samples/complex/NestedArra...	http://129.11.147.50:8080/ogs...	INACTIVE
samples/complex/NestedFact...	http://129.11.147.50:8080/ogs...	INACTIVE
samples/weather/WeatherFac...	http://129.11.147.50:8080/ogs...	INACTIVE
samples/serialization/Serializ...	http://129.11.147.50:8080/ogs...	INACTIVE
samples/google/GoogleSearc...	http://129.11.147.50:8080/ogs...	INACTIVE
samples/any/AnyFactoryService	http://129.11.147.50:8080/ogs...	INACTIVE
samples/servicedata/Service...	http://129.11.147.50:8080/ogs...	INACTIVE
samples/array/ArraySampleF...	http://129.11.147.50:8080/ogs...	INACTIVE
samples/chat/ChatFactorySer...	http://129.11.147.50:8080/ogs...	INACTIVE
ogsi/NotificationSubscriptionF...	http://129.11.147.50:8080/ogs...	INACTIVE
ogsi/HandleResolverService	http://129.11.147.50:8080/ogs...	INACTIVE
KinalcService Factory	http://129.11.147.50:8080/ogs...	ACTIVE
KinalcService Instance	http://129.11.147.50:8080/ogs...	ACTIVE
chem/SenkinFactoryService	http://129.11.147.50:8080/ogs...	INACTIVE
chem/ChemkinFactoryService	http://129.11.147.50:8080/ogs...	INACTIVE
chem/TranfitFactoryService	http://129.11.147.50:8080/ogs...	INACTIVE
chem/PSRFactoryService	http://129.11.147.50:8080/ogs...	INACTIVE
chem/PremixFactoryService	http://129.11.147.50:8080/ogs...	INACTIVE
gsi/AuthenticationService	http://129.11.147.50:8080/ogs...	INACTIVE

Figure E.7: Grid Services

can publish service information to more than one group. After finishing your publishing, click **Close** button to close this window.

b. Discovering service information

You have published information about services you know to other people in your community. It is now you turn to search for what other people have published.

Select "**Search for Services**" option on the **Functions** menu of the Grid Services window. The window similar to Figure E.10 should appear.

The search criteria you need to enter for your search is similar to those you have entered to publish service information. You only need to enter the criteria that you are looking for, for example service name. Then select the working group you want to search within it on the left, and finally click **Search** button. It might take a while to find results. If any results are found, they will be listed on the "**Search Results**" table with their matching score, which tell how closely the information about the service matches you criteria. From the list of results, you can select and start (**Start Client**) a service if the selected result is a service instance, or, browse service instances available if it is service factory. In turn, you can publish information about services you have found from the Service Browser window (see Figure E.11).

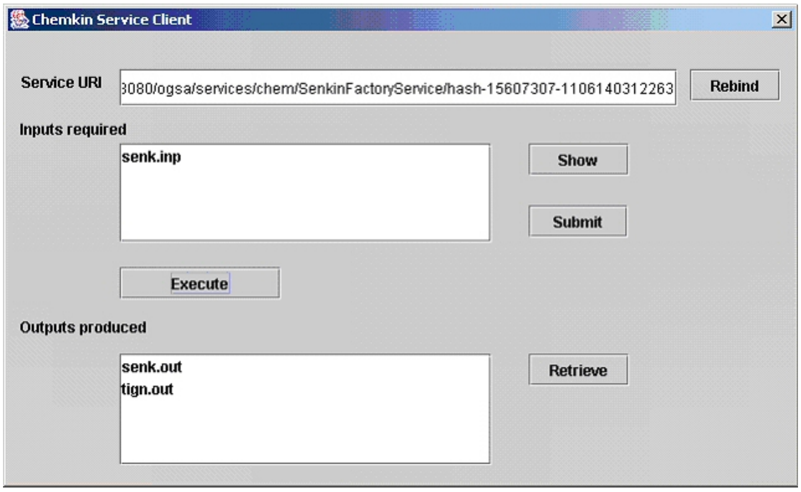
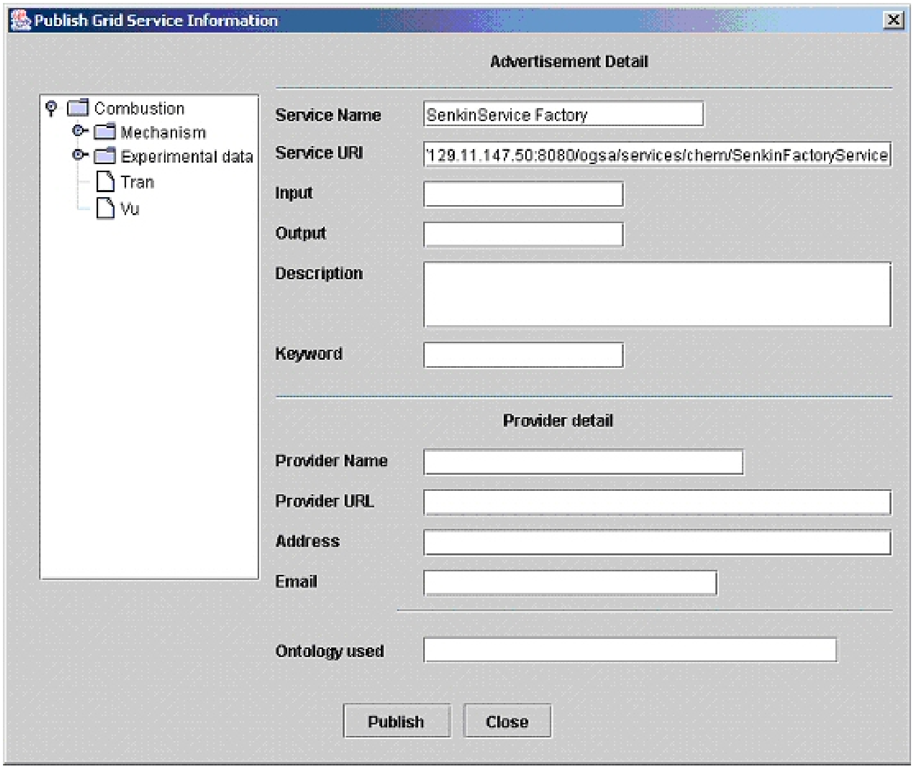


Figure E.8: Chemkin Service Client

To experimenters: please publish information about one or some of the services provided, then try to search for service published by others. Then, please answer all questions in section 3 of the questionnaire.

4. General Feedback

Please also answer all questions in section 4.



Advertisement Detail

Service Name: SenkinService Factory

Service URI: '129.11.147.50:8080/ogsa/services/chem/SenkinFactoryService

Input:

Output:

Description:

Keyword:

Provider detail

Provider Name:

Provider URL:

Address:

Email:

Ontology used:

Figure E.9: Publishing service information

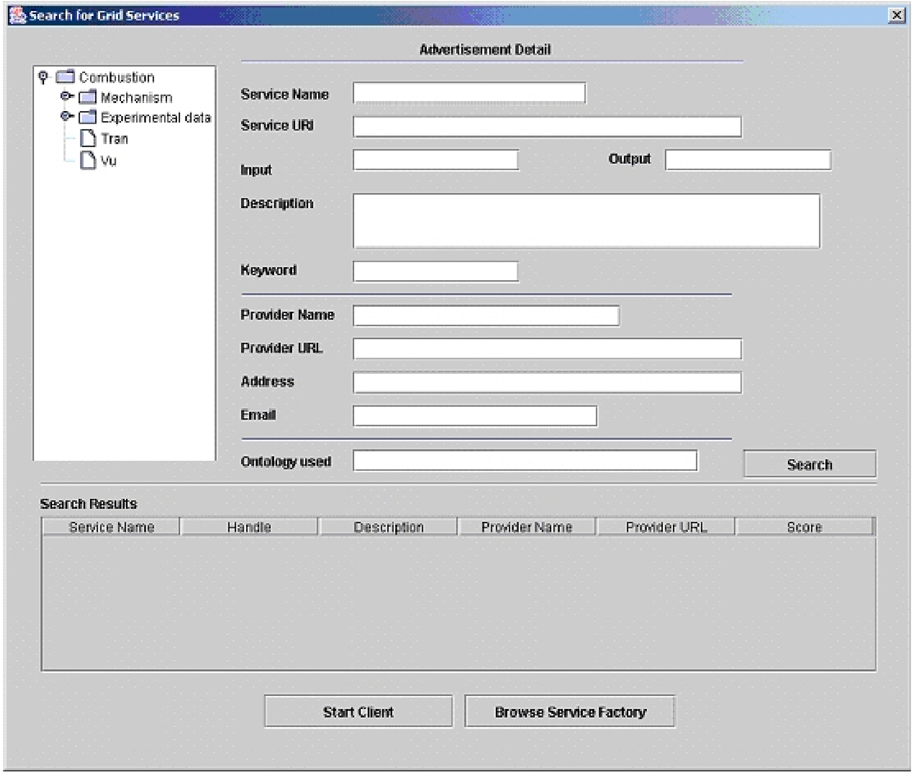
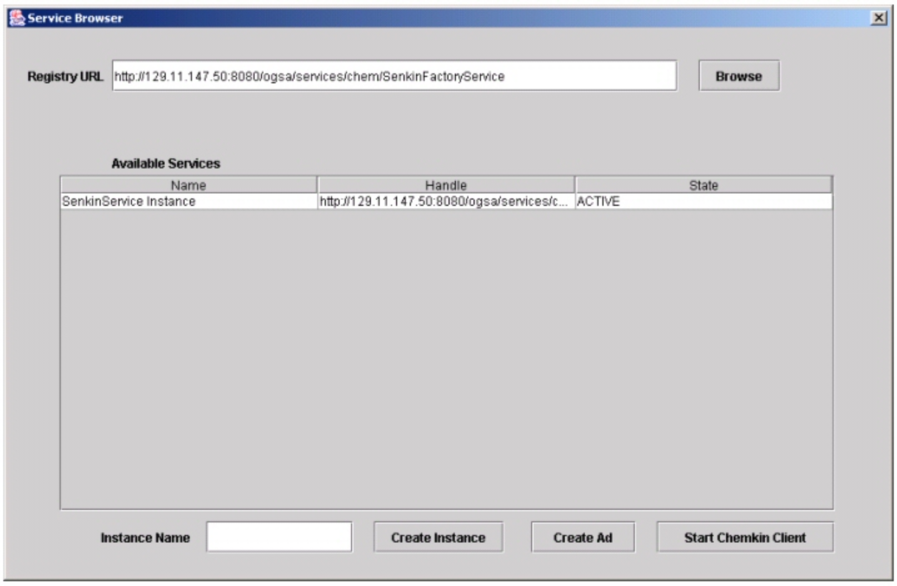


Figure E.10: Discovering services



Registry URL http://129.11.147.50:8080/ogsa/services/chem/SenkinFactoryService

Available Services

Name	Handle	State
SenkinService Instance	http://129.11.147.50:8080/ogsa/services/c...	ACTIVE

Figure E.11: Service Factory Browser

Bibliography

Aberer, K., P. Cudre-Mauroux, A. M. Ouksel, T. Catarci, M.-S. Hacid, A. Illarramendi, V. Kashyap, M. Mecella, E. Mena, E. J. Neuhold, O. D. Troyer, T. Risse, M. Scannapieco, F. Saltor, L. Santis, S Spaccapietra, S Staab & R. Studer (2003), Emergent semantics principles and issues, *in* Y.Lee, J.Li, K.-Y.Whang & D.Lee, eds, 'Database Systems for Advanced Applications: 9th International Conference', Lecture Notes in Computer Science, Springer-Verlag, Jeju Island, Korea, pp. 25–38.

Adamic, Lada A., Rajan M. Lukose, Amit R. Puniyani & Bernardo A. Huberman (2001), 'Search in power-law networks', *Physcal Review E* **64**.

Allan, R.J. (2006), 'Hpcportal'.
URL: *http://esc.dl.ac.uk/HPCPortal (last accessed 25/09/2006)*

Allan, Rob, Alison Allden, David Boyd, Rob Crouchley, Nicole Harris, Liz Lyon, Alan Robiette, D. De Roure & Scott Wilson (2005), Roadmap for a uk virtual research environment, Technical report, JCSR VRE Working Group.

Allan, Rob, Chris Awre, Mark Baker & Adrian Fish (2004), Portals and portlets 2003, Technical report.
URL: *http://www.grids.ac.uk/Papers/Portals/portals.pdf*

Anderson, Terry & Heather Kanuka (1997), On-line forums: New platforms for professional development and group collaboration, Technical report, US Department of Education.

Andriessen, J. H. Erik (2003), *Working with Groupware: Understanding and Evaluating Collaboration Technology*, CSCW, Springer, London.

Atkins, D. E., Kelvin K. Droegemeier, Stuart I. Feldman, Hector Garcia-Molina, Michael L. Klein, David G. Messerschmitt, Paul Messina, Jeremiah P. Ostriker &

Margaret H. Wright (2003), Revolutionizing science and engineering through cyberinfrastructure: Report of the national science foundation blue-ribbon advisory panel on cyberinfrastructure, Technical report, National Science Foundation.

Awre, Chris (2003), 'Portals: Frequently asked questions'.
URL: *http://www.jisc.ac.uk/index.cfm?name=ie_portalsfaq (last accessed 25/09/2006)*

BADC (2005), 'The nerc datagrid'.
URL: *http://ndg.badc.rl.ac.uk/ (last accessed 25/09/2006)*

BADC (2006), 'British atmospheric data centre'.
URL: *http://badc.nerc.ac.uk/ (last accessed 30/09/2006)*

Baker, Mark, Rajkumar Buyya & Momenico Laforenza (2002), 'Grid and grid technologies for wide area distributed computing', *Software - Practice and Experience* **32**(15), 1437–1466.

Barkai, D. (2001), *Peer-to-peer computing: technologies for sharing and collaborating on the net*, Intel Press.

Baulch, D. L., C. T. Bowman, C. J. Cobos, R. A Cox, T. Just, J. A. Kerr, M. J. Pilling, D. Stocker, J. Troe, W Tsang, R. W. Walker & J. Warnatz (2005), 'Evaluated kinetic data for comubstion modeling: Supplement ii', *Journal of Physical and Chemical Reference Data* **34**(3), 757–1397.

Beaver, D. deB. & R Rosen (1978), 'Studies in scientific collaboration: Part i. the professional orgins of scientific co-authorship', *Scientometrics* **1**.

Beaver, D. deB. & R Rosen (1979*a*), 'Studies in scientific collaboration: Part ii. scientific co-authorship, research productivity and visibility in the french scientific elite, 1799 - 1830', *Scientometrics* **1**(2).

Beaver, D. deB. & R Rosen (1979*b*), 'Studies in scientific collaboration: Part iii. professionalization and the natural history of modern scientific co-authorship', *Scientometrics* **1**(3).

Berners-Lee, Tim (2006), 'Bio'.
URL: *http://www.w3.org/People/Berners-Lee (last accessed 28/07/2006)*

Berners-Lee, Tim, James Hendler & Ora Lassila (2001), 'The semantic web', *The Scientific American* .

Bhandarkar, Milind, Gila Budescu, William F. Humphrey, Jesus A. Izaguirre, Sergei Izrailev, Laxmikant V. Kale, Dorina Kosztin, Ferenc Molnar, James C. Phillips & Klaus Schulten (1999), Biocore: A collaboratory for structural biology, *in* A. G.Bruzzone, A.Uchrmacher & E. H.Page, eds, 'SCS International Conference on Web-Based Modeling and Simulation', San Francisco, California, pp. 242–251.

BioCoRE (2006), 'A biological collaborative research environment'.
URL: *http://www.ks.uiuc.edu/Research/biocore/ (last accessed 30/09/2006)*

Bly, Sara, Steve R. Harrison & Susan Irwin (1993), 'Media spaces: Bringing people together in a video, audio, and computing environment', *Communication of the ACM* **36**(1), 28–47.

Buyya, Rajkumar (2002), 'Grid computing info centre: Frequently asked questions (faq)'.
URL: *http://www.gridcomputing.com/gridfaq.html (last accessed 30/09/2006)*

Buyya, Rajkumar, D. Abramson & J. Giddy (2000), Nimrod/g: An architecture for a resource management and scheduling system in a global computational grid, *in* 'The 4th International Conference on High Performance Computing in Asia-Pacific Region', Beijing, China.

Chawathe, Yatin, Sylvia Ratnasamy, Lee Breslau, Nick Lanham & Scott Shenker (2003), Making gnutella-like p2p systems scalable, *in* 'Proceedings of the ACM SIGCOMM 2003', pp. 407–418.

Chetty, Madhu & Rajkumar Buyya (2002), 'Weaving computational grids: How analogous are they with electrical grids', *Computing in Science and Engineering* **4**(4), 41–71.

Chinnici, Roberto, Jean-Jacques Moreau, Arthur Ryman & Sanjiva Weerawarana (2003), Web services description language (wsdl) version 2.0 part 1: Core language, Technical report, W3C.

Chohan, D, A Akram & Rob Allan (2005), Grid middleware portal infrastructure, *in* 'the 3rd international workshop on Middleware for grid computing', ACM, Grenoble, France.

Chu, Heting & Marilyn Rosenthal (1996), Search engines for the world wide web: A comparative study and evaluation methodology, *in* 'ASIS 1996 Annual Conference Proceedings', American Society for Information Science. Last accessed 07/09/2006.
URL: *http://www.asis.org/annual-96/ElectronicProceedings/chu.html*

CMCS (2004), 'Technical details'.
URL: *http://cmcs.org/technical.php (last accessed 30/09/2006)*

CMCS (2005), 'Collaboratory for multi-scale chemical science'.
URL: *http://cmcs.org/home.php (last accessed 30/09/2006)*

Cohen, Edith, Amos Fiat & Haim Kaplan (2003), Associative search in peer to peer networks: Harnessing latent semantics, *in* 'Proceedings of the IEEE Infocom'03', San Francisco.

CollabNet (2006), 'Jxta - get connected'.
URL: *http://www.jxta.org (last accessed 25/09/2006)*

CONDOR (2006), 'Condor high throughput computing:http://www.cs.wisc.edu/condor/'.

Crocker, Dave (2006), 'Email history'.
URL: *http://www.livinginternet.com/e/ei.htm (last accessed 01/10/2006)*

Czajkowski, K., D. F. Ferguson, I. Foster, J. Frey, S. Graham, I. Sedukhin, D. Snelling, S. Tuecke & W. Vambenepe (2004), The ws-resource framework, Technical report, Computer Associates International, Inc., Fujitsu Limited, Hewlett- Packard Development Company, International Business Machines Corporation and The University of Chicago.
URL: *http://www.globus.org/wsrf/specs/ws-wsrf.pdf*

Czajkowski, K., D. Ferguson, I. Foster, J. Frey, S. Graham, T. Maguire, D. Snelling & S. Tuecke (2004), From open grid services infrastructure to ws-resource framework: Refactoring & evolution, Technical report, Fujitsu Limited, International Business Machines Corporation, The University of Chicago.
URL: *http://www.globus.org/wsrf/specs/ogsi_to_wsrf_1.0.pdf*

De Roure, D., M. Baker, N. R. Jennings & N. Shadbolt (2003), The evolution of the grid, *in* F.Berman, G.Fox & A. J. G.Hey, eds, 'Grid Computing - Making the Global Infrastructure a Reality', John Wiley and Sons Ltd., pp. 65–100.

De Roure, D., N. R. Jennings & N. Shadbolt (2003), The semantic grid: A future e-science infrastructure, *in* F.Berman, G.Fox & A. J. G.Hey, eds, 'Grid Computing - Making the Global Infrastructure a Reality', John Wiley and Sons Ltd, pp. 437–470.

De Roure, D., N. R. Jennings & N. Shadbolt (2005), 'The semantic grid: Past, present and future', *Procedings of the IEEE* **93**(3), 669–681.

DOE - Office of Science (2005), 'DOE - National Collaboratories'.
URL: *http://www.doecollaboratory.org/ (last accessed 25/09/2006)*

Dourish, Paul & Sara Bly (1992), Portholes: Supporting awareness in a distributed work group, *in* 'CHI 1992', ACM, Monterey, CA.

Edge, David (1979), 'Quantitative measures of communication in science: A critical review', *History of Science* **17**, 102–134.

EGEE (2006), 'Enabling grids for e-science'.
URL: *http://www.eu-egee.org/ (last accessed 25/09/2006)*

Egido, Carmen (1988), Videoconferencing as a technology to support group work: a review of its failure, *in* 'CSCW 88', ACM, Portland, Oregon.

eMule (2006), 'emule-project'.
URL: *http://www.emule-project.org/ (last accessed 25/09/2006)*

ESC (2006), 'The earth simulator center'.
URL: *http://www.es.jamstec.go.jp/esc/eng (last accessed 25/09/2006)*

Farnhill, James (2006), 'e-Infrastructure Programme'.
URL: *http://www.jisc.ac.uk/index.cfm?name=programme_einfrastructure (last accessed 25/09/2006)*

Felber, P., E. Biersack, L. Garces-Erice, K. Ross & G. Urvoy-Keller (2004), Data indexing and querying in DHT peer-to-peer networks, *in* 'Proceedings of ICDCS 2004'.

Fish, Robert S., Robert E. Kraut & Robert W. Root (1992), Evaluating video as a technology for informal communication, *in* 'CHI 1992', ACM, Monterey, CA.

Fish, Robert S., Robert Kraut & Barbara L. Chalfonte (1990), The videowindow system in informal communication, *in* 'CSCW 1990', ACM, Los Angeles, California.

Foster, I. (2002), 'The grid: A new infrastructure for 21th century science', *Physics Today* **55**(2), 42–47.

Foster, I. & C. Kesselman (1997), 'Globus: A metacomputing infrastructure toolkit', *International Journal of Supercomputer Applications* **11**(2), 115–128.

Foster, I. & C. Kesselman (1999), Computational grids, *in* I.Foster & C.Kesselman, eds, 'The Grid: Blueprint for a New Computing Infrastructure', Morgan-Kaufman, chapter 2, pp. 15–51.

Foster, I., C. Kesselman, J. Nick & S. Tuecke (2002), *The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration*, Open Grid Service Infrastructure WG, Global Grid Forum.

Foster, I., C. Kesselman & S. Tuecke (2001), 'The anatomy of the grid: Enabling scalable virtual organisations', *International J. Supercomputer Applications* **15(3)**.

Foster, I., H. Kishimoto, A. Savva, D. Berry, A. Djaoui, A. Grimshaw, B. Horn, F. Maciel, F. Siebenlist, R. Subramaniam, J. Treadwell & J. Von Reich (2005), The open grid services architecture, version 1.0, Technical report, Global Grid Forum (GGF).

Foster, I., J. Frey, S. Graham, S. Tuecke, K. Czajkowski, D. Ferguson, F. Leymann, M. Nally, I. Sedukhin, D. Snelling, T. Storey, W. Vambenepe & S. Weerawarana (2004), Modeling stateful resources with web services v. 1.1, Technical report, Computer Associates International, Inc., Fujitsu Limited, Hewlett- Packard Development Company, International Business Machines Corporation, The University of Chicago.

Foster, I., J. Geisler, W. Nickless, W. Smith & S. Tuecke (1997), Software infrastructure for the i-way high performance distributed computing experiment, *in* 'The 5th IEEE Symposium on High Performance Distributed Computing', IEEE, pp. 562–571.

Foster, Ian (2002), 'What is the grid? a three point checklist', *GRID Today* **1**(6).

Fox, Geoffrey & David Walker (2003), e-science gap analyis, Technical report, National e-Science Centre.

Frey, J. G., D. De Roure & L. A. Carr (2002), Publication at source: Scientific communication from a publication web to a data grid, *in* 'Euroweb 2002 Conference - The Web and the GRID: from e-science to e-business', Oxford.

Frey, J. G., M. Bradley, J. W. Essex, M. B. Hursthouse, S. M. Lewis, M. M. Luck, L. Moreau, D. C. De Roure, M. Surridge & A. Welsh (2003), Combinatorial chemistry and the grid, *in* F.Berman, G.Fox & T.Hey, eds, 'Grid Computing — Making the Global Infrastructure a Reality, Wiley Series in Communications Networking and Distributed Systems', John Wiley & Sons Ltd, pp. 945–962.

FusionGRID (2004), 'FusionGRID'.
URL: *http://www.fusiongrid.org/ (last accessed 25/09/2006)*

Gnutella (2001), 'Gnutella'.
URL: *http://www.gnutella.com (last accessed 25/09/2006)*

Goble, C. A., S. Pettifer, R. Stevens & C. Greenhalgh (2003), Knowledge integration: In silico experiments in bioinformatics, *in* I.Foster & C.Kesselman, eds, ‘The Grid: Blueprint for a New Computing Infrastructure’, 2 edn, Morgan-Kaufman.

Gong, Li (2002), Project jxta: A technology overview, Technical report, Sun Microsystems.
URL: *http://www.jxta.org/project/www/docs/jxtaview_01nov02.pdf*

Greenberg, Saul (1991), Computer-supported cooperative work and groupware, *in* S.Greenberg, ed., ‘Computer-supported cooperative work and groupware’, Academic Press, pp. 1–8.

GridPP (2006), ‘Gridpp - uk computing for particle physics’.
URL: *http://www.gridpp.ac.uk/ (last accessed 25/09/2006)*

Groove Networks (2006), ‘Groove virtual office’.
URL: *http://www.groove.net/home/index.cfm (last accessed 25/09/2006)*

Gruber, T. R. (1993), ‘A translation approach to portable ontologies’, *Knowledge Acquisition* **5**(2), 199–220.

GSC-Chinook (2006), ‘Chinook: P2p informatics’.
URL: *http://www.bcgsc.bc.ca/chinook/ (last accessed 30/09/2006)*

Gulli, A. & A. Signorini (2005), The indexable web is more than 11.5 billion pages, *in* ‘The 14th International World Wide Web Conference’, Chiba, Japan.

Hagstrom, Warren O. (1965), *The Scientific Community*, Basic Books, London/New York.

Hendler, James, Tim Berners-Lee & Eric Miller (2002), ‘Integrating applications on the semantic web’, *Journal of the Institute of Electrical Engineers of Japan* **122**(10), 676–680.

Herbsleb, James D., David L. Atkins, David G. Boyer, Mark Handel & Thomas A. Finholt (2002), Introducing instant messaging and chat in the workplace, *in* ‘CHI 2002’, ACM, Minneaplolis, Minnesota.

Hey, Tony & Anne E. Trefethen (2002), ‘The uk e-science core programme and the grid’, *Future Generation Computing Systems* **18**(8), 1017–1031.

Hynes, Robert & R. Anthony Cox (2006), 'Iupac subcommittee for gas kinetic data evaluation'.
URL: *http://www.iupac-kinetic.ch.cam.ac.uk/ (last accessed 25/09/2006)*

Iamnitchi, A. & I. Foster (2005), Interest-aware information dissemination in small-world communities, *in* 'The 14th IEEE International Symposium on High Performance Distributed Computing', North Carolina, USA, pp. 167–175.

Iamnitchi, Adriana, Matei Ripeanu & Ian Foster (2004), Small-world file-sharing communities, *in* 'Infocom', Hong Kong.

Isaacs, Ellen A., Steve Whittaker, David Frohlich & O'Conaill Brid (1997), Informal communication re-examined: New functions for video in supporting opportunistic encounters, *in* K. E.Finn, A. J.Sellen & S. B.Wilbur, eds, 'Video-Mediated Communication', Lawrence Erlbaum, New Jersey, pp. 459–485.

Isaacs, Ellen, Alan Walendowski, Steve Whittaker, Diane J. Schiano & Candace Kamm (2002), The character, functions, and styles of instant messaging in the workplace, *in* 'CSCW2002', ACM, New Orleans, Louisiana.

Isaacs, Ellen & John C. Tang (1993), What video can and can't do for collaboration: A case study, *in* 'Multimedia', ACM Press, Anaheim, CA, pp. 199–205.

Java Community Process (2006), 'Jsr 168: Portlet specification'.
URL: *http://www.jcp.org/en/jsr/detail?id=168 (last accessed 25/09/2006)*

Kan, Gene (2001), Gnutella, *in* A.Oram, ed., 'Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology', O'Reilly, pp. 94–132.

Kaplan, Bonnie & Dennis Duchon (1988), 'Combining qualitative and quantitative methods in information systems research: A case study', *MIS Quarterly* **12**(4), 571.

Katz, J. Sylvan & Ben R. Martin (1997), 'What is research collaboration?', *Research Policy* **26**.

Kazaa (2006), 'Kazaa - search, download & share!'.
URL: *http://www.kazaa.com/ (last accessed 25/09/2006)*

Kouzes, R.T., J.D. Myers & W.A. Wulf (1996), 'Collaboratories: Doing science on the internet', *IEEE Computer* **29**(8), 40–46.

Kraut, Robert, Carmen Egido & Jolene Galegher (1990), Patterns of contact and communication in scientific research collaborations, *in* J.Galegher, R.Kraut & C.Egido, eds, 'Intellectual teamwork: social and technological foundations of cooperative work', Lawrence Erlbaum Associates, Inc, Mahwah, NJ, USA, pp. 149–171.

Kraut, Robert, Jolene Galegher & Carmen Egido (1986), Relationships and tasks in scientific collaborations, *in* 'The 1986 ACM conference on Computer-supported cooperative work', Austin, Texas, pp. 229–245.

Kraut, Robert, Robert S. Fish, Robert W. Root & Barbara L. Chalfonte (1990), Informal communication in organisations: Form, functions, and technology, *in* R.Baecker, ed., 'Readings in Groupware and ComputerSupported Cooperative Work: Assisting human to human collaboration', Morgan Kaufmann Publishers Inc, San Francisco, CA.

Lau, Lydia M. S., Jayne Curson, Richard Dew, Peter M. Dew & Christine Leigh (1999), Use of virtual science park resource rooms to support group work in a learning environment, *in* 'the international ACM SIGGROUP conference on Supporting group work', ACM, Phoenix, Arizona.

Lawrence, A & et al. (2004), Googling secure data, *in* S. J.Cox, ed., 'Proceedings of the U.K. e-science All Hands Meeting'.

LCG (2006), 'Lhc computing grid project'.
URL: *http://lcg.web.cern.ch/LCG/ (last accessed 25/06/2006)*

Lee, Alison, Andreas Girgensohn & Kevin Schlueter (1997), Nynex portholes: Initial user reactions and redesign implications, *in* 'GROUP 97', ACM, New York.

Lee, Sooho & Barry Bozeman (2005), 'The impact of research collaboration on scientific productivity', *Social Studies of Science* **35**(5), 673–702.

Liu, Yan & Ian Gorton (2004), An empirical evaluation of architectural alternatives for j2ee and web services, *in* 'The 11th Asia-Pacific Software Engineering Conference', IEEE Computer Society, Busan, Korea.

Lv, Quin, Pei Cao, Edith Cohen, Kai Li & Scott Shenker (2002), Search and replication in unstructured peer-to-peer networks, *in* 'The 16th ACM International Conference on Supercomputing(ICS'02)', New York.

Mathes, Adam (2004), Folksonomies - cooperative classification and communication through shared metadata, Technical report, Graduate School of Library and Information Science, University of Illinois Urbana-Champaign.
URL: *http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html*

McCool, Rob (2005), 'Rethinking the semantic web, part 1', *IEEE Internet Computing* **9**(6), 88, 86–87.

McCool, Rob (2006), 'Rethinking the semantic web, part 2', *IEEE Internet Computing* **10**(1), 96–95.

McIlraith, S.A., T.C. Son & Honglei Zeng (2001), 'Semantic web services', *Intelligent Systems* **16**(2), 46–53.

Minar, Nelson & Marc Hedlund (2001), A network of peers: Peer-to-peer models through the history of the internet, *in* A.Oram, ed., 'Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology', O'Reilly, pp. 3–20.

Mingers, John (2001), 'Combining is research methods: Towards a pluralist methodology', *Information Systems Research* **12**(3), 240–259.

Muller, Michael J., Mary E. Raven, Sandra Kogan, David R. Millen & Kenneth Carey (2003), Introducing chat into business organizations: Toward an instant messaging maturity model, *in* 'GROUP'03', ACM, Sanibel Island, Florida.

Myers, J. D. & et al. (2004), A collaborative informatics infrastructure for multi-scale science, *in* 'Proceedings of the Challenges of Large Applications in Distributed Environments (CLADE) Workshop', Honolulu, USA.

Myers, J.D., Thomas C. Allison, Sandra Bittner, Brett Didier, Michael Frenklach, William H. Green Jr., Yen-Ling Ho, John Hewson, Wendy Koegler, Carina Lansing, David Leahy, Michael Lee, Renata Mccoy, Michael Minkoff, Sandeep Nijsure, Gregor V. Laszewski, David Montoya, Luwi Oluwole, Carmen Pancerella, Willian Pitz, Larry A. Rahn, Branko Ruscic, Karen Schuchardt, Eric Stephan, A. Wagner, Theresa Windus & Christine Yang (2005), 'A collaborative informatics infrastructure for multi-scale science', *Cluster Computing* **8**, 243–253.

myGrid (2006), 'myGrid'.
URL: *http://www.mygrid.org.uk (last accessed 25/09/2006)*

Nardi, Bonnie A., Steve Whittaker & Erin Bradner (2000), Interaction and outeraction: Instant messaging in action, *in* ‘CSCW2000’, ACM, Philadelphia.

National Research Council (1993), ‘National collaboratories: Applying information technology to scientific research’.

NeSC (2006), ‘National e-Science Centre’.
URL: *http://www.nesc.ac.uk/ (last accessed 25/09/2006)*

Newman, David (2006), ‘Combechem’.
URL: *http://www.combechem.org/ last accessed 01/10/2006*

Newman, M. E. J. (2001*a*), ‘Scientific collaboration networks. i. network construction and fundamental results’, *Physcal Review E* **64**.

Newman, M. E. J. (2001*b*), ‘Scientific collaboration networks. ii. shortest paths, weighed networks, and centrality’, *Physcal Review E* **64**.

Newman, M. E. J. (2001*c*), ‘The structure of scientific collaboration networks’, *Proc. Natl. Acad. Sci.* **98**(2), 404–409.

NGS (2006), ‘Ngs national grid service portal’.
URL: *https://portal.ngs.ac.uk (last accessed 25/06/2006)*

NIST (2005), ‘The nist chemistry webbook’.
URL: *http://webbook.nist.gov/chemistry/ (last accessed 25/09/2006)*

Nunamaker, J.F. Jr. & M. Chen (1990), ‘Systems development in information systems research’, *Proceedings of the Twenty-Third Annual Hawaii International Conference on System Sciences* **3**(2), 631–640.

Obata, Akihiko & Kazuo Sasaki (1998), Officewalker: A virtual visiting system based on proxemics, *in* ‘CSCW 98’, ACM, Seatle, Washington.

Office of Science - U.S. Department of Energy (2002), ‘Report of the high-performance network planning workshop’.

OGCE (2006), ‘Open grid computing environments portal’.
URL: *http://www.collab-ogce.org/ogce2/ (last accessed 25/09/2006)*

O’Reilly, Tim (2001), Remaking the peer-to-peer meme, *in* A.Oram, ed., ‘Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology’, O’Reilly, pp. 38–58.

Papazoglou, M. P. (2003), Service-oriented computing: concepts, characteristics and directions, *in* 'The Fourth International Conference on Web Information Systems Engineering (WISE 2003)', IEEE Computer Society, pp. 3–12.

Papazoglou, M. P. & D. Georgakopoulos (2003), 'Service-oriented computing', *Communication of the ACM* **46**(10), 25–28.

Parameswaran, M., A. Susarla & A.B. Whinston (2001), 'P2p networking: an information sharing alternative', *Computer* **34**(7), 31–38.

Pierce, Marlon E., Choonhan Youn & Geoffrey Fox (2002), The gateway computational web portal: Developing web services for high performance computing, *in* 'International Conference on Computational Science'.

Pilling, M. J., ed. (1997), *Low-temperature combustion and autoignition*, Vol. 35 of *Comprehensive Chemical Kinetics*, Elsevier, Amsterdam.

Pouwelse, J., P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. Epema, M. Reinders, M. van Steen & H. Sips (2006), Tribler: A social-based peer-to-peer system, *in* 'In 5th Intl Workshop on Peer-to-Peer Systems (IPTPS)'.

PPDG (2006*a*), The particle physic data grid: From fabric to physics - final report july 2006, Technical report.
URL: *http://www.ppdg.net/docs/ppdg-final-report-july06.pdf (last accessed 25/09/2006)*

PPDG (2006*b*), 'Particle physics data grid'.
URL: *http://www.ppdg.net/ (last accessed 14/09/2006)*

PrIMe (2006), 'Prime - process informatics model'.
URL: *http://prime.citris-uc.org/ (last accessed 25/09/2006)*

PrIMe (n.d.), 'Process informatics - a new paradigm for building complex chemical reaction models'.
URL: *http://prime.citris-uc.org/filemanager/active?fid=1 (last accessed 31/08/2006)*

Project JXTA (2003), *Project JXTA v2.0: Java Programmer's guide*, Sun Microsystems, Inc.

Quan-Haase, A., J. Cothrel & B. Wellman (2005), 'Instant messaging for collaboration: A case study of a high-tech firm', *Journal of Computer-Mediated Communication* **10**(4).

Ratnasamy, Sylvia, Paul Francis, Mark Handley, Richard Karp & Scott Shenker (2001), A scalable content-addressable network, *in* 'Proceedings of the ACM SIGCOMM 2001', pp. 161–172.

Research Councils UK (2006), 'About the uk e-science programme'.
URL: *http://www.rcuk.ac.uk/escience/ (last accessed 25/09/2006)*

Rickard, Andrew and Pascoe, Stephen (2006), 'The Master Chemical Mechanism'.
URL: *http://mcm.leeds.ac.uk/MCM/ (last accessed 25/09/2006)*

Rowstron, Antony & Peter Drusche (2001), Pastry: Scalable, decentralized object location and routing for large-scale peer-to-peer systems, *in* 'Proc. of the 18th IFIP/ACM International Conference on Distributed Systems Platforms (Middleware 2001)', Heidelberg, Germany.

Sakai (2006), 'Sakai: Collaboration and learning environment for education'.
URL: *http://sakaiproject.org/ (last accessed 25/09/2006)*

Schlosser, M., M. Sintek, S. Decker & W. Nejdl (2002), Hypercup - hypercubes, ontologies and efficient search on p2p networks, *in* 'Intl. Workshop on Agents and Peer-to-Peer Computing', Bologna, Italy.

Schlosser, M., M. Sintekand S. Decker & W. Nejdl (2002), A scalable and ontology-based P2P infrastructure for Semantic Web Services, *in* 'Proceedings of the Second IEEE International Conference on Peer-to-Peer Computing', pp. 104–111.

Schmidt, Cristina & Manish Parashar (2004), 'A peer-to-peer approach to web service discovery', *World Wide Web Journal* **7**.

SETI@HOME (2006), 'What is seti@home?'.
URL: *http://setiathome.berkeley.edu/ (last accessed 25/09/2006)*

Shirky, Clay (2001), Listening to napster, *in* A.Oram, ed., 'Peer-to-Peer: Harnessing the Benefits of a Disruptive Technology', O'Reilly, pp. 21–37.

skype (2006), 'Skype. the whole world can talk for free'.
URL: *http://www.skype.com/ (last accessed 14/09/2006)*

SmartTea (2006), 'The smart tea project'.
URL: *http://www.smarttea.org/ (last accessed 25/09/2006)*

Sollazzo, T, S Handschuh, S Staab & M Frank (2002), Semantic web service architectureevolving web service standards toward the semantic web, *in* 'The 15th International FLAIRS Conference'.

Sonnenreich, Wes (1997), 'A history of search engines'.
URL: *http://www.wiley.com/legacy/compbooks/sonnenreich/history.html (last accessed 07/09/2006)*

Srinivasan, Latha & Jem Treadwell (2005), An overview of service-oriented architecture, web services and grid computing, Technical report, HP Software Global Business Unit.

Sripanidkulchai, Kunwadee, Bruce Maggs & Hui Zhang (2003), Efficient content location using interest-based locality in peer-to-peer systems, *in* 'Infocom', San Francisco, USA.

Stoica, Ion, Robert Morris, David Karger, Frans Kaashoek & Hari Balakrishnan (2001), Chord: A scalable peer-to-peer lookup service for internet applications, *in* 'Proceedings of the ACM SIGCOMM 2001', pp. 149–160.

Tang, John C., Ellen A. Isaacs & Monica Rua (1994), Supporting distributed groups with a montage of lightweight interactions, *in* 'CSCW 94', ACM, Chapel Hill, North Carolina.

Taylor, Ian J., Matthew S. Shields, Ian Wang & Roger Philp (2003), Distributed P2P Computing within Triana: A Galaxy Visualization Test Case., *in* '17th International Parallel and Distributed Processing Symposium (IPDPS 2003)', IEEE Computer Society, pp. 16–27.

The DataGrid Project (2006), 'http://eu-datagrid.web.cern.ch/eu-datagrid/'.

The Globus Alliance (2006), 'The Globus Toolkit'.
URL: *http://www-unix.globus.org/toolkit/ (last accessed 25/09/2006)*

Thompson, Rich (2006), 'Oais web services for remote portlets (wsrp) tc'.
URL: *http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wsrp (last accessed 12/09/2006).*

Tian, Yang, Lydia M S Lau & Peter M Dew (2003), A peer-to-peer knowledge sharing approach for a networked research community, *in* 'Proceedings of the Fifth International Conference on Enterprise Information Systems', Angers, France, pp. 642–645.

Traversat, Bernard, Mohamed Abdelaziz & Eric Pouyoul (2003), *Project JXTA: A Loosely-Consistent DHT Rendezvous Walker*, Sun Microsystems, Inc.

Triana (2003), 'The triana project'.
URL: *http://www.trianacode.org/index.html (last accessed 14/09/2005)*

Tuecke, S., K. Czajkowski, I. Foster, J. Frey, S. Graham, C. Kesselman, T. Maguire, T. Sandholm, P. Vanderbilt & D. Snelling (2003), Open grid services infrastructure (ogsi) version 1.0, Technical report, Global Grid Forum.

UDDI Consortium (2001), Uddi executive white paper, Technical report.

Uram, Tom (2006), 'Accessgrid'.
URL: *http://www.accessgrid.org/ (last accessed 30/09/2006)*

Verma, Kunal, Kaarthik Sivashanmugam, Amit Sheth, Abhijit Patil, Swapna Oundhakar & John Miller (2003), 'METEOR-S WSDI: A scalable p2p infrastructure of registries for semantic publication and discovery of web services', *Journal of Information Technology and Management* .

Vogels, Werner (2003), 'Web services are not distributed objects', *IEEE Internet Computing* **7**(6), 59–66.

VRE (2006), 'Virtual research environments programme'.
URL: *http://www.jisc.ac.uk/index.cfm?name=programme_vre (last accessed 12/09/2006).*

W3C (2004*a*), 'Owl web ontology language for services (owl-s)'.
URL: *http://www.w3.org/Submission/2004/07/, (last accessed 12/09/2006).*

W3C (2004*b*), 'Resource description framework(rdf): Concepts and abstract syntax'.
URL: *http://www.w3.org/TR/rdf-concepts/ (last accessed 12/09/2006).*

W3C (2004*c*), 'Web ontology language'.
URL: *http://www.w3.org/TR/owl-features/ (last accessed 12/09/2006).*

W3C Web Service Architecture Working Group (2004), Web service architecture,http://www.w3.org/tr/ws-arch/, Technical report.

Wainfan, Lynne & Paul K. Davis (2004), Challenges in virtual collaboration, Technical report, National Defense Research Institute.

Wasson, Glenn (2006), 'Wsrf.net'.
URL: *http://www.cs.virginia.edu/g̃sw2c/wsrf.net.html (last accessed 25/09/2006)*

Whittaker, Steve (1995), 'Rethinking video as a technology for interpersonal communication: Theory and design implications', *International Journal of Man-Machine Studies* **42**(5), 501–529.

Woolf, A. & et al. (2004), Enterprise specification of the nerc datagrid, *in* S. J.Cox, ed., 'Proceedings of the U.K. e-science All Hands Meeting'.

Yin, Robert K. (1994), *Case Study Research - Design and Methods*, Vol. 5 of *Applied Social Research Methods Series*, 2nd edn, Sage Publications, Thousand Oaks, London, New Delhi.