Project no. FP6-045520

EchoGRID

European and Chinese Cooperation on Grid

SSA Project

Information Society Technologies

D.2.3 – **Final Consolidated Roadmap**

Due date of deliverable: 30 April 2009

Actual resubmission date: 06 May 2009

Start date of project: 1 January 2007 **Duration**: 24 months

Organisation name of lead contractor for this deliverable: ATOS

Project co-funded by the European Commission within the Sixth Framework Programme (2002-2006)		
Dissemination Level		
PU	Public	PU

Keyword List: roadmap, vision, EU-China, cooperation, Grid standards, research collaboration, strategy, cloud, virtual Organisation, workflow, new programming paradigms

MODIFICATION CONTROL			
Version	Date	Status	Modifications made by
1	24-06-2008	Draft	Thierry Priol - Template
2	20-07-2008	update	Reworked by Engineering – Andrea Manieri
3	22-10 -2008	Update Section Grid Architectures	Ignacio Soler Grid Architectures reworked. Includes Cloud Computing.
4	22-10-2008	Update within comments from version 3	Andrea Manieri
5	23-10-2008	Update from the initiatives on China	Depei
6	27-11-2008	Update of Virtual Organisation section	Ignacio Soler
7	30-11-2008	Component Model section updated	Ignacio Soler , Ji
8	2-12-2008	Update of chapter 6 VO	Gregory (NTUA)
9	2-12-2008	5.2.1. update	CNIC
10	31-12-2008	Grid Management	STFC Erica update.
11	31/12/2008	Final consolidated Roadmap	Ignacio (Atos) Integration of all the reviews and consolidate the final version.
21	15/3/2009	Revision to resubmit to the EC	Updated the sections of 3,5,10
22	30/4/2009	Last Revision	Included the comments from the endorsement in this document to have a single one to be handed.

Deliverable manager

- Ignacio Soler - Atos Origin

List of Contributors Partners

ERCIM	ICCS/NTUA	ATOS	ENG	THALES	BUAA	ICT	CNIC	NUDT	HUW

List of Evaluators

- Michael Wilson, ERCIM/STFC
- Nicola Tonellotto, ERCIM/ISTI-CNR

Summary

The document presents the final roadmap to achieve interoperable Grid systems between Europe and China. The roadmap proposed for the 3, 5, and 10 year stages addressing issues which have been identified as of mutual concern to researchers in both geographical areas: New Programming Paradigms, Grid Architectures, Grid Management, Virtual Organisations, the Component Model, Workflow – Business Processes. Finally recommendations on how to make this roadmap happen are made for EU and Chinese funding and research players.

Acronym	Definition
Server Virtualization	refers to uncoupling server operating systems from hardware hosts, allowing multiple isolated operating system environments to share the same physical server.
Desktop Virtualization	refers to uncoupling a client operating system environment from underlying hardware, allowing end user workspaces to be hosted on servers and accessed remotely, or for corporate workspaces to be isolated from personal workspaces on client machines.
Storage Virtualization	refers to an abstraction layer between hosts and physical storage that provides a single management point for multiple block-level storage devices in a SAN and presents a set of virtual storage volumes for hosts to use.
Application virtualization	refers to the uncoupling of applications from host operating systems, dramatically easing deployment and allowing the virtualized application to run in its own isolated sandbox.
File Virtualization	creates an abstraction layer between end users or applications and physical file servers, creating a single namespace across multiple filers or NAS devices, providing a single view of their directories and files, and giving administrators a single control point for managing their data.
Application grids	refers to the dynamic provisioning of application execution environments throughout a network of host systems.
I/O Virtualization:	refers to an abstraction of the links between physical servers and network storage, allowing applications to be provisioned and re-provisioned without requiring changes to the servers or SAN fabric.
Enterprise/data center virtualization	refers to the use of virtualization technologies and integrated management tools to enable more flexible, automated, even dynamic matching of hardware resources to variable application workloads and changing business requirements.

Table of Contents

1 Introduction

The state-of-the-art background to the current interaction of European and Chinese Grid activities was presented in the first EchoGRID deliverable D1.1 in August 2007, describing EU and Chinese Research Initiatives/Grid strategic orientations. The orientation for the project presented there was to boost interactions among researchers in Europe and China through several mechanisms:

- develop gateways
- develop common components
- test interoperability
- develop common standards
- use common security and certification infrastructure supported by the IGTF
- establish common controlled vocabularies for policies and semantically rich descriptions
- agree common accounting mechanisms
- establish an international governance institution

The endpoint of this orientation is a model for interoperable European & Chinese grids for academic/research and commercial use, built on the generic OGSA architecture, with semantically rich descriptions of services, workflows, registries, quality of service, policies and accounting to move towards the SOKU (Service Oriented Knowledge Utility) vision as proposed by the EU Next Generation Grid (NGG) Expert Group and further adopted by the NESSI service-based economy by 2018.

The current document is a consolidation of the Draft Roadmap (D2.2) towards this vision in two ways:

- By dividing the Grid technological challenges into six specific areas.
- By defining the expected progress along the roadmap at three points, in three, five and 10 years.

The six technical aspects of Grid computing were identified at the first joint European/Chinese technical workshop of the EchoGRID project in Beijing, China held in February 2007. Two workshops were further organized to refine the roadmap and to produce this document: in Rome (January 2008) and in Athens (June 2008). Finally gathering the results from the second EchoGRID conference held in Shenzen (October 2008) and in the workshop in Beijing (October 2008).

These technical challenges were ones where both the Chinese and European experts believe that joint technical development was strongly required to ensure the interoperability of open Grid services:

- New programming paradigms
- Grid architectures

- Grid management
- Virtual Organisations
- The component model
- Workflow – business process

In this report, each of these topics is addressed as a single chapter, covering a detailed analysis of current state of the art and their expected advances at each of the three points on the roadmap, towards the vision of a globally interoperable grid.

2 Rational behind the six selected challenges

The six challenges chosen to be addressed by the roadmap in the project were selected following the first Euro-Chinese EchoGRID workshop in Beijing in February 2007 where an overview of running R&D projects, Industrial and Research initiatives and experiences, and a general situation summary of the market in Europe and China were presented.

The essential difference between grids and other forms of distributed computing is that grids involve the sharing of resources between organizations. There are many forms of outsourcing arrangement whereby computing resources can be recruited from outside an organization for their use, but these apply a one way subcontract relationship. In contrast, grids allow two or more organizations to share each other's resources. The organizational and legal structure used to control this sharing relationship is that of the Virtual Organisation. Since the legal and social institutions in Europe and China potentially vary so much, the future of such VO was an obvious candidate for consideration by the project.

The legal structure of the VO requires support from the Grid infrastructure to manage the sharing of resources, initially through authentication, authorization and accounting, but also through security, time, quality and cost of service delivery. Since these aspects of Grid management are core to the VO management, they were also selected as a topic for the project to address.

Given that the project is considering the different roadmaps for VO and Grid management, the third area selected to address was that of the Grid architecture over which this management would operate.

The architecture is populated with components created by different developers, therefore the component model used to bind these together is the forth area selected to be addressed. Within Web Services there is considerable debate about the technical efficiency of heavyweight approaches using SOAP as opposed to the ease of learning more lightweight approaches such as REST. Similarly, in Europe many users of university campus grids do not use national or international grids (e.g. EGEE) because the gLite software stack they use is too heavyweight and does not offer either fast enough interactive response or ease of maintenance. The Grid management model assumes that authentication, authorization, accounting, security, time, quality and cost of service delivery will all be addressed. However, the components required to address these non-functional requirements may be too heavyweight, and subsets of these components may be appropriate for subsets of applications. Possible roadmaps for component models may vary between Europe and China in the future as the importance of different applications and management models varies.

The applications that users run on grids vary considerably. The simple submission of multiple jobs to batch queues was the first application style to justify large European grids for parameter setting in various research and economic models. In contrast, research facilities require the interactive steering of experiments through the visualization of large parallel simulations, which relies upon real time control of the grid. The third main style of application, and that which is most accessible to commercial use, uses workflow co-ordinated web services. The orchestration or choreography of Web Services hosted by different providers as a workflow, moves away from the shared resource basis of grids to a collaboration between providers who are each responsible for their own high level service. This high level service oriented architecture (SOA) considerably increases the manageability of each service, but puts the responsibility for the Grid management and the support for the virtual organization in the workflow management. If Grid computing is seen as enabling the sharing of distributed computing and data resources such as processing, networking and storage capacity to create a cohesive resource environment for executing distributed applications in e-Science and businesses, then is this co-ordination of high level services going to be viewed as something different by business in Europe, China or both ? The future of both the technologies themselves, and the brand management could move in many directions in coming years. Consequently, workflow management becomes the fifth area to be considered in the roadmap.

Having agreed five high level topics to consider in the joint road mapping activities, there are a great many low level issues which, are research topics addressed in recent Grid conferences, and were also considered for joint work inside the project, including:

- Grid applications, including e-Science and e-Business Applications
- Problem Solving Environments
- Internet-based Computing Models
- Collaborative Engineering Environments
- Autonomic and Utility Computing on Global Grids
- Cluster and Grid Integration
- Grid vs Cloud computing models
- Distributed and Large-Scale Data Access and Management
- Grid Economy and Business Models
- Information Services
- Metadata, Ontologies, and Provenance
- Middleware and Toolkits
- Monitoring, Management and Organization Tools
- Networking and Security
- Performance Measurement and Modelling
- QoS and SLA Negotiation
- Resource Management, Scheduling, and Runtime Environments
- Scientific, Industrial and Social Implications

There are risks of both spreading the project’s resources too thinly and of producing a roadmap which was dominated by local low level issues if all of these topics were addressed. However, if only the five high level topics identified so far are addressed, the roadmap will lack any fundamental, long term, research topics. Consequently, it was agreed to include one further topic which was both fundamental in that it is a basic topic in computer science, but also fundamental to the essence of grids. It was decided to include new programming paradigms that arise from the shared, distributed computing environments that grids provide.

Following this rationale, the six technical challenges addressed in the roadmap are:

- New programming paradigms
- Grid architectures
- Grid management
- Virtual Organisations
- The component model
- Workflow – business process

Both the Chinese and European experts involved in the first workshop, and the project management, believe that joint technical development in these six areas will be required to ensure the interoperability of open Grid services and produce the benefits that grids can provide to both Europe and China.

3 NEW PROGRAMMING PARADIGMS

This chapter argues that there is a clear need for new programming paradigms for Grid infrastructures and to a larger extent service infrastructures. Current programming practices are based on low-level abstractions inherited and sometimes extended from existing parallel or distributed systems. They are too tightly coupled to the underlying architectures and thus are not able to cope with the dynamicity, the heterogeneity and the presence of failures, which are the key properties of these infrastructures. We describe three approaches that seem promising and that are currently being investigated in China and Europe.

3.1 Vision

Over the last ten years, Grid computing has seen tremendous developments of research to build effective Grid infrastructure and to a larger extent service infrastructures. Despite all these efforts to design effective and usable computing infrastructures, there is still a concern about how we can program such infrastructures with a higher level of abstraction than actual programming models do. To some extent, programming grids nowadays is very similar with what computer programmers faced before programming languages came to light. Taking into consideration the increasing complexity of existing Grid or service infrastructures, it is of prime importance to study novel programming models that can both hide the complexity of the underlying computing infrastructure and provide a certain level of abstraction to enhance programmers' productivity. This has been already pointed out by Ian Foster and Karl Kesselman in their second edition of their book "The Grid 2: Blueprint for a New Computing Infrastructure":

Grid environments will require a rethinking of existing programming models and, most likely, new thinking about novel models more suitable

Programming the Grid means that a suitable way to express the coordination of computations among a set of distributed resources has to be provided. As the service-oriented paradigm has been adopted by the Grid research community, programming the Grid can be seen as specifying the orchestration of services. For a given problem or applications, specification of the orchestration should be done using high-level abstractions having the following properties: intuitive so that non-expert programmers can use them, generic to handle a large spectrum of applications and parallelism should be implicit and fully hidden to the programmers. Moreover, to cope with the large scale and unreliable dimension of the grid, programming languages, providing these high-level abstractions, have to be associated with a distributed execution model to avoid any bottleneck.

3.2 State of the Art & Limitations

3.2.1 State of the Art

Current Grid programming paradigms came from existing programming practices that have been developed in the context of parallel and distributed systems. Naturally, communication plays an important role within Grid and is thus reflected in the available programming paradigms. Three main communication paradigms have influenced the way Grids are programmed: shared-memory, message-passing and remote Procedure/Method call Furthermore, the design of programming paradigms depends on how a Grid infrastructure is seen by the programmer: a distributed system, a parallel systems or both. Pioneering works, such as around Linda, have shown that shared-memory programming paradigms can simplify greatly the programming by decoupling the software entities that produce a data (stored into the shared data space) and the ones that consume the data (read from the shared space). This idea has lead to the development of several technologies such as JavaSpaces from Sun or TSpaces from IBM. For most Grids aiming at providing computation capability (i.e. parallel system), message-passing programming paradigms, such as MPI, is the de-facto approach. From the perspective of an MPI programmer, the Grid is a single computer, and it is a specific computer according to the individual application.

When the programmer sees the Grid as a distributed system, it can uses existing approaches such as those based on remote procedure calls, remote method invocations (CORBA/Java) or services. Most programming models available today are based on Java, using RMI, sockets as the basic communication infrastructure; they provide features like multithreading, group communication, global synchronization, and so on.

To increase the level of abstraction, thus facilitating Grid programming, skeleton-based programming approaches can hide much of the complexity of the underlying Grid infrastructure, by providing a set of a standard templates that are required by distributed applications (such as master/slave, map-reduce, …)

As to the interaction between programming components, flow-based language and approaches are used to indicate the data and control flow at run-time. Grid workflow is used in many e-Science environments. This paradigm enables the orchestration of Grid services via GUI or XML-based standards that are used for the description of service infrastructures.

Debugging tools in general have a very long history. The three basic forms of debugging, trace, dump, and break date back to the EDSAC computer of the 1940's. In a survey of on-line debugging technique, Evans and Darley remark that computer use is becoming debugging-limited rather than limited by memory size or processor speed. They further predict that this will be the state of affairs until methods for proving that programs have certain properties are successfully developed and come into wide use. Source level debugging mechanisms, such as assertions and breakpoints, allow programmers to inspect the program states when the program control reaches specific code locations. These techniques let the user conveniently observe and monitor dynamic behaviors during debugging a program. The relevant research results for debugging Grid applications can be divided into four broad categories: extension mechanisms of source level debugger, low-level system virtualization, mirror-based technique, manual instrumentation. Here we discuss some instances of them. (1) Extension mechanisms. Kurniawan and Abramson proposed a WSRF-Compliant Debugger for Grid Applications. One important component in it is the middleware compatibility layer. It is written according to an extension mechanism of the middleware, which acts as a translation layer between Grid debugging service and generic debug interface. This extension allows a GDB/GDBServer to be utilized as the back-end debug engine. NASA Ames's researchers build a debugger for applications running on heterogeneous computational Grids, called p2d2, which uses the client-server architecture to isolate the platform-dependent code in a debugger server. The debugger server transforms C++ objects to GDB commands that control a remote GDB server and does the GDB response back to the object level. Generally, these extension mechanisms depend on some conventional debugging facilities. Moreover, they focus on Grid applications but on the Grid Services. (2) Virtualization. Researchers in Computer Laboratory of University of Cambridge propose the pervasive debugging approach for debugging distributed systems, called PDB. Based on the Xen Virtual Machine Monitor to virtualize the system resources of a single machine, PDB can eliminate the probe effect, and can reproduce the exact behavior of a distributed system. By using Xen to virtualize Grid resources, one can deterministically debug Grid applications. However, this debugging technique is difficult to attain for applications that need to be run on a large-scale distributed system. (3) Mirror-based technique. Designers and developers in Sun Microsystems propose the Java Platform Debugger Architecture (JPDA) based on mirror-based systems. As interfaces of building Java debugging tools, the mirror-based API in a Java virtual machine is Java Debugger Interface (JDI), which is the uppermost layer of JPDA. The JDI defines interfaces that describe all program entities that might be of interest to a Java debugger, including classes, interfaces, objects, stack frames and more. All these interfaces are subtypes of the interface com.sun.jdi.Mirror. A mirror is always associated with a particular virtual machine as a whole, in which the entities being mirrored exists, so the implementation of a virtual machine often requires command line options to load the debugging agent or infrastructure at first. For debugging Grid or Web Services, more elaborate facilities have appeared with some systems, but the source level debugger still mainly provides only technique for accessing the dynamic state of an entire container. Therefore, programmers are continuing to debug their service applications in their original or demo machines, but a testbed even or multiple servers in different geographical locations. (4) Manual instrumentation. Often software programmers trace the execution of their programs by manually inserting instrumentation statements that may print out code locations and values of one or more variables. Thus a program can generate its own trace that can be inspected to understand or verify its behavior as it executes on the actual environment or platform. One approach to manually instrumentation at source code is those inserted statements to dump logging information, e.g., logging services, and some programming languages support program annotation, e.g., Java and CLR. Code annotations constitute a powerful mechanism that enables passing information between programmers, tools, and the runtime, from the source code level up to the execution time. Program annotations enrich the program semantics and facilitate optimizations. They describe method usage, convey optimization hints, or aid in code development and maintenance. Manual instrumentation and annotation to measure a program should be done when one gathers profiling information by inserting code to track various quantities. However, the presence of such code can often influence the execution speed and introduce imprecision in the measurement. As with any addition to the program, the instrumentation code or collected data may inadvertently break the program or cause it to no longer fit on a run-time environment. In fact, manual instrumentation is intrusive, passive, imperative and labor-intensive.

3.2.2 Limitations

There are some limitations for current Grid programming paradigms and practices. When using MPI to program Grids, it exposes too many low level architectural details to the application developers, so that programmers must deal with resources allocation, load balancing, dynamicity, fault tolerance, and so on. This makes computational grids quite limited for use by the traditional parallel application developers. More generally, in current Grid programming models, physical resources are too much exposed to the programmers, in terms of number of resources and resources characteristics. It is difficult to deal with dynamically changing large scale Grid environment using the static information of Grid resources. Programmers must handle really low-level technical details to make their applications highly efficient, adaptable to the dynamic and complex context of Grids. For the feature of low-level abstraction, these paradigms are difficult to learn, they are restricted into these well-trained Grid programmers. Also for this reason, the reusability of current models is low. Existing Grid programming models lack of scalability in term of dynamicity. They lack autonomic adaptation to the changing environments. Deterministic execution is insufficient to support the development of complex systems. Concerning the specification of workflow, most of the today approaches are based on XML which has not be designed to be exposed to the programmers or using graphical tools that do not scale well when several thousands of activities have to be coordinated.

3.3 New Ideas

In the area of programming, there are many unconventional approaches being currently investigated in the context of Grid: such as Chemical Programming, Autonomic Element Programming and Pipe-based Programming. We review briefly these approaches in the next paragraphs.

Source: http://www.ercim.org/publication/Ercim_News/enw64/banatre.html
Chemical programming gets its inspiration from the chemical metaphor, formally represented here by a chemical language, such as HOCL that stands for Higher-Order Chemical Language. A chemical program can be seen as a (symbolic) chemical solution where data is represented by floating molecules and computation by chemical reactions between them. When some molecules match and fulfill a reaction condition, they are replaced by the body of the reaction. That process goes on until an inert solution is reached: the solution is said to be inert when no reaction can occur anymore. Formally, a chemical solution is represented by a multi-set and reaction rules specify multi-set rewritings. In HOCL, computation is seen as reactions between molecules in a chemical solution. HOCL is higher-order: reaction rules are molecules that can be manipulated like any other molecules, i.e., HOCL programs can manipulate other HOCL programs. Reactions only occur locally between few molecules that are chosen non-deterministically. The execution can be seen as chaotic (non-deterministic) and possesses nice autonomic properties. It is thus a good candidate to express the orchestration of computation or services within a grid. Moreover, it

provides a high-level of abstraction since the chemical rules are very closed to the problem specification.

Autonomic element programming: Most of the resources sharing activities are distributed, parallel, heterogeneous, dynamic, and most important, managed by different parties autonomously. For any Grid programmer, he/she has to write the codes which manipulate the resources owned by autonomic entities. Being the fundamental building block of autonomic systems, autonomic element is a natural programming concept for modeling and encapsulating the diversity relating to the local resource sharing activities. Autonomic element programming is very similar to the agent-oriented programming paradigm in the heart of the programming model, but it targets more on providing dynamic modeling capability to support data resource aggregation tasks with a property of best-effort. These kinds of tasks are becoming pervasive in the field of information fusion and knowledge engineering, including search engine, knowledge management, and many other Web 2.0 applications. With the evolvements of Grid applications and its marriage with the de facto Web 2.0 applications, we believe that the requirement of autonomic modeling of heterogeneous resources which may support dynamic service orchestration more naturally will become more and more pervasive. Modeling Grid resources from an autonomous element's perspective is likely to become one of the most widely used programming paradigms.

Pipe-based programming: GSML is proposed by ICT, CAS to simplify the developers' efforts of building Grid applications via the techniques of resource virtualization and abstraction, loosely coupled component architecture and high-level reusability. Funnel is GSML component model. Funnel means pipe, it is a software module that conceals low-level details of resources and exposes to the developers only event-based interfaces: input events and output events. Funnels are made upon state-of-art middleware, toolkit and even resource APIs, and can be used to interact with Grid middleware, Web Service, WWW resources, API resources and legacy applications as well. In a Grid resource space, run-time funnels represent a set of available virtualized and abstracted resources usable by applications, which called *funnel library*. A *funnel* may be referred by a URI globally in a given resource space, and represents a *virtual resource* which may be served by more than one real *physical resource* together. The mappings between virtual resources and physical resources, also called *resource binding*, are performed automatically and transparently at runtime for load-balance and fault-tolerance.
Funnels in resource space are connected to form Grid applications according to users' application-level requirements. A Grid application is specified as an XML document which defines which funnels and how they are connected. Funnels are defined and referenced by URI which implies the usage of a certain virtual resource. The connections between funnels are specified by how the output events from source funnels are disseminated and transformed to the input events of target funnels. The loosely coupled component architecture in GSML is event-based by nature: applications register the events that interest them, trigger developers' handlers when feed by the output events, and possibly perform some transformations and send out input events to funnels. The connections and transformation between funnels are made transparent to funnels: funnels neither know what destinations to send output events nor from what sources to receive input events. Therefore funnels may be developed and deployment independently, but are reused many

times on demands of applications. When application-level requirement are altered, the changes can be adapted by the flexibility of event-based connections of funnels: no need to edit, compile and link the application when changing the high-level logic of Grid applications in XML document. The new paradigm also encourages building a large Grid application by connecting in event-based fashion some small parts, which are themselves funnels and built by smaller parts so on. The divide-and-conquer solution of complexity in Grid applications is enabled by the high-level reusability of GSML.

Apart from these aforementioned programming paradigms, aspect-oriented programming, skeleton programming, meta-programming or high-order languages could be explored in the context of Grid programming.

Service-Oriented Computing is based around the concept of message interactions. The mechanism of message processing always provides a framework to support service's behaviours, e.g., roles of a node, syntax and semantics of a single computational unit, exchange patterns, reliability, security, correlation, routes, and service providers. Service debugging can be organized in behaviours of message processing. Detecting and localizing bugs typically require to inspect dynamic behaviors and related contexts during the problem execution, and to check consistency between the observed behaviours and the expected behaviours. We call it as a message-based mode for service debugging. Message-based mode allows programmers debugging one or a sequence of services in one or multiple un-debugged containers. By a transformation unit, a programmer could also interactively inspect a single service's behaviour in source level mode within a new debugged process. Therefore, one can focus on a truly errant behaviour or a service provider without needing to debug the entire container in source level. With a complementary mechanism one could use new modes to debug Grid Services. (1) Message-based mode. It will enable programmers interactively debugging one or a sequence of services in an un-debugged container. In this mode, debugged services in the container can be examined separately. Debugging facilities could co-exist with debugged services in the same container. Therefore, it supports to debug different services in one or multiple containers. (2) Composite mode. It could be classified into two types: mixed debugging and layered debugging. a) Mixed debugging. It is assumed that a program in the client side, an original caller, is easy to start upon with conventional debugging. Therefore, a mixed mode is that the caller can be debugged in source level, and the remote service or callee in the server side can be inspected in a message-based mode. That is, different modes can be separately used for the caller and its callee. b) Layered debugging. With message-based debugging, one could also isolate a single behaviour of the callee in source level, where a new debugged process either local or remote can start up for the debugged behaviour. A source level debugger could connect or attach it, and then debugging activities can be performed. The complementary mechanism provides following capabilities for multi-user to simultaneously debug their services at one or multiple containers. Two-layer Debugging. By debugging stratification, the mechanism separates debugging operations to two layers with different specifications and communicational protocols. The message-based is built on the standards and protocols of Service-Oriented Architecture. It provides the unique capabilities to support service debugging in an un-debugged container. The source level debugging can be used to inspect programs for both the original caller and an isolated service behaviour. It uses breakpoints to enable one to understand and control a program, including pausing and stepping through execution and inspecting and modifying data values. For multiple programmers. By the self-identifiers, multiple programmers starting with client programs on different machines can simultaneously debug remote services at one un-debugged container. For each debugged service, a unique identifier marks the exchanged messages of debugging information of debugged services. For multiple containers. By the co-existence, one can dynamically deploy or publish message-based debugging back-end in multiple un-debugged containers. With different addresses of System Debugging Services and unique identifiers, debugging activities can manipulate a group of debugged services at these containers.

3.4 Make it happen

To satisfy the requirements of application, we think the following two programming paradigms should be considered in the 3, 5 or 10 years. A Data-driven Programming Paradigm Grid computing programming paradigms are deeply influenced by the traditional programming model such as CORBA, EJB, and Web Service etc. But almost all of these programming paradigms are RPC-style, it's reasonable in local computer or local network environment which can satisfy the features such as high bandwidth, low latency etc. Grid computing target at Internet-scale application with the features such as high latency, low bandwidth and so on, in the meanwhile, the rapid growth in size of data make it become more and more difficult to transfer the necessary data to the execution context which cause many problems in many scenarios such as HEP application etc. To solve this problem, there are many rising distributed computing model, like map-reduce. MapReduce can be treated as a variant of traditional Master/Worker mode. But the main difference is that it provides the ability to split a large data set into thousands of small fractions and process them locally in parallel.

In our opinion, a data-driven programming paradigm is needed to process the huge data set distributed across the Internet to complement the control-driven programming paradigm which are widely adopted by EGEE, CNGrid and many other grid infrastructures and it is proven to be not efficient in some scenarios. Platform-as-a-Service/Infrastructure-as-a-Service SaaS is short for Software-as-a-Service. It's an effective mode that provides softwares through the Internet. In this mode, end users don't buy softwares directly from vendors, instead vendors adopt web-accessed mode to provide their softwares. In grid computing, grid infrastructure can be treated as a super computing and how end users can effectively use the infrastructure is a big issues.

At present, grid infrastructures mainly provide two different functionalities: batch job execution and data storage; end users utilizes these functionalities through web service or other internal interfaces. SOA is the most widely adopted manner. But it greatly limits how end users can fully utilize the computing and storage resources provided by underlying grid infrastructures. AWS (Amazon Web Service) and GAE (Google Application Engine) are two successfully examples that expose more controls and functionalities to end users through more flexible manner. 1. AWS consists of Amazon EC2, Amazon S3, Amazon SMS, Amazon SimpleDB etc, which expose the computing, storage resource of Amazon e-infrastructure as a service. End user can package their applications deployment as a virtual machine image including OS, database, applications etc and launch an Amazon EC2 instance running on

Amazon's e-infrastructure and using the huge data storage through Amazon S3. 2. GAE adopts the similar way of AWS, but it has more restrictions which require end user develop according to a predefined component model which only includes applications. In AWS and GAE, the resource usage is billed based on accounting information. AWS can be treated as Platform-as-a-Service and GAE can be treated as Infrastructure-as-a-Service. Both of them have demonstrated successfully how to open underlying infrastructures and give end users more freedom to effectively utilize them.

3.4.1 Three years roadmap

A particular emphasis has to be made to explore other unconventional programming paradigms and to identify those that can be adapted to the context of Grid. Among them, one can cite amorphous computing, bio-inspired computing, autonomic computing and generative computing. These models have to be compared between each other to assess their relevance to the programming of Grids.

3.4.2 Five years roadmap

Most of the research activities dealing with unconventional programming are mostly theoretical and few implementations can show their effectiveness in the programming of large scale distributed systems such as Grids. Moreover, their high-level of abstractions often come with a poor efficiency when implementing them. It will be thus necessary to experiment these models and provide effective implementations assessed with real applications and not just toy problems.

3.4.3 Ten years roadmap

We mainly introduced these new programming paradigms to design applications for Grids. However, they could also be relevant to design Grid middleware systems as well. Many of these unconventional programming paradigms have nice properties to model autonomic and self-* systems. It can be foreseen that the use of such models can be valuable to design and to implement more reliable systems that can repair and adapt themselves to the dynamic configuration of Grid systems.

3.5 How it is covered in existing EC and Chinese R&D programmes

It is widely accepted that the programming paradigm is one of the most important facts for a new concept to be widely used and adopted. Meanwhile, it is also one of the most challenging and risky topics in software engineering, because it has so many facets, technical but also non technical, and requires efforts from both academy and industry.

3.5.1 EU Work Program Initiative Status

During the last decades, there have been many funding projects from the EC on the new/unconventional programming paradigms especially by the Future Emergent Technology unit and its Global Computing Initiative. Research efforts were presented at the workshop 'Unconventional Programming Paradigm 2004', supported by the European Commission Information Society Technologies Program, Future and Emerging Technologies Activity. We can also mention the CoreGRID network of excellence that has some research activities aiming at using chemical programming for workflow enactment. One can also cite the BIONETS project funded under the FP6 Situated and Autonomic Communications (SAC) initiative - Part of "Communication Paradigms for 2020" Proactive Initiative within FET. The goal of this initiative is to promote research in the area of new paradigms for communication/networking systems that can be characterised as situated (i.e. reacting locally on environment and context changes), autonomously controlled, self-organising, radically distributed, technology independent and scale-free. The BIONETS project developed the Fraglets model to program active network. Fraglets takes its root from the chemical programming paradigm. There is another EU-funded project called Interlink that covers Software intensive systems and new computing paradigms. This project is exploring the use of several new paradigms to program future computing systems such as membrane-computing or DNA-based computing.

The GRIDComp project main goal is the design and implementation of a component based framework suitable to support the development of efficient grid applications. The framework implements the "invisible grid" concept: abstract away grid related implementation details (hardware, OS, authorization and security, load, failure, etc.) that usually require high programming efforts to be dealt with. Therefore, GridCOMP makes possible to seamlessly compose applications and services deployed on large scale infrastructures, e.g. several thousand machines all over the world. The GridCOMP project bridges the gap between cutting edge research and industrial applications. Through the collaborative experience between academic and industrial partners, GridCOMP has forged strong collaborations thus producing sustainable results which will still be exploited beyond the project lifetime.

3.5.2 China Work Program Initiative Status

In China, most of the research work programs including Natural Science Foundation of China (NSFC), National High-tech R&D Program (863), National Basic Research Program of China (973 programme), have support for research activities into new programming paradigm for Grid computing. Most leading projects, such as CNGrid GOS (supported by the 863 programme), Crown Grid middleware (supported by NSFC and 863), and internet-based Virtual Computing Environment (supported by 973 programme) have sub-projects on this topic. For example, the CNGrid GOS team has developed a Grid mark-up language called GSML and associated execution environment to address the programming paradigm issue.

For Grid Services, we agree with the evaluation of Evens and Darley remarks, debugging-limited rather than limited by memory size or processor speed, and further envision that service debugging tools will become a staple in Service-Oriented Computing laboratories. Meanwhile, debugging is one of the most important topics in software engineering, although service debugging has received less attention. In China, most of the research work programs including Natural Science Foundation of China (NSFC), National High-tech R&D Program (863), National Basic Research Program of China (973), have supports on the research activities of grid service debugging. Leading projects have sub-project on this topic.

3.6 Recommendations for EC and China

New programming paradigms are seen as long-term research on both sides (Europe and China). The main recommendation is to put these research activities in the mainstream of R&D projects. More experiments with real applications and mature implementation of unconventional programming paradigms should be encouraged. The two research communities: the one addressing unconventional programming and the Grid community should participate to some joint projects to put these novel approaches into practice. Although we think that these new programming paradigms have a good potential to address the existing limitations, they have to be experimented with some industrial test cases.

The whole area of Biomedical sciences due to their sheer complexity (besides biological imaging which is already technologically ripe even regarding to Grid projects) is extremely amenable for Grid implementations.

New debugging techniques are seen as long-term research. The main recommendation is to put these research activities in the mainstream of R&D projects. Firstly, we are considering a clear definition of Grid Service Debugging Architecture on the functionalities and debugging facilities should be proposed. This may involve requirements, standards and interfaces for service debugging. Secondly, more experiments with real applications and mature implementation of service debugging should be encouraged. Although we think that new debugging techniques have a good potential to address the existing limitations, they have to be experimented with some Grid software systems. Thirdly, we plan to integrate a complementary debugger into Grid-Programming Environment for service debugging in large-scale distributed systems.

Develop and design a common integrated programming environment which can provide grid programming interfaces and design patterns well. A wealth of programming examples and applications are also needed. For typical applications in some area such as astronomical calculation or water resources application, it would be easy to use. Grid version of the software development may be considered.

3.7 References to the elements that lead us to the conclusions and roadmap

Website of ECHOGRID contributions from chinesse partners.

http://echogrid.ercim.org/component/option,com_contribution/task,view/

4 GRID ARCHITECTURES

This chapter presents a vision of the actual Grid Architectures, as well as the state of the art and limitations that we have to face during the next years, starting up a smooth transition from the classical Grid Architectures, to the brand new concept called Cloud Computing, passing through what is called Software as a Service (SAAS) or Product as a Service (PAAS).

We present a taxonomy of Cloud Computing and several predictions of the near future on GRIDs as well as a comparisonof what was Grid and what is Cloud. At the end of the chapter we can see the industrial perspective and endorsement of Cloud and the Infrastructure Maturity Model of the IT& Business Integration.

4.1 Vision

It is commonly accepted that the short-term architectural vision for Grids must include more successful paradigms from the industry such as Service-Oriented and Web-Service Architectures or even including Cloud Computing (SOA & WSA & CC). In the short-run this will greatly reduce the interoperability issues between the various Grid designs and implementations assisting in the collaboration and convergence of the Chinese and European infrastructures.

These infrastructures will support the virtualization of data and networks, removing complexity from the application layers of the Grids. Virtualization can lead to the support of flexible business models and application domains, thus increasing compatibility between existing and -perhaps- independently developed middleware.

However, it is expected that in the long-run grids will eventually move away from the traditional Web1.0’s fixed Provider/Consumer architecture. Moreover, extending the Web 2.0 principles in the Grid area, we can model various resources as equal entities acting not only as providers but also as consumers, whose actions are driven by on-demand aggregation and autonomic collaboration based on their own interests. This revolutionary approach will be driven by an effort to develop a Grid that will be seamless to the application end-user, hiding the complexities of the underlying mechanisms. The architectures in the long-run will manage to support a wider audience of end-users, providing them the tools to develop, provide and consume services concluding to an actual democratization of grids and through that, resolving numerous technological issues.

In addition to the Web 2.0 , there are also recently emerging initiatives about Internet of Things and the Future Internet, that are stating the basis of a new relationship between the virtual actors and the physical ones.

4.2 State of the Art & Limitations

In order to reach the short and long term visions, the two communities can build on some progress that has already taken place in some selected works that are already leading the way to the realization of the visions, especially the short-term.

4.2.1 State of the Art

As it is widely accepted that the short-term vision has already started to be applied, the main work taking place in China and Europe is a mix of Grid and SOA infrastructures and Cloud Computing. The developed frameworks comprise the state of the art in the architecture area in Grids. This is witnessed in a number of projects for the development of Grid frameworks such as XServices, CNGRID, CROWN and UNICORE, GRIA and NextGRID ISTs.

The basic Grid architectures for China consist of CNGRID, CROWN and XServices. China has funded a series of Grid projects to establish a Grid infrastructure in recent ten years. CNGrid and CROWN are two well-known projects for building Grid testbeds, which integrate high-performance computers with a new generation of information infrastructure and facilitate scientific research in different disciplines. The Grid architectures inside testbeds are both based on Service-Oriented Architecture and OGSA, while compliant to many Web Services standards of W3C, OASIS and WS-I. XServices seems to be an interesting case, as it is largely based on a web service computing principles. XServices is a software suite and an effective way to implement SOA, which consists of Web Services Runtime & Application Server, Web Services Workflow Designer & Engine, UDDI Server, Web Services-based Information Portal and several useful tools to facilitate service development, deployment & management, and to provide QoS & security support. XServices, only based on JDK, is open source and fully implements WSRF & WSN to support resource management and state notification.

Respectfully, the main representatives of Grid architectures for Europe in the context of this roadmap, seem to be GRIA, UNICORE and NextGRID. The UNICORE 6.0 is a WSRF based implementation of the UNICORE Grid middleware. It forms a software stack that implements an extensible service-oriented architecture compliant to current Web services standards. Similarly, GRIA is a service-oriented infrastructure designed to support B2B collaborations through service provision across organizational boundaries in a secure, interoperable and flexible manner. Again, it is based on a number of Web services standards. Finally, NextGRID is working towards the direction that OGSA has indicated, extending the standards and formulizing a prototype Grid architecture, targeting mainly in the economic viability of Grids.

Of course, important variations of the abovementioned architectures have appeared that worth being used for building towards the vision. Such are the SIMDAT, TrustCoM, Brein and XtreemeOS ISTs. SIMDAT uses the GRIA framework and extends it with dynamic properties deriving from NextGRID in order to support a number of business applications. TrustCoM, focuses on frameworks where trust plays a central role. Brein is dealing with the business perspective of the relationships between the actors in a SOA environment. Finally, XtreemeOS takes a different route by attempting a vertical virtualization of Grid layers. All these, along with a few more interesting ISTs can significantly contribute in the democratization of Grids, which seems to be the long-term vision:

- EzWEB project is addressing the defining a reference architecture and implementation of an open platform supporting the retrieval, combination and utilization of front-end layer components in a next- generation, global Service-Oriented Architecture;

- MASTER project is addressing the provision of methodologies and infrastructures that facilitate the monitoring, enforcement, and audit of quantifiable indicators on the security of a business process, and that provide manageable assurance of the security levels, trust levels and regulatory compliance of highly dynamic service oriented architectures in centralised, distributed (multi-domain), and outsourcing contexts;

- SLA@SOI project is addressing the definition of a holistic approach for the management of service level agreements (SLAs) and to implement an SLA management framework that can be easily integrated into a service-oriented infrastructer (SOI);
- SOA4ALL project is addressing the definition and implementation of a comprehensive framework and infrastructure that integrates four complimentary and revolutionary technical advances (Web principles, Web 2.0, Semantic Web, and Context management) into a coherent and domain independent service delivery platform.

From the Classical Grid Computing to Cloud computing smooth transition:

The state of the art in one year (since last draft of the roadmap) have slightly changed to what we call Cloud Computing, even the term is losing sense, and start stepping to the Cloud vision. In order to align the Roadmap to the new trends, we believe that have a brief definition of Cloud computing is compulsory in this section:

Source: Reservoir White Paper

Cloud computing is just the latest incarnation of a concept that has been around since the 1960's [7][8] - the emergence of a general-purpose public computing utility. Throughout the story of computing we have seen such utilities appear in one form or another and even though some success stories exist, in particular in the area of high performance scientific computing, where Grid computing [9][10][11] made significant progress over the past decade, none of these attempts materialized into a general purpose compute utility that is accessible by anyone, at any time, from anywhere.

A comparative made by the Israeli Association of Grid Technologies (IGT) is provided as a proof of concept of the evolution of the Grid Architectures to the so-called Cloud Computing.

Topic Computing	Classic Grid	Cloud computing
The Problem	Computation over large data sets, or of parallelizable compute-intensive applications. Problem areas are often in pure research or in compute-intensive commercial R&D (e.g. in chip development)	On-demand scalability for all applications, including research, development and business applications.
Main Target Market	First - Academia Second – certain industries	Industry and academia
Business Models – Where the money comes from?	Sponsor-based (Mainly government money) to enable the research community to achieve its research goals in reasonable time. Large corporations may implement it internally for cost saving. Users pay by contributing resources to the shared pool. Otherwise the usage is free (to those allowed to use the grid).	Hosted by commercial companies, paid-for by users. Based on the economies of scale and expertise. Only pay for what you need, when you need it: (On- Demand + Pay per Use).
The Customer	A member of a defined Grid community, or a function	Open to anyone with a credit card - anyone who can pay

	within a large company	
User motivation	Low cost for large computations and processing lf of large amounts of data	- On demand scalability - Lower IT costs (infrastructure operations, energy, personnel)
The capability offered	Access to computers	Access to VMs, services or applications
Unit of work	Grid computing application: A batch job, or a group of parallel batch jobs with a storage service	Either - A dedicated VM instance assigned to the user, on which anything can be run. - A Storage Service - usage of an online application
Current capacity allocation limits per customer request	10,000s of machines	10s of machines
Interface	-Proprietary middleware interface (Not simple) - gLite, Globus - Condor, Unicore	- Simple - Web Services (Industry standard) - Proprietary API
Storage Services	Mainly to enable the sharing of the Grid applications data	- General purpose storage services: including backups, media/software distribution, data sharing, on-line storage, applications storage
Continuous Availability -24x7	Nice to have	Must
Security	X509, single sign on, Grid Security Infrastructure (GSI)	x509, single sign on
Communication infrastructure	Internet based	Internet based, Pay per Use for network bandwidth
Administration	Distributed, Virtual organizations	Centralized (most of the times)
Examples of implementations	EGEE, Intel NetBatch, Condor, LSF, MICRON (CONDOR), UNICORE, GLOBUS	Amazon EC2, Google Apps, Joyent, Flexiscale, GoGrid, IBM BlueCloud, SUN network.com, Microsoft (coming soon)
Key Challenges	Interoperability, SLA, Security, Standards, complexity, competition with commercial clouds	(The list is longer due to broader applicability): The usual IT challenges: Security, Automation, SLA, Cost/Performance, Clouds Interoperability, universal application access, Data Protection

		& Recovery, Management Tools, Integration with enterprise Internal Infrastructure, Availability of compact economical applications, Cost/Performance prediction and measurement, maintaining transparency and Simplicity

4.2.2 Limitations

In the way towards realizing the vision starting from the current state of the art there are some obstacles. These obstacles form the limitations that are likely to encumber further progress of the Grids. However, at the same time, there are some ideas the implementation of which will comprise a breakthrough in the area, bringing Grids one step closer to their vision while at the same time, contributing to the convergence of the two research efforts.

Standards have entered the world of Grid architectures in order to resolve interoperability issues, not only between the -by default- heterogeneous resources but also between the various designs. However, in practice, things have turned out differently as using a standard in the design-phase and implementing it in the development phase has often led to incompatibilities. Moreover, it is not uncommon to see two different standardization efforts to work separately towards the same goal (e.g. WS-Eventing and WS-BaseNotification). The result is a set of standards that on one hand specify and profile the operation of at least a part of a Grid architecture but on the other create conflicts as to whether a specific version is the appropriate one or not. This mainly yields from the different experiences as well as of different goals that each development team might have, leading architects to a varying assessment of the maturity and the need of standards.

Another issue that seems to be hindering the process of grids and of the convergence in the initiatives of the two communities is the different use of grids in general. It seems that Grids in China are designed with a strong focus on the particular application that it is intended to serve, usually coming from the public sector. Grids do not necessarily have a business-oriented perspective, at least to the extent that European research and development communities would like to have. Thus, there is a difference in the business models support, at least at a conceptual level.

Still, there is a perennial problem to all the architectures. This is the different understanding of an architecture, a problem which especially in Grids is much more evident. The multilayer nature of Grids (from a resource layer to an application layer through the middleware layer and so on so forth) usually causes problems to architects that tend to design based on a different model. E.g. one may consider that an architecture is needed in order to define the profiles of the interactions between conceptual models whereas one other may suggest to a specific component model, explaining in detail what classes are needed. In summary, the lack of formulization of Grid architectures presents obstacles in realizing the vision.

Finally, country specific political policies which govern the nature of sharing resources, data, applications, user access to information, etc will play a major role in shaping the advances for this topic.

4.3 New ideas

One idea that will greatly help in resolving the compatibility between Grid infrastructures is the development of architectures based on principles (e.g. standards) that will focus on the re-use of existing infrastructures. Moving to more service-oriented frameworks will render specific Grid middleware obsolete, however specific techniques that have been applied there, are still useful. Moreover, as the “Grid Frameworks” will grow bigger to become large scale, distributed environments, they will very much resemble Enterprise Service Buses (ESBs). Each “service” on such an infrastructure can be a smaller framework that will have specific technical properties and may even satisfy different application requirements. Such a sub-framework could even be the Grid infrastructures themselves.

Another innovative idea that would benefit the Grid vision would be to design architectures that will support dynamic trust and security and SLAs. The components of these architectures must be deployed and configured in such a way that will enable trust and security assessment of the relationships in real time. The relationships must be fully described by SLAs that must also be dynamically assessed. Assessment will be achieved by profiling customers and providers, using historic information on their requirements and capabilities (e.g. QoS).

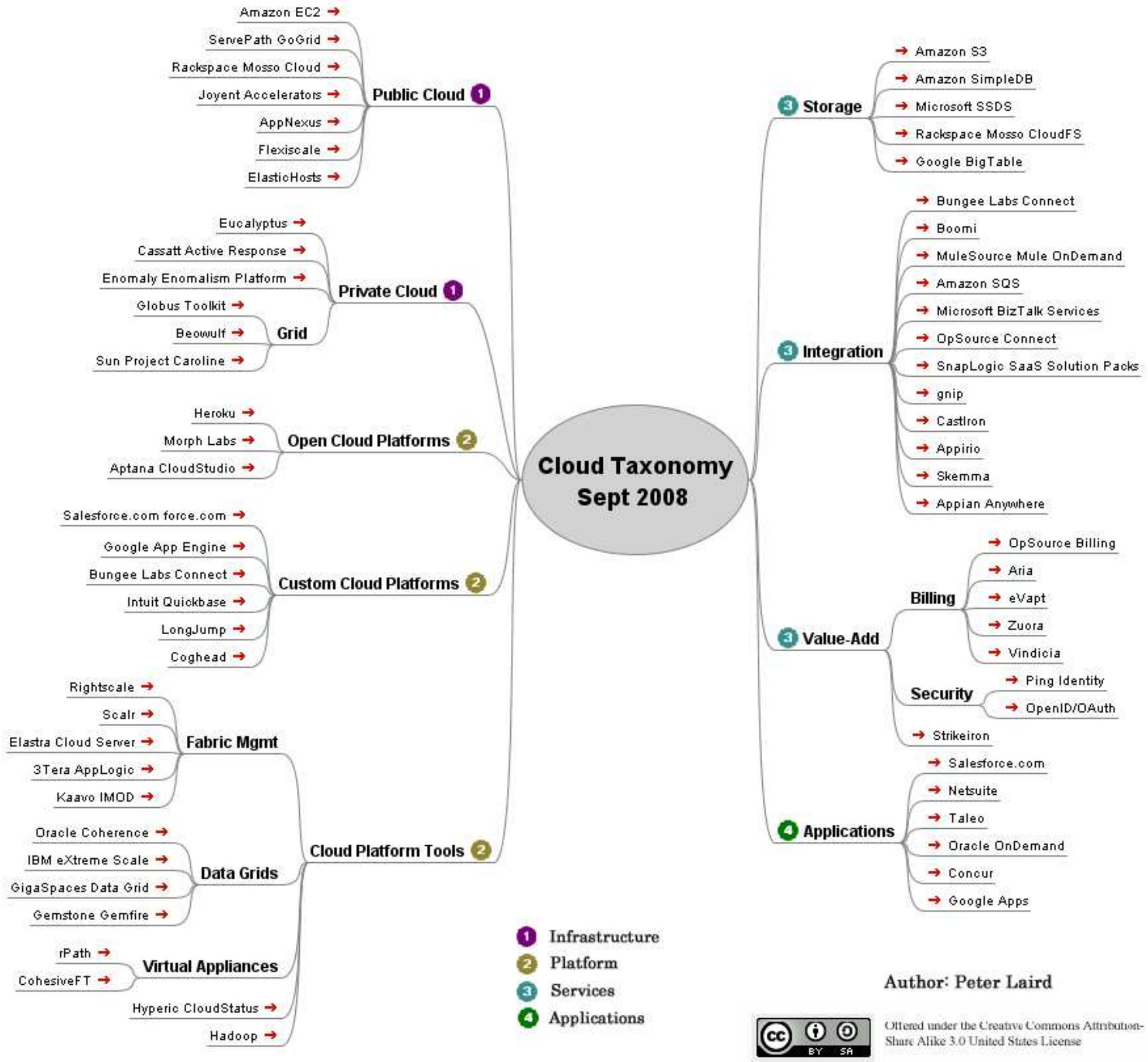
Another area that will help in the Chinese and European effort to materialize the vision is to make specific breakthroughs in the resource management mechanisms. Especially, it has been identified that designing and including mechanisms that will realize improved coordinating of the resource reservation. The idea is that the democratization of Grids will soon require a specific pool of resources to be used by a greater amount of applications. This means that resource reservation will become crucial in satisfying both the functional and non-functional requirements of the applications. It is expected that future resource reservation planning is a topic that, once properly addressed, will cover the needs of the vision.

By combining the European efforts around GRIA and the Chinese efforts around Xservices it should be possible to develop an open source framework for distributed SOA deployments, which should be able to compete with US developments like Websphere and WebLogic, which currently dominate this market.

Finally, there is a new emerging idea called “Cloud Computing” that shares a large amount of common assets with Grids. Both approaches can be seen as an implementation of “Utility Computing”. However, today Cloud computers expose resources to the users in a very different and simpler way than Grids usually do. With Grids, programmers have to adapt their applications to use specific Application Programming Interface while with Clouds, applications and their supportive environments (operating systems and middleware) are uploaded to remote resources and thus do not require any modification to benefit from

Clouds. The architecture behind cloud computing is a massive network of "cloud servers" within a data center. Resources are allocated to users on demand with their own software stack from the operating system. The business model behind Cloud computing is to charge users on their amount of CPU and storage space they use as well as the cost of transferring data from and to the cloud. Google, Amazon, IBM and Sun are very active in this area by providing Cloud services to customers.

A possible taxonomy of what is called Cloud Computing is shown in the figure below, which depicts a classification of the Cloud, by Creative Commons.



Other interesting predictions of new ideas about Cloud can be found in Appirio:

"Today’s economic climate will force enterprises to pick technology winners and losers for their environment in order to cut costs, be more efficient and deliver business-relevant innovation. Cloud computing makes this seemingly impossible task a possibility – much more so than with traditional software. This is why we believe cloud computing will be counter cyclical, with SaaS and Platform as a Service (PaaS) investment accelerating, and traditional software spending declining."

1. **The "cloud of clouds" expands but sees traction revolve around open platforms.** We'll see Microsoft and other traditional software players invest even more in new but closed cloud platforms. At the same time, proponents of a more open approach, like Amazon, Facebook, Google and Salesforce, will push more and deeper "cloud connections" like they did this year. This will create a more heated debate between the value of closed versus federated platforms.

2. **At best, Microsoft Azure will be a better platform for Exchange.** Microsoft will continue to shower attention on Azure but will see relatively limited adoption from ISVs and customers. While it will likely disappoint users and remain well behind established cloud players for the first few years, it will become a viable platform by 2010 – primarily as a better foundation for Microsoft Exchange and existing on-premise .NET applications.

3. **Google doubles down on the enterprise, enterprises return the favor by racing to Google Apps.** Google has already shown they're serious about winning over enterprises with acquisitions like Postini and investments in Google Apps. They'll continue to expand their support for enterprise-class security, transparency, and development languages. In return enterprise customers, faced with economics that overcome preconceptions, will substantially increase their pace of adoption. We expect to see at least 3X the number of enterprises evaluating and moving to Google Apps, at the direct expense of Microsoft Exchange, Office and Lotus Notes (the Asbestos of Software).

4. **A major SaaS 1.0 company will fail.** Although SaaS and cloud investments will increase next year, a number of SaaS 1.0 companies – stand-alone companies who built their SaaS products from scratch on their own - will either falter due to the demands of creating infrastructure, or chose to re-platform. The progress of enterprise-ready platforms like Force.com makes it much easier for SaaS 2.0 companies to build advanced products that can leap ahead of the competition at a much lower cost.

5. **A rise in serverless companies with 1000+ employees.** In 2009, the market will start to hear about more and more companies going completely server-less. While this is already happening at smaller companies, larger and larger companies will optimize their business processes and cut IT expenses by outsourcing to cloud providers.

6. **The rise and fall of the private cloud** - While private clouds will continue to generate a significant amount of hype, customers in most cases will realize they are little more than a better data center implementation. They will be valuable for customers who have significant transaction volumes and stringent regulatory or security requirements, but will have little ROI for the average IT organization. In the end, private clouds will create more value for service providers than for customers.

7. **Business Intelligence (BI) becomes the next functional area to SaaSify.** Just as CRM and HRM applications became poster children for the shift to SaaS these last few years, we'll see the same thing happening with on-demand BI. We'll also see a

bifurcation in this space, with one set of applications built from the ground up to leverage the inherent benefits of cloud computing and one set a repackaging of traditional BI features just delivered over the Internet.

8. **SAP or Oracle gets into the PaaS game.** While these companies may have hedged their bets in 2008 (or even berated the SaaS model), we believe one of these companies will see the writing on the wall and start at least talking about a new cloud platform they're building over the next few years. In fact, they will attempt to switch the conversation and convince the market they have been working on this for years but called it something different.

9. **Enterprises will figure out how to use social networks in the right way.** Companies – especially their HR and marketing organizations - will finally figure out how to utilize social networks in day-to-day operations. More and more business (employees, leads, market intelligence) will come directly through business applications that tap into Facebook, Twitter, LinkedIn and other social networks that are already being used by employees and customers outside the workplace.

10. **There will be at least one $100M software product built on Force.com.** The myth that it is impossible to build a big business on an on-demand platform will finally be debunked by the emergence of a PaaS-enabled application in 2009 that has the potential for a $100M run rate.

Source: http://www.appirio.com/

The purpose of grid computing is to collect the resource scattered across the Internet. This purpose originally has close relationships with two principles: REST (Representational state transfer) and Enterprise Integration Architecture REST (Representational state transfer) Representational state transfer (REST) is a style of software architecture for distributed hypermedia systems such as the World Wide Web. The terms "representational state transfer" and "REST" were introduced in 2000 in the doctoral dissertation of Roy Fielding, one of the principal authors of the Hypertext Transfer Protocol (HTTP) specification. The terms have since come into widespread use in the networking community. REST's proponents argue that the Web's scalability and growth are a direct result of a few key design principles: 1. Application state and functionality are divided into resources 2. Every resource is uniquely addressable using a universal syntax for use in hypermedia links 3. All resources share a uniform interface for the transfer of state between client and resource, consisting of a) A constrained set of well-defined operations b) A constrained set of content types, optionally supporting code on demand REST is still HTTP-based solution, but provides a more natural way to represent resource state which is similar to WS-Address in Web Service Specification. RESTful design has gained widely accepted, and considered as an effective way for building computing infrastructure in the Internet environment. Enterprise Integration Architecture EIA (Enterprise Integration Architecture) has gained widely used in enterprise integration. The recently idea in this area includes ESB (Enterprise Service Bus), SCA (Service Component Architecture) etc. ESB and SCA can be an effective way to assemble heterogeneous grid middlewares to form a bigger grid infrastructure. Besides REST and Enterprise Integration Architecture which provide new ways to encapsulate different resources and construct grid infrastructure, we think more attentions should be paid to some vital features for long-running grid infrastructure such as fault-tolerance, scalability etc.

4.4 Make it happen

4.4.1 Three years roadmap

There is a clear need to carry out standardization in the area of Cloud to allow customers to choose their resource providers. Today, deployment of applications within a software stack that can be uploaded on cloud systems cannot be simply done. Each cloud provider has its own way to represent the software stack. As a result, the lack of standards creates obstacles in the interoperability or even the integration of applications within the Cloud infrastructure. In this context, EU and China should collaborate closely on defining these common interfaces, APIs and standards that will set the foundations of this new architecture.

Business adoption: In 3 years, the technology would be more mature, and ease-to-use for large companies to adopt in several sectors. The standardization of a few core "activities" that are targeted at the interoperability will make Cloud an effective solution easily adoptable from the SMEs. Suppliers of Cloud computing will be able to stabilize their infrastructures to be able to offer a reliable and a very well qualified Grid system to the users.

4.4.2 Five years roadmap

It is foreseen that there will be several cloud providers and thus the selection of one of them for a particular application will be an intricate issue due to the complex business model associated with such computing systems. The cost will depend on many factors: CPU usage, storage usage and communication. A brokering mechanism will be thus required to select the less expensive cloud taken into account the characteristics of the applications. There will be a clear interaction here with the research activities that have been carried out in the context of Grids to broker resources. Moreover, applications will probably require more than a cloud system and thus several clouds will have to be combined into a single system. Again, all the work done around the concept of virtual organization concept will need to be adapted to this context.

In addition the foreseen evolution of Grids within the next five years will be mainly focused on the improvement of e-infrastructures. This improvement includes elaboration of user friendliness so that the complexity of infrastructures will be hidden from the users.This will facilitate massive “democratization” of Grids within the communities of citizens and individual users, through the introduction of the innovative applications they enable. The evolution will also consist development of infrastructures comprising multiplicity if sensors both in number and nature. Nanotechnology and biosensors are expected to be introduced to infrastructures and achieve performance and interconnectedness. Finally advancements in QoS and security issues remain key factors for commercialization of Grids and their wide adoption by the business sector.

4.4.3 Ten years roadmap

Trust and security are important issues in the context of Cloud computing. Since the model promote the use of remote resources instead of local resources, companies will have to trust the clouds systems they will use. Today, there is no way to protect a cloud user from an attempt to monitor what it is being done within the cloud by the cloud provider itself. To be successful, there is a clear need to study some isolation mechanisms to prevent anyone to get access to what is being running within a slide of the cloud allocated to a particular user.

Security and legal issues are some of the concerns about cloud computing that are delaying its adoption. In order to confront this threat, the development plan for the next ten years should be focus on the next three topics:

- Application Security
- Encryption and Key management
- Identity and access management

Business adoption: In ten years: The trust in the technology won't be an issue any more and the providers will trust the users and vice-versa.

4.5 How it is covered in existing EC and Chinese R&D Programmes

It is commonly accepted that the both the Chinese and European workprogrammes for Grids have already paved the way for realizing what has been set as the short-term vision for Grid architectures.

4.5.1 EU Work Programme Initiative Status

In FP7, research in Grid Technologies is part of the ICT Challenge "Software & Services Architectures and Infrastructures". Its purpose is the continuity of the FP6 Grid Technologies projects and activities as well as the closer integration with current trends and initiatives towards service-oriented architectures and infrastructures.

In the area of Cloud computing, the newly funded FP7 project, called RESERVOIR, aims at enabling massive scale deployment and management of complex IT services across different administrative domains, IT platforms and geographies. The project will provide a foundation for a service-based online economy, where, using virtualization technologies, resources and services are transparently provisioned and managed on an on-demand basis.

4.5.2 China Work Program Initiative Status

In China, The National High-tech R&D Programme (863), sponsored by the MOST, is the major source of government funding on Grid research in China. The biggest effort is the China National Grid (CNGrid). The first phase of CNGrid was supported by the project "High performance Computer and Core Software" from 2002 to 2005, with a government funding of 100 Million RMB. The second phase of CNGrid was launched in Sept. 2006 with a government funding of 640 million RMB from the new project "High Productivity Computers and Grid Service Environment" from 2006 to 2010, and one of the project's targets is to support SOA inside Grid environments, too.

So far under the China national R&D programme, the cloud computing has not been formally supported. But projects with similar concepts are supported, for example, virtualization of computer systems including the servers and desktops, and data centric computing, etc. In the current round of call from the 863 programme, mission oriented projects are called on "virtualization based cross-domain resource sharing and cooperative working" and on "data driven cross-domain sharing and collaboration. The contents of these called projects are quite close to the cloud computing. Also, cloud computing has had its influence to the current development activities on Grid software.

4.6 Recommendations for EC and China

There are a number of steps that it would be advisable for the two research communities to follow in order to take the state of the art closer to the vision. These steps are:

- Put effort in dynamism (such as dynamic service selection, composition, business models support, etc)
- Emphasize the development of WS standards for grids that will also support backwards compatibility
- Develop open source competitors for tools that support distribute SOA infrastructures (e.g. WebSphere)

Starting from the visionary goal to move to SOA frameworks rendering obsolete monolithic Grid systems, it is advised that effort must be set to support dynamism in various fields: service selection, service composition, SLA composition, business models, etc. Broadening the capability of Grid infrastructures to support more applications and end-users, more systems and services, greatly increases the dynamicity of the environment. Along to that must come management systems to enable the administration of these environments while encompassing policies and several important criteria (e.g. business-oriented criteria) for any selection made.

Moreover, the two research communities should invest more in developing web services' standards for promoting the Grid concept. These standards must enhance the capability of the developed infrastructures to integrate with Grid systems that are now outdated. Moreover, it is important to find a way to design architectures that will be able to put together frameworks that perhaps are created to support a shortened range of applications but in a very successful way.

Cloud computing will be sharing resources among China and Europe, once the relationship among the two communities will be closer. We have to think about the idea of a global world, where the business will be closer to the end-user side and the companies shall cooperate together to provide the best solutions to the customers.

Clarify the concept of grid architecture. The standards of grid platform should be unified. The services can be transplanted and deployed easily. The differences among client hardware can be shielded well. At the same time, upgrading the software version and keeping compatibility can be taken into account.

As a conclusion we can comment that the collaboration of the two research communities should be enforced in terms of common research projects and proposals. The bonds created through the EchoGRID project should become the foundations for future cross-continent research initiatives.

4.7 References to the elements that lead us to the conclusions and roadmap

http://www.appirio.com/about/pr_predictions-09_121808.php
http://aws.amazon.com/
http://code.google.com/appengine/
http://www.cloudsecurityalliance.org/
http://www.irmosproject.eu/
http://www.gria.org/
http://www.nextgrid.org
http://www.scai.fraunhofer.de/about_simdat.html
http://www.eu-trustcom.com/
http://www.soa4all.eu/
http://www.sla-at-soi.eu/
http://www.reservoir-fp7.eu/
http://www.nexof-ra.eu
http://www.cngrid.org/
http://echogrid.ercim.org/component/option,com_contribution/task,view/

5 GRID MANAGEMENT

This chapter presents the vision for management problems in the next generation Grid systems. Grid management is term that in reality encompasses a wide range of issues, including, but not limited to, security, autonomic computing, self-managed systems, flexible data management, accounting, auditing, and context awareness. We shall follow the popular "FCAPS" management methodology to analyze the Grid management problem and state-of-the-art and present a roadmap for Grid management research and industrial trends in the upcoming 3, 5, and 10 years.

5.1 Vision

The term Grid Management is in itself quite self explanatory and also in the same sense as Grid induces ambiguity to scope of this topic. The approach to define a vision is to start identifying the fundamental aspects associated with management and couple it with motivational scenarios for the usage of the Grid and its relevance to the stakeholders.

The Grid in its basic essence has emphasized on "Widely Distributed Secure Sharing" aspect as applied to the shared entity usually referred to as resource. This driving force should be supported based upon the following:

a) Local provision considering security and privacy issues.
b) On-demand access to content and adaptation of the content to context.
c) An ecosystem of autonomic, self managed systems.
d) The contexts could be users, devices, location and policies including and not limited to state, organization and user.
e) Capabilities of handling very large data volumes with long term digital curation assessed by business. Eg: 2 to 10 PBs per year per business.
f) Implementing dynamic business models with clear accountability of consumption and pricing issues across industry, enterprise and research community.
g) Economy of scale considering factors like cost, GreenIT, energy, cooling – modelling renewable resources.

With the above motivations, we envision scoping the management of Grid through a "Utility" model scaling across geographic zones of enterprises and the heterogeneous devices and appliances leading to a merged vision – Pervasive Utility model. This utility model will be applicable to the widely distributed secure sharing of resources including but not limited to computing, storage and bandwidth.

The management is best understood by analyzing based on the fundamental aspects as mentioned below and popularly termed as "FCAPS" management:

a) Fault Management
b) Configuration Management
c) Accounting or Charging management
d) Performance management

e) Security management: Authentication, Authorization, etc.

We elaborate a roadmap for Grid management for 3, 5 and 10 years by rationalizing the above fundamental aspects with a unison vision of Grid layers as below:

a) Resource
b) Connectivity
c) Collective
d) Applications – supported by the Grid infrastructure
e) Users – To be analyzed from stakeholder perspective.

5.2 State of the art & Limitations

The current status of Grid management is still evolving in EU and China with both addressing specifics in resource management.

5.2.1 State of the Art

- EU engineering initiatives in Grid management:

  a) UK National Grid Service (NGS): STFC operates the UK National Grid Service (NGS) for researchers to use national and international computing and data resources. The NGS has about 400 real users of whom about 100 use about 90% of the resources. The STFC e-Science department of 140 people operates the UK NGS as well as other national resources for computing and data management including the 6 Petabyte Atlas data store, the national scientific visualisation service and compute servers for several research communities.
  b) Efforts on resource management, scheduling and load balancing. Development of an Execution Management System (EMS) where QoS plays a significant role in the distribution of tasks. Efforts on SLA management and accounting for supporting systems that provide strong QoS guarantees.
  c) Other EU efforts: Quite a few projects under the FP6 program tackle various aspects of Grid management problems. These include, but not limited to, EGEE, Grid4All, and XtreemOS.
     a. **EGEE III** has putting accounting as a priority service as it is paving its way to provide quality supports for EGEE production Grid infrastructure. Since 2006, EGEE's accounting service has undergone serious development phases, including development for accounting client and server software, and accounting portal, which have all been included in the latest GLite distribution. Since 2006, EGEE accounting service collects over 1 million records on a monthly basis, which has gradually moved up to 10s of millions monthly in 2008.
     b. **Grid4All** aims to transform Grids into a ubiquitous utility for domestic users, small organizations and enterprises without them repeatedly investing in Grid technologies and infrastructure. Component-based management architecture is exploited by the project to provide self-organizing P2P overlay infrastructure underpinning Grids.

c. **XtreemOS** is a serious effort funded by EU FP6 programme to adopt a bottom-up approach to provide natively support for Grids from operating systems. Unlike the ad-hoc fashion that Grid management issues were taken into account by Grid middleware systems, XtreemOS considers Grid management as an integral part of its design from the very day one by envisaging the role of P2P technologies (e.g. distributed hash tables) and fault tolerance techniques (e.g. virtual nodes, distributed servers) in its overall system architecture. Specifically, a layer of highly available and scalable services is incorporated into the system infrastructure to provide seamless support to enhance the availability and scalability of application management, distributed file management, and VO management.

- China engineering initiatives in Grid management:

a) CNGrid Operation Center: CNIC operates the China National Grid Operation Center since 2004. On Dec. 21, 2005, XU Guanhua, who was the Minister of MOST, and Sir David King, who is Chief Scientific Advisor of the UK Government attended the Unveiling Ceremony of China National Grid Operation Center in CNIC. CNGrid operation center is responsible for Grid operation and monitoring, providing support for Grid nodes and users, promoting international cooperation as CNGrid representitive. CNGrid operation center also makes rules and regulations under the guidance of the CNGrid academic committee.

b) iVCE: Internet-based Virtual Computing Environment, is a 5-years project funded by National Basic Research Program of China (aka. 973 program).The first focus is Resource Aggregation Model, Mechanism and Computational Properties (iVCE-2), the other is called Resource Collaboration Model, Mechanism and Computational Properties (iVCE-3).

c) CNGrid: China National Grid is engaged in Grid resource monitoring and management in CNGrid Environment. CNGridEye is an operational monitoring and management system for CNGrid environment.

d) NSFC: National Science Foundation of China supports a Grid standard project which includes Grid management aspect.

e) SDG: Scientific Data Grid has focused on SDGFinder to search for registered services in a registry

f) Bird Flu Monitor Grid: This focuses on security of data resource sensors and mobility management issues with bird migration.

g) CNGrid GOS: Naming management in GOS is a resilience decentralized registry for variety kinds of global object. It provides low latency object locating by object GUID, high success rate searching by multiple attributes match, stable object view based on linked naming services to enable the effective-virtual-physical address space. User management, resource management, agora management functions are based on the naming. The static metainfo in HPCG is also stored in naming.

5.2.2 Limitations

The state of the art and limitations in Grid management are presented in the below descriptive table:

	Fault	Configuration	Accounting	Performance	Security	Dynamic capabilities
Resources	Tools for individual resources	Tools for individual resources	Tools for individual resources	Tools for individual resources	Tools for individual resources	Tools for individual resources
Connectivity	ITU-T TMN, MSeries	Mseries	MSeries	MSeries	MSeries	Proprietary implementations possibly based on the CMIP stack or SNMP.
Collective	(gLite); Globus GOS GRIA (Bridge) Monitoring & notification	gLite (resource discovery) GOS (UDDI-like registry), iVCE	EGEE (resources not money; compute not storage) GOS (no accounting)	Some monitoring	trust certificate PMA	GOS (monitoring)
Applications (supported by Grid) solutions)	Some jobs reallocate resources on failure. No check-pointing framework	Stand-alone with much local knowledge required; limited hot service deployment	Not even defined	Isolated applications only; no performance metrics	No application-level security	No autonomic management. Heterogeneous application profiling/management requirements.
User	User currently manages faults	Aligns application requirements to resources	nothing	Tunes job submission	Certificates with proxies (manages own credentials)	No portable user context

Table 1: State of the Art and Limitations in Grid Management

To summarize, the management issues that a Grid Service Provider will face from the current date to 2018 are as follows:

a) Regulation around Grids with emphasis on Certification, Accounting rules, establishing trust domains.
b) Technologically, standardization of interfaces and protocols.
c) Management of Fault, Configuration, Accounting, Performance, Security and Dynamic capabilities dependent on resource, environment or the object.

5.3 New ideas

The new ideas proposed are primarily aligned as 3 milestones to be achieved over the next 10 years. This aims to bring about a consistent thought process of what is achievable and what really requires long term research within the time frames.

5.4 Make it happen

The new ideas will evolve over time; the format provided below suggests a rationale to align the focus as innovations are achieved through research.

a) 3 years roadmap: Standardization and prototype
b) 5 years roadmap: Declarative semantics based management
c) 10 years roadmap: Autonomic Grid management

5.4.1 Three years roadmap

Capturing the ideas of SOKU, the definition of the three key notions:

- Service Oriented: the architecture comprises services which may be instantiated and assembled dynamically, hence the structure, behaviour and location of software is changing at run-time;

Emphasis is on Standardization and prototyping of the vision of Grid management to make it consistent across EU and China. It demands extensive use of existing technologies and development of consistent frameworks across vendors. The below table illustrates the vision:

2011 - STANDARDIZATION & PROTOTYPE	Fault	Configuration	Accounting	Performance	Security	Dynamic capabilities
Resources	Cross-vendor standardization					
Connectivity	Better wireless (heterogeneous) support within the current policy framework					
Collective	Improved accuracy in fault location; fault correlation	Standardized definition & procedures	Defined common cost model for (bandwidth), data storage, & processing	Defined monitoring performance, management of failure & standardized SLA.	Interoperability of diverse security management systems	Extending existing management information models like CIM to cope with heterogeneity & dynamics of Grid

Applications (supported by Grid) solutions)	Defining check-pointing framework for basic recovery.	Generic description of applications (prototypes)	Definition of application cost model & prototype optimisation tools.	Definition of application metrics.	Rationalize application-level security	Specify supporting environment for Virtual Organizations.
User	Fault-analysis tools	Configuration management tools	Define framework for Grid economics	Performance analysis tools	Standardization of user access to grids & associated tools	Specify framework for analysis of FCAPS analysis and management

Table 2: Vision 2011 - Standardization and Prototype

5.4.2 Five years roadmap

- Knowledge: SOKU services are knowledge-assisted ("semantic") to facilitate automation and advanced functionality, the knowledge aspect is reinforced by the emphasis on delivering high level services to the user;

Emphasis is on <u>Declarative semantics based management</u> for grids. This will meet the primary aspect of Grid – Interoperability in dynamic environments. Rules based management of Grid systems is suggested to be the focus. The below table illustrates the vision:

2013- Grid MANAGEMENT BASED ON DECLARATIVE SEMANTICS	Fault	Configuration	Accounting	Performance	Security	Dynamic capabilities
Resources	Standard specifications for FCAPS management					Interoperability of management across the areas of management considering heterogeneous site Grid.
Connectivity	Research on declarative semantics (rules) based management of Underlying networks.					
Collective	Automated resource & connectivity re-allocation	Automatic configuration tools & tools-assisted configuration & composition.	Declarative semantics for accounting policy specifications	Declarative semantics for Performance management specifications	Declarative semantics for security management specifications	

Applications (supported by Grid) solutions)	Standards for check-pointing & fault reporting. Demonstration of migration of processes across nodes after faults	Standardisation of application descriptions & mapping to configuration. Workflow libraries.	Development of optimisation tools covering costs semantics	Development of application level semantics for statistical analysis	-	
User	Helper tools for fault recovery, management.	Tools and Semantics for capturing inputs for self configuration and reconfiguration	Declare requirements for Grid usage accounting & decision-making	Helper tools for Performance management	Rationalize requirement for universal identity management for Grid.	

Table 3: Vision 2013 - Declarative semantics based management

5.4.3 Ten years roadmap

- Utility: A utility is a directly and immediately useable service with established functionality, performance and dependability, illustrating the emphasis on user needs and issues such as trust.

Envisioning the usage scenarios of Grid in 10 years, the proposal is to focus on Autonomic Grid management across federated systems with reasoning capabilities. The below table illustrates the vision:

2018 - AUTONOMIC Grid MANAGEMENT	**Fault**	**Configuration**	**Accounting**	**Performance**	**Security**
Resources	A framework for self management for reasoning and executing run-time policies governing FCAPS management				
Connectivity	Self Healing Networks	Context Aware self configuring/re-configuring networks	Dynamic and adaptive Grid economics implementation like 1+1+1=2 cost units. (Sum of parts is not equal to the whole)	Self tuning/ optimizing networks	Self aware networks. Develops Dynamic shield across different types of network attacks.

Collective	Self Healing enablers and management	Self-configuration & composition. Autonomic framework with semantic-based configuration.	Requirement & resource-aware accounting system (context aware)	Self tuning / optimization enablers and management	Self awareness enablers and management
Applications (supported by Grid) solutions)	Automated management of fault	Application workflow/semantics to consider configuration semantics	Adaptive accounting system to optimize resource usage based on cost	-	-
User	Physically replace resources.	Policy specification that would enable auto-configuration or self configuration/ re-configuration	Decision delegation for accounting needs.	No action required	Advanced security interfaces including biometrics.

Table 4: Vision 2018 - Autonomic Grid Management

5.5 How is it covered in existing EC and Chinese R&D Programmes

Grid management has been the non-priority topic in the research world. The main focus is usually on the functional aspect of the grid. The functional aspect has been evolving and aimed at interoperability in the existing Grid infrastructures. The management aspect of the Grid had been only seen as a tool for easing the workload upon the stakeholders for assisting analysis of resource metrics. Both EU and China might not be thoroughly aware of the need for establishing strategic partnerships in Grid Management.

5.5.1 EU Work Programme Initiatives Status

The EC funded Framework Programme (FP) projects have been instrumental in the development of Grid research and have attained considerable maturity in all aspects of Grid. The FP6 projects like NESSI-Grid, BEinGRID, Akogrimo, XtreemOS and many more have been focusing on various Grid aspects for specific resources and are evolving specific management functions. There is now a need to bring in all the common functions and tabulate them more formally and understand the implications for the standard interactions for management across resources. The EC funded FP7 initiatives should consider explicit focus on Grid Management.

5.5.2 China Work Programme Initiatives Status

The National High-tech R&D Programme (863), sponsored by the Ministry of Science and Technology (MOST) of China, and National Natural Science Foundation of China (NSFC) are the two major sources of government funding. The National Basic Research Program of China has also involved in funding through the 973 projects. The above initiatives have made considerable efforts for general Grid initiatives in China. The projects under these programs like iVCE, CNGrid have taken considerable initiatives in pursuing challenges for Grid Management as illustrated in the State of the art. A Grid monitoring and management system CNGridEye has been developed by CNGrid teams, which provides basic resource monitoring functionalities for the CNGrid environment. CNGridEye has been deployed on 10 CNGrid sites and is supporting the operation of CNGrid.

5.6 Recommendations for EC and China

To provide further inputs and realize the above mentioned visions, the following recommendations are proposed:

a) Grid management should be an explicit focus of research in both EU and Chinese research initiatives like FP7 and 863 programmes and focus should be on SMEs and industry to experiment with innovative business models upon them.
b) Grid management should take considerable input from the Telco standard fraternity like ITU-T, TMF for Grid to enter into a clear managed scope of business enabler. Today we have considerable working Grid efforts and the motivation to keep these efforts functional will require the accountability of these Grid infrastructures in a global business scenario. Without Grid management practices in place, the funding for these initiatives will lack clear motivation.
c) Additionally, ETSI could take the role for Grids in Europe, the Ministry for Information in China that ITU does for Telcos, as Regulator and Standardization body.

Grid management system can provide a better interface to monitor grid operation and dynamic user management. Provide fault-tolerant mechanisms to enhance stability.

5.7 References to the elements that lead us to the conclusions and roadmap

http://www.hoise.com/primeur/06/articles/monthly/AE-PR-03-06-24.html
http://portal.acm.org/citation.cfm?id=1270324

6 VIRTUAL ORGANISATION

While Grid Computing is gaining ground and public recognition - not only within academia but also in industry - as a flexible and coordinated resource sharing technology, the technology that will define and manage the relationships between the groups of individuals or organizations is of great importance. Many years ago, distinctive scientists identified the need for VOs[1] as well as Virtual Enterprises[2] and provisioned the activity that is necessary in that context.
In this chapter we present the vision of Virtual Organizations (VOs) that has been emerged as an interconnected concept along with evolution of Grid Computing. In addition, we indicate the state-of-the-art in that filed as well as the limitations of the current associated technology. The outcome of this inquiry is a three, five and ten year roadmap that depicts the future trends.

This chapter is intended to compile all the actual background of what is going on in terms of Virtual Organisations research, what the vision, state of the art and new ideas of them are. Furthermore we try to summarize the work being done in the European Countries as well as in China, and try to foster the cooperation and the relationship among all the trends in the research field that tries to compile this chapter. Finally the make it happen section try to foresee and visualize the short, mid, and long term predictions of the so called Virtual Organisation term, that is converging to the new virtualisation and cloud computing trends.

6.1 Vision

In a changing and evolving world of services, a virtual organization (VO) is a dynamic group of individuals, groups, or organizations who define the conditions and rules for sharing resources. The concept of the VO is the key to Grid computing. It is a buzzword in recent years. Different projects have different definition for their Virtual Organization. But all VOs share some characteristics and issues, including common concerns and requirements that may vary in size, scope, duration, sociology and structure.

From our survey among some partners of EchoGRID, a VO should have following functionalities, such as user/role management, resource management, trust and security management; and a VO should have the properties including: dynamicity, QoS guarantee, scalability; also for a VO that wishes to exist stably for longer time, it should have a business

---

[1] The Anatomy of the Grid Enabling Scalable Virtual Organizations [Ian Foster, Carl Kesselman, Steven Tuecke]
[2] VEnterprises [Skyrme et al]

model underneath.

The rising of SOA (Service-Oriented Architectures) has impacted the Grid technologies. Since SOA encompasses architecture issues in both business and information systems, it has great potential to enable alignment of VO approaches with IT-based business networking approaches, such as enterprise interoperability. So while SOA can contribute to VOs, VOs can also make their own demands from the new architecture.

The enterprises should start adapting their organisations to be part of a big one world wide organisation, being able to interoperate within other similar organisations to collaborate and aggregate services to provide to the final customer a sensation of specialization and organisation as a whole, keeping the individualities of every enterprise.

The future internet assembly started in Bled in April 2008 stated the basis of a new world of services and things, that will ensemble and make happen a new reality of what the end users expect from the internet, and a key issue for achieving the abovementioned expectations was the special focus on virtual organisations.

The secure enactment of collaborative business processes in self-managed and dynamic value-chains of businesses and governments. One of the central aspects of the current research is VO management, which aims to support dynamic virtual communities throughout their entire life cycle. Although proprietary implementations of VO management tools exist, secure tools based on interoperating open standards are not yet available. The open standards on which to build them are just being released as reliable implementations.

6.2 State of the art & Limitations

VO is a key part of a Grid platform; almost every project on Grid will do research on VO, or use the mature techniques of VO. We list projects from EchoGRID partners that deal with VO below; an already mature idea is the one coming from BEinGRID that is related with the concept of Virtual Hosting Environment. Other initiatives like BREIN depict the evolution of the state of the art of TrustCoM from a non solved issues in this previous project to a new state of the art of the VO concept.

6.2.1 State of the Art

One of the efforts visible today is XtreemOS. They are concerned with building and promoting Linux based operating system to support virtual organizations for next generation grids. It is compliant with OGF SAGA (Simple APIs for Grid Applications) specification. Installed on each participating machine (personal computer, cluster of workstations, mobile devices), the XtreemOS system will provide for the Grid what a traditional Operating System offers for a single computer: abstraction from the hardware and secure resource sharing between different users. It will thus considerably ease the work of users belonging to virtual organizations giving them the illusion of using a traditional computer, and releasing them from dealing with the complex resource management issues of a typical Grid environment. By integrating Grid capabilities into the Linux kernel, XtreemOS will also provide a more robust, secure and easier-to-manage infrastructure for system administrators. This will be experimentally demonstrated with a set of real applications, provided by well-known industrial partners that cover a large spectrum of application fields.

Another big project in the FP6 program is BeinGRID.The mission of it is to establish effective routes to foster the adoption of Grid technologies across the EU and to stimulate research into innovative business models, namely

- To understand the requirements for Grid use in the commercial environment, involving software vendors, IT integrators, service providers and end-users.
- To enable and validate the adoption of Grid technologies by business.
- To design and build a Grid toolset repository with components and solutions based on the main Grid software distributions including: the Globus Toolkit, gLite, Unicore, Gria and basic **Web Service specifications**.
- To develop and deploy a critical mass of Grid-enabled pilots, embracing a broad spectrum of economic sectors with different needs and requirements in terms of technological Grid challenges.

In particular there is one business experiment that deals with the Virtual Hosting environment that is an implementation of the next generation virtual organisation that aims to provide a distributed ”virtual hosting environment” (VHE) that allows securely exposing applications as a managed services and enforcing quality of service in dynamic federations of users, application service providers and application hosts:

- From an application service provider’s perspective, this innovation allows businesses to provide, manage and control own application services, while outsourcing service deployment and management, security, and contact enforcement to the infrastructure services hosted by the network. In addition, they can take advantage of the distributed VHE in order to dynamically scale and load balance by distributing the hosting of their resource intensive or response time constrained application.
- From an application host provider’s perspective, this innovation allows businesses to optimize the use of their resources by accommodating diverse application services from potentially different ASPs that are can be provided in distinct contexts and distinct federations that meet different security and quality of service requirements, as required.
- From a converged IT and communication services provider’s perspective, this innovation has the potential to create a new market of network-hosted infrastructure services that are offered as “common capabilities” over the network, that enable the VHE operation and life-cycle management.

The main technical challenges to be addressed through use of standards-based Grid and Web Services technologies are:

- To architect scalable solution, for efficient and reliable program execution over distribute Virtual Hosting Environments.
- To implement a Virtual Hosting Environment supporting the operation and life-cycle management for federations of Common Capabilities, Game Servers, and Game Instances, and ensure the applicability of the result across vertical markets.
- To provide a suitable model for securely aggregating: Groups of gaming servers hosting game instances in virtual hosts; Collections (converged network) services supporting the execution a game instance; Communities of gamers sharing the same game instance; while maintaining the separation of administrative authority and concerns between gamer communities, application service providers, service host providers and the enabling infrastructure.
- Provision of internet programs to the consumer via non-intrusive, secure, pervasive and scalable technology.

Another related project in the field of the VOs is BREIN which tries to solve several problems regarding security, access control policies, Workflow enactment, Service Level Agreements, Business process management and legal contractual issues. A brief Idea of what the project do in an scenario of virtual engineering design is described below:

Workflow enactment is the process in which a functional specification of an activity is transformed into an executable service, which is then invoked. This is a complex process where the process activities have to be identified, their flow, the procedural rules for joining those activities and the associated control data. Workflow enactment also requires an orthogonal business process for each activity, which includes discovery for services capable of satisfying the activity's functional specification, SLA negotiation, security support, resource allocation, job distribution and job deployment.
During the workflow enactment, a job execution may fail for several reasons (e.g. SLA violation or resource failure). These failures cause the service to be unavailable to the customer, which is an undesirable situation for the customer, and the customer will most likely have given the provider a reasonable incentive to provide an available service though the use of compensation terms in the SLA that are payable to the customer in the event of a service failure. Alternatively, a provider may choose to violate an SLA, if they have a good enough reason to do so. A 'good enough reason' is that a better customer requests service and the benefit of serving the better customer is greater than the cost of paying penalties to the lesser customer.
To counteract failure and provide a reliable system, adaptation is required on different levels: in the case of an SLA violation, the workflow enactment could locate another provider and relocate the job execution to the new provider, or a provider whose resources have failed could re-allocate resources to that they can fulfil their SLA.
Given that providers face a number of conflicting demands from their users (of which there may be many), the complexity of the resource management is substantial. Resource management implies job distribution over the available resources, execution monitoring to detect faults, and SLA fulfilment, especially for the most important aspects, such as service cost, SLA mapping, and security issues. All of these need to be mediated by the service provider's business policies, which provide the basic principles the provider operates by.
The service provider deploys its services and makes them available on the Internet. They also adhere to the BREIN architecture by providing functionality support for SLA negotiation and

access to the services. The service provider aims to improve their services visibility by advertising their services in service registries, and supporting the ability to negotiate service conditions and constraints.
The customer requires the execution of a set of activities with a specific thread of control. The customer wants to take advantage of the Business Process Management (BPM) theory in order to improve the efficiency and quality of its internal process. For this reason the Business Process is modelled as a workflow that is design using a Workflow Designer. The workflow can be simple or complex depending on the customer's needs. Once the authoring phase is concluded, the workflow is ready to be enacted, and may be saved for future re-use.
The enactment process is performed by the Workflow Enactor component. This tries to discover at least one service for each requested activity. This step is performed by interacting with the service registry to discover service provider that offer services capable of satisfying the functional specification of the activity in question. Once a provider has been found and selected, the enactor starts the SLA negotiation. If all parties are happy with the agreement proposed, a contract is committed, which permits the customer to access the providers services via the issuance of a security token.

Started in December 2005, project iVCE, Internet-based Virtual Computing Environment, is a 5-years project funded by National Basic Research Program of China (aka. 973 program). There are 7 sub-projects in this project. One of the sub-projects is called Resource Aggregation Model, Mechanism and Computational Properties (iVCE-2). One of the research topics of this sub-project is to figure out basic mechanism which supports the process of establishing virtual community/organization. Virtual commonwealth is a basic mechanism for establishing and managing VOs within a task context. It specifies abstract services to publish, discover and organize Grid resources for aggregation, and describes the closure of autonomic elements and environments needed for current task. The service interfaces are realized as a set of common protocols and common rules between autonomic elements and virtual commonwealth. The dynamic binding of autonomic elements and virtual commonwealth is achieved by join/adapt semantics.

In the project CNGrid which is funded by China National High-Tech 863 programme and China Science Grid supported by Natural Science Foundation of China, VO is implemented to maintain the user, resource, security information of a Grid community, and it is named as Agora. Agora provides address mapping between physical and virtual resources and a cross domain access control facility combined with DAC and MAC. Agora keeps records of runtime sessions; it provides resource abstraction to applications through a runtime construct: Grip.

Several essential characteristics in VOs had arisen in the Athens Workshop in June 2008, namely:

- Security
- Resource Provider tends to Service Provider
- Semantics Interoperability
- SLA Terminology
- Access control Policies
- Heterogeneous Platform

- Metadata associated matters

As well as come attributes that the VO have to contain:

- VO lifecycle management: Formation. Configuration, Negotiation, Adaptation, dissolution.
- Architecture OGSA.
- Mechanisms of reasoning in the negotiation phases.
- Standarisation in WS-* model.
- Conceptualisation
- Explicit semantics.

Especial attention to the so-called Short Living VOs, and the Identity attributes and the roles of the VO should be taken into account.

6.2.2 Limitations

Although many VO projects exist, there are still limitations that slowdown the step towards a practical VO.

For the functionalities, current VO lacks of:

- Supporting from the underlying OS and Grid middleware
- Supporting of commercial applications
- Underlying administration facilities for VO bootstrapping
- Usability is far away from the point it should be under the perspective of the user
- VO Management.
- Authentication mechanisms
  - X.509 –static
- Authorisation
  - Attribute certificates
  - SAML

Current VO lacks some good properties for better resource sharing and better user usages:

- Consuming too much resources, heavy-weighted;
- Quality of Service guarantees
- Dynamic mechanisms in several fields as contracts or creation and formation.
  - Scalability
  - Interoperability
  - Dynamicity
  - Flexibility
  - Lifecycle of the VO.
    - Planning, Formation, Operation, Adaptation, Dissolution.

Furthermore, most of the Grid resources are bound together by government funding, seldom industries participate into the VO operation, there is no stable and long-term model to maintain the VO. Current VO lacks:

- Business and economic models for long-term running and operation.

6.3 New ideas

In order to minimize the gap towards the vision, there are several new ideas proposed from academia to achieve the goals mentioned above. We list them below.

- Adding advanced functionalities into VO to better support for resource sharing.
- Better VO support for coordinated job execution, meta-scheduling, job migration. Job execution in Grid environment will focus on
  - a) state-based sessions or services or jobs,
  - b) deployment times lasting more than several minutes or hours, and
  - c) restricted mobility.
- Jobs (services or sessions) would be lasting from several minutes to a few of days. Coupled with dynamic allocation or de-allocation of resources, job monitor should be able to execute migration of jobs. Meanwhile, cooperation between jobs restricts the mobility of jobs, which should be guaranteed by job monitor to prevent jobs becoming uncontrolled. Check-pointing is also adopted to make sure secure execution of jobs, especially for parallel applications. It seems parallel jobs are more vulnerable in Grid environment since members (user and resources) in Grid are permitted to modify in runtime.
- Adding people as an interaction entity of VO. Grid-enabled workspace for people collaboration: Desktop tools incorporating all kinds of media, IM, emails, software … etc.
- Semantic support in VO. Semantic description of service and information is an enrichment of Service-Oriented architectures. This significant mechanism will bring great benefits to the scalability and efficiency of resource discovery, service adaptation and service composition. Knowledge-based techniques should be revisited in the dynamic and asymmetric Grid context, how they will be used based on incomplete and partial-correct information, how to minimize the complexity while still get benefit from it?

The new ideas related with the effective evolution of the VOs are consistent with the proposed roadmap steps. Bellow we present some indicative actions:

- Increase interoperability and flexibility
- Transition from user certificates to SAML assertions for user attributes (lean PKI for servers, scalability)
- Considering VO membership/attributes for Grid Scheduling and service orchestration
- Generic business models to support Virtual Organizations and Virtual Value Creation
- Manage the VO life cycle to succeed short lived VOs
- Insert SLAs within VO management

6.4 Make it happen

The identification of the future actions and ideas for the evolution of the VO technology is within the scope of this document. The analysis is distinguished in three phases: three, five and ten year planning.

Now grid study is mainly focused on theoretical researches and lacks successful applications. So, I think that one branch in the roadmap, in three, five and 10 years, is increasing grid practical applications, which means that the grid researches should be application-driven. Based on this idea, virtual organizations (VO) should be chosen as a breakthrough point and be considered significantly. As an application-driven research, the first problem is to select an application background, in which we can conduct researches on VO organization, grid' architecture, grid's platform and so on. We here propose a resource sharing application, which can be considered as an application background for grid study. Currently the Kong Zi College is becoming more and more popular in the world. These collages are setup in different countries, but give similar courses on Chinese culture. Each collage has its own digital teaching resources, like teaching video, teaching software, electric books and so on. The copyrights of these resources belong to different colleges. If we can share these materials, we can save a lot of teaching effort. So we need to setup a resource grid to share these teaching resources. Under this background we should organize a virtual organization(VO) and explore its user/role management, resource management, security management and strategies to produce resources. Based on this background, we should also need to explore grid architectures, resource discovery algorithms, resource replica strategy, scalability, QoS guarantee and so on. In one word, we should do research on the problems we are met when the resource grid is constructed. In a long run, maybe we can reach a pattern which can be used as a routine operation in constructing such grid. As to our work, we have done some researches on education resource grid under the support of NSFC. The background is to explore the key problems in constructing a middle school teaching resource grid.

6.4.1 Three years roadmap

Considering the current limitations and deficiencies of VO technology as well as the feedback provided by the academia we can identify the main activity on the topic for the next three years. Interoperability and increased flexibility of the VO components is of major importance for a generic yet consistent system. The standardization of the basic VO services and the adoption of WS Security, which are within the scope of the three years roadmap, will set the foundations for the efficient operation of VOs across various platforms. Further requests deriving from the business community are related with the introduction of SLAs in VOs and the support of commercial applications. Finally, the need for a better VO management will lead in the optimization of the current tools for Trust and User Access control.

6.4.2 Five years roadmap

The trend and the provisioning for the next five years regarding the evolution of VOs, is to become highly dynamic and easy to manage. In addition, the current burst of Cloud Computing will presumably drift VOs to the creation of Virtual Organizations within Clouds. Furthermore, the legal issues and IPRs will need to be addressed while VOs are gaining ground and expanding in a global scale. The optimization of the VO management components will result in the identification of its lifecycle and prospectively to short lived VOs. In conclusion, the five year outlook contains also the effective integration of legacy systems as well as methodologies for transforming existing organizations into VO-ready organizations.

6.4.3 Ten years roadmap

We need a dynamic, easy and full life-cycle management infrastructure of VO.

- Basic infrastructure to create, deploy the VO. Current VOs lack of self-features for the management of itself. Self-configuration, self-deployment, self-adaptation are important properties for the autonomic administration of VO. This provides great help to the operation of VO.
- Dynamic policy-driven user, resource and trust relationship management. Current VOs make accounting and trust policies according to the identification, this kind of information will not change in the dynamic and large scale Grid context. User access permissions to resources are usually defined by the local system administration or the Grid administration centre, that won't be changed in a relative long time period. Dynamic information like user behaviours is not taken into consideration. As the VOs may evolve over time, to our point of view, this should have great impact on trust relationship definition. For example, a user who often exceeds his resource quota or often incur the crash of the hosting environment should be limited for his access to Grid resources. Just like the social trust system, a person's rights should be defined according to his reputation.
- More efficient and finer-grained QoS control mechanisms. Quality of a VO may include performance, reliability, SLA. So that we think a VO should provide following mechanisms to facilitate fine-grained quality control:
  - Fine-grain resource and job monitoring. The information of resource performance/cost, availability, job submission, job queuing may be needed for quality control.
  - User history profiles and resource usage tracks. The information may give important hints for Grid workload description, usage pattern and resource assumption pattern.
  - High efficient information retrieval utilities. Past and current system status is critical to decide future actions; it is a fundamental underlying service for Grid platform. Many modules will get information from it. So it is important to provide a high performance information retrieval service, and also some semantic info may be used to locate a term with higher correct.
  - Job scheduling strategies taking into consideration load balance, user reputation, resource reliability and so on.
  - Failure detection and fault tolerance framework. Failures in VO should be detected in time, and provide applications with flexible and practical recovery and fault

tolerant mechanisms, such as reboot, process migration, multi-version processes, etc.

Enabling VO across different platforms and infrastructure

- Interoperability between different VOs. Grid applications and services may access resources belonged to multiple management domains and hosting environments. That may incur policy level interoperability problems. Different Grid services and resources may require various certificates and access control policies, thus, the VO should record application session information and transfer roles between different sessions.
- Single system image of different VO information, resources, users, and so on. Different VOs may have different users and resources registration services, making them seeing and understanding each other is a problem of interoperability. It may require some standards to describe the information of resources, users profile, policies, and some credential data that are involved during an interoperation process.

From the second workshop celebrated in Athens a few issues regarding the business models to support Virtual Organisation and Virtual Value creation:

- Research on the inter-organisational process layer for business services collaboration
- Change in collaborations and content of communication in Virtual Organisations and effects to Grid technologies
- To what extent can task-related networks substitute for personal relationships as inter-firm co-ordination mechanisms?
- Will formalised, programmable communication replace more informal, customised communication?
- Building and maintaining trust in virtual teams
- Organisational identity and cohesion in virtual organisations
- Legal Issues and IPR in virtual organisations

Future directions of VO can be summarized as follows:

- Dynamic VOs
- Short Lived Credential Service
- International Grid Trust Federation GridPMA
- Introducing SLAs in VOs: Guarantees
    - SLAs between the actors
        - User and VO
        - VO and Resource Provider
        - User and Resource Management Systems
- Current concepts resource usage oriented
    - Need to increase of granularity: resource, application, application feature, data bases, services
- Considering VO membership/attributes for Grid Scheduling and service orchestration
- Considering VO membership/attributes for commercial applications:
    - license issues

- Transition from user certificates to SAML assertions for user attributes (lean PKI for servers, scalability)

6.5 How it is covered in existing EC and Chinese R&D Programmes

Many projects from EC and China are involved in the field of Virtual Organisations, and a sample is summarized below.

6.5.1 EU Work Programme Initiative Status

In EU, there are a number of different initiatives related with the evolution and development of VOs. Bellow we present a partial list of projects and Working Groups on VO in FP6 or before: Akogrimo, Daidalos, EGEE, HPC4U, NextGRID,EU-Provenance, SIMDAT, TrustCoM, UniGridS, InteliGrid, CoreGRID, Grid4ALL, BEinGRID, XtreemOS, TG8 - Virtual Organisations, etc. In the FP7 work programme 2007-08 includes VO related topics, ICT-2007.1.2, ICT-2007.1.4, focusing on “service and software architectures, infrastructures and engineering”, and “Secure, dependable and trusted infrastructure”.

6.5.2 China Work Programme Initiative Status

In China, the National High-tech R&D Programme (863) and the National Basic Research Program of China (973 program), sponsored by the Ministry of Science and Technology (MOST) of China, and the National Natural Science Foundation of China (NSFC), provide government funding for VO development and operation. The virtual organization (Agora) in CNGrid is sponsored from the key research project “High performance computers and software” in 863. The guideline of 863 project proposal mentions VO research as follows: “Provide general support for various applications and resources, such as virtualization and service mechanisms, naming, resource and service access, dynamic service management, batch and interactive job scheduling, Grid user management, Grid resource information service, Grid security.”

6.6 Recommendations for EC and China

As making a roadmap for future 3, 5 and 10 years, we suggest that priorities be given to VO research themes presented above.

For the first 3 year period, a clear definition of VO framework on the functionalities and management infrastructures should be proposed. This may involve the discussion from Europe and China academy, some standards and interfaces may be proposed during this period. It is important to involve related industries to the discussion, they will provide requirement from real end-users, and help with setting up the business model.

For the middle-term aim, advanced functionalities of VO except for semantic support, cross platform VO and high efficient VO should be provided to the users.

For the 10 year goal, semantic support in VO should be realized, and relevant standards should be raised. Also, autonomic capabilities for full life-cycle management of VO should be provided.

6.7 References to the elements that lead us to the conclusions and roadmap

TrustCoM →www.eu-trustcom.eu

7 COMPONENT MODEL

7.1 Vision

It has been widely accepted that the component-based software development is an effective means to tackle the complexity of modern software, such as software to support Grid applications. With the evolution of internetworking and pervasive computation, it can be anticipated that in the near future, it is likely that we will witness large scale interactions between components developed by different parties across the Internet. It is improbable that component developers will be able to develop such kind of internet-scale applications with today's component models and tools, which are mostly stemmed from the enterprise computing environments of the near past.

If we could efficiently handle the predicted complexity derived from the large-scale interactions between future Grid applications, we could conclude to the following visionary ideas that may be proved useful as a guide in the technological advance of this area:

1. Produce component model to support autonomic software development so as to improve adaptation and reusability at a high level.
2. Develop a unified component model that will fit the needs of future industrial use cases as well as those of today's Grid applications.
3. Clearer separation of concerns regarding the functional and non-functional requirements to ease the maintenance and the flexibility.
4. Standardize software certification models (such as the testing model for Grid software) at the component model level in order to improve software quality assurance.
5. The construction of reliable and scalable software systems require better behavioural guarantees. Elements towards this goal include: more expressive specification formalisms, development environment providing "correct by construction" code, dynamic adaptation techniques, etc.
6. Ensure QoS at the component level which will allow component applications to provide support SLA requirements.

7.2 State of the Art & Limitations

7.2.1 State of the Art

Reusability has been improved with component models originating from industry (e.g. EJB, CCM, etc.) as well as from academia (e.g. Fractal, GCM, SCA, etc.). Fractal is an extensible component model proposed by ObjectWeb that can be used with various programming languages to design, implement, deploy and reconfigure various systems and applications. Fractal forms the basis of the GCM (Grid Component Model) currently being defined within European project CoreGRID. GCM supports hierarchical component composition, collective and data/steam ports, automatic management of notable composition components etc. The essential characteristics proposed by the GCM include structure hierarchy, model extensibility, reflection, adaptivity and interoperability. In recent years, service oriented computing and component model are becoming integrated. SCA(Service Component Architecture) flexibly incorporates reusable components in SOA programming style. VGE-CCA is a distributed CCA (Common Component Architecture) framework to apply port-based composition to Grid application Web services. It provides an extensible API for basic CCA constructs and specialized components, and separates applications composition from runtime control flow steering. Other well-known component models include Koala, SOFA, Kobra, PECOS, etc.

In China, component model and related issues have been studied for the past decades. Project StarCCM and PKUAS, both being funded by National 863 high Tech Program, implemented CCM and EJB respectively but not been limited to them. A series of enhanced research work, such as dynamic composition and runtime adaptation, have been conducted. ABC (Architecture-based Component Composition) approach proposed by Peking University uses software architecture-based technology such as ADL to facilitate the composition of applications. Its component model consists of external interfaces and internal specification. A light-weight component model equipped with autonomic ability, so called autonomic elements, have been proposed in iVCE project for internet virtual computing environment.

7.2.2 Limitations

Some work has been done regarding the integration of various component models and tools. For examples, effort has been focused on integrating PKUAS and Jonas (Chinese component model), while ProActive/GCM has been used in Plugtests between European and Chinese Grid platforms. With respect to SCA, some approaches have been tried to integrate component ideas into Service Oriented Architectures. However, such a platform does not yet feature the power of full-fledged components with, for instance, the capacity to impose -when needed- the interoperable and flexible deployment of components upon execution.

Besides the improvement above, it is clear that academia and industry are still lacking mechanisms to support adaptation and reconfiguration of components during execution, based on the dynamic event occurring in real-time; for instance machine failure or overload. Moreover, we are still missing the mechanisms to take into account application level requirements, such as Service Level Agreements, in order to improve contracted Quality of Services.

Some of the component models are too complex to be widely adopted by the industry and domain users. Part of the problem is the lack of tools to assemble component configuration, to deploy components onto various platforms, and to monitor them at execution.

The difficulties to identify the basic requirements for a given component in order to dynamically select the right set of machines on which they can be executed are yet another problem that prevents components from being used in SOA. Such functionality will require a unified model to describe application needs, which shall be valid for most of the Grid application. In turn, this will require a standardization of applications description in order for them to really be promoted at the level of reusable components.

Finally, as in practice we are dealing with various component models, in various languages, there would be a need for interoperability in component management at execution (life cycle management, reconfiguration, migration from one machine to another). A challenge is to achieve a full differentiation of functional and non-functional part of the codes, in order to ease the maintenance and the flexibility. Overall, we believe we must achieve a full integration of components into Service Oriented Architecture, with the capacity to monitor and dynamically maintain the Quality of Services at execution.

7.3 New ideas

One idea is to develop a new light-weight component model for the autonomic support in Grid applications. It is widely accepted that a few component models tend to be over-weighted for developing autonomic software entities. But in some applications, not all the properties are needed as well. If the component model itself can be configurable or reconfigurable according to the development contexts, developers can have much more freedom in choosing those real useful functionalities. Therefore, new light-weight component model is needed to support the modeling of autonomy of software. The main research topic in this area is how to implement the semantics of autonomic support in a component-based way. Or, put it in another way, what are the necessities a component model should have in order to fulfill the basic functionalities of an autonomic software entity. We argue that this is important and feasible because (1) there are many component models available but they are targeted for different domains; and (2) the component model should be kept light-weight enough and at the same time sufficient to support the dynamic nature of Grid application.

Component-based paradigm has been one of the dominant programming paradigms for its modularity, reuse, and runtime flexibility. Another future idea is to intersect Grid component model research with other software engineering frontier technologies, such as Service-Oriented Computing, Aspect-Oriented Programming (AOP) and MDA. (1) The gap between component technologies and Service Oriented architecture (SOA) have to be bridged to achieve code and service composition for flexibility, agility, etc while keeping the benefits of the component programming model. (2) Focusing on various aspects of Grid environments, there are system characteristics such as user interfaces, component persistency and distribution, collaborative work and end user configuration support that can be found everywhere. Component developers tend to use aspects to describe component requirements, and use aspects to guide component implementation as well. With the development of aspect-oriented programming methods, it can be anticipated that AOP will contribute to tackle the decoupling of functional and non-functional codes within components. Previous research work relating to component-based systems engineering could be a very good start towards this research direction. (3) In order to make components “talk” to each other in a scalable way, platform independent models are necessary. There were/are a lot of standardization institutions and work relating to component-based software engineering in enterprise computing context, such as the OMG's PIM/PSM models in MDA. The practice of future Grid component standardization may benefit from them.

CM design has a direct impact on the development phase. The final product can get many advantages by good CM design that may have aspects that ensure that the software produced is high quality software and is characterized by some unique features that the architect demands (e.g. security). Therefore, the CM design is an important step towards the need of high quality Grid Software. CM can act as a Quality Assurance mechanism, but this is true only if a standard shared and accepted model is reached and it is well applied. A good idea could be to propose/develop something similar to GRID-QCM that is oriented to test the result of the design phase by automatable metrics. This can be done only having a really strong and accepted standardization of the CM so as to find really generally applicable metrics.

"Components for Grids", Marco introduces the components model research in CoreGRID project and basically shows that the draft roadmap gives a correct vision on component models for grids. By analyzing the difference between general software and Grid software, he points out the key aspects in Grid software development and the desiderata from the view of components for grids. He emphasizes on the role of autonomic domain that will make significant contributions to performance tuning, fault tolerance and security of grids. Then, GCM, the Grid Component Model by CoreGRID project, is briefly introduced, such as GCM autonomic managers and security management and so on. To show the vision, Marco compares GCM with SCA (Service Component Architecture) and gives the ongoing SCA/GCM activity. He presents that autonomic components and deployable autonomic services are the directions for the future encapsulation in grids. Finally, Marco concludes that autonomic management and services must be the focus in the roadmap.

In his talk entitled "Component Framework and Programming Approaches for Application Web Services", Rainer presents the VGE-CCA Component Framework and the Planets Interoperability Framework. VGE-CCA is a distributed CCA (Common Component Architecture) framework to apply port-based composition to Grid application Web services. It provides an extensible API for basic CCA constructs and specialized components, and separates applications composition from runtime control flow steering. Then, Rainer introduces Planets Project for integrated services for digital preservation. He gives the detail on Planets interoperability framework including its architecture, components model, programming and execution. In the conclusion, it is pointed out that bridging SOA and component technologies is important to composition and dynamic resource selection in grids.

7.4 Make it happen

New light-weight component model for the autonomic support in Grid applications
It is widely accepted that a few component models tend to be over-weighted for developing autonomic software entities. But in some of the applications, not all the properties are

needed as well. If the component model itself can be configurable or reconfigurable according to the development contexts, developers can have much more freedom in choosing those real useful functionalities.

New light-weight component model is needed to give more support on the modeling of autonomy of software. This component model can be derived from existing models such as CCM but must be light-weighted to preserve only the functionalities which are of necessity.

The main research topic in this area is on how to implement the semantics of autonomic support in a component-based way. Or, put it in another way, what are the necessities a component model should have in order to fulfill the basic functionalities an autonomic software entity. We argue that this is important and feasible because (1) there are many component models available but they are targeted for different domains; and (2) the component model should be kept light-weight enough and at the same time sufficient to support the dynamic nature of Grid application.

Component model supporting service composition
Extend component model, like GCM, to provide a programming framework fitted for both scientific and business applications. Also, this will have to bridge the gap between component technologies and Service Oriented architecture (SOA) in order to achieve code and service composition for flexibility, agility, etc while keeping the benefits of the component programming model.

Standardization of platform independent models to let the different component models to "talk" to each other
Component-based paradigm has been one of the dominant programming paradigms for its modularity, reuse, and runtime flexibility. Quite a lot of research initiatives and projects have been launched on this topic and has achieved many advantages including the GCM, the Fractal, and the GridCCM component models, to name a few. In order to make them "talk" to each other in a scalable way, platform independent models are necessary. There were/are a lot of standardization institutions and works relating to component-based software engineering in an enterprise computing context, such as the OMG's PIM/PSM models in MDA. The practice of future Grid component standardization may benefit from them.

Aspect-oriented programming combined with component model
Focusing on various aspects of Grid environments, there are system characteristics such as user interfaces, component persistency and distribution, collaborative work and end user configuration support that can be found everywhere. Component developers tend to use aspects to describe component requirements, and use aspects to guide component implementation as well. With the development of aspect-oriented programming methods, it can be anticipated that AOP will contribute to tackle the coupling of functional and non-functional codes within components. Previous research works relating to component-based systems engineering could be a very good start towards this research direction.

Quality Assurance for Component Model design

What about the component model? The component model seems to be a very good way to develop Grid software. What is the connection between it and the quality assurance of the final product?

CM design has a direct impact on the development phase. So the final product can get many advantages by good CM design that may have aspects that ensure that the software produced is high quality software and is characterized by some unique features that the architect demands (e.g. security).

So the CM design is an important step towards the need of high quality Grid Software. CM can act as a Quality Assurance mechanism, but this is true only if a standard shared and accepted model is reached and it is well applied. A good idea could be to propose/develop something similar to GRID-QCM that is oriented to test the result of the design phase by automatable metrics. This can be done only having a really strong and accepted standardisation of the CM so as to find really generally applicable metrics.

Following this path we can reach the goal to make CM a certified way in the production of good Grid software.

The next 3 years could be a good time to standardise a common way to project CM designed software. The next 5 years should be enough to define a quality certification model for CM design. This could be added as integration of existing models (e.g. GridQCM) or a standalone model. Finally, it is expected that in the next 10 years it could be made possible to model a standard for the Grid community.

7.4.1 Three years roadmap

A light-weight component model should be proposed with its reference implementations in widely-accepted language. This component model can be derived from existing models such as CCM but must be light-weighted to preserve only the functionalities which are necessary for the modeling of autonomy of Grid applications. Besides, we should standardize a common way to project CM designed software to facilitate future work of quality assurance for component model design.

7.4.2 Five years roadmap

Most of the research activities dealing with extending light-weight component model to provide a programming framework fitted for both Grid-based scientific and business applications. Service-Oriented Architecture and Aspect-Oriented Programming should be integrated to the component model as optional features. A quality certification model for CM design will be defined, which can be added as integration of existing models (e.g. GridQCM) or a standalone model

7.4.3 Ten years roadmap

Platform independent models will be defined to ensure interoperability among component models in a scalable way. Modeling a standard of quality assurance for CM design is necessary, and its mapping on the light-weight component model will be a reference implementation. As a result, it could be made possible to model a standard for the Grid community.

7.5 How it is covered in existing EC & China R&D Programmes

7.5.1 EU Work Programme Initiative Status

A number of projects are conducting research on component models under the Framework Programme, such as

- CoreGRID FP6
In CoreGRID, the institute on programming models aims to deliver a definition of a lightweight, component programming model that can be usefully exploited to design, implement and run Grid applications.

- GridCOMP FP6
GridCOMP main goal is the design and implementation of a component based framework suitable to support the development of efficient Grid applications. It provides the reference implementation of the GCM defined in CoreGRID.

7.5.2 China Work Programme Initiative Status

The National Grand Basic Research and Development Program of China (973) and National High-tech R&D Programme (863), sponsored by the Ministry of Science and Technology (MOST) of China, have supports on the research activities of component technology for future computing environment. Sponsored by 973, researchers in PKU and NJU and other partner universities have developed an architecture centric technical framework for the definition, incarnation and engineering of Internetware. Internetware entities will be packaged as components, behaving as agents, interoperating as services, collaborating in a structured and on demand manner. The project has obtained the successive support for next 5 years (2008-2013) from MOST. MOST also supports 973 iVCE project (2005-2010) led by NUDT, which develops the on-demand aggregation and autonomic collaboration of resources packed as autonomic elements over Internet. With the support of 863 projects, the component technologies are also studied in the context of pervasive computing in China.

7.6 Recommendations for EC and China

Based on the above analysis, we would like to present the following recommendations.

(1) Clarify the relationships among component-based and service-based models, environments and tools, in particular concerning the possibility to migrate the innovative component features of Grid component model (such as GCM) to WS/SOA platform (such as X-Service)

(2) Both the EC and China need to put together research groups that will tackle these important issues.

(a) New component models capable of supporting new properties of future Grid software, such as autonomic execution, dynamic roles, adaptive behaviors', QoS etc;

(b) Collaborations should be funded to speed-up the development of programming framework fitted for both scientific and business applications;

(c) Interoperability of different component models, at different layers.

7.7 References to the elements that lead us to the conclusions and roadmap

- Component reliability extensions for Fractal component model. http://kraken.cs.cas.cz/ft/public/public index.phtml.
- T. Barros, L. Henrio, and E. Madelaine. Behavioural models for distributed components. In FACS'05 Workshop, Macao, 2005. LNCS , Spinger Verlag.
- Dream project. http://dream.objectweb.org
- Fractal specification. http://fractal.objectweb.org/specification/index.html
- Fractal ADL. http://fractal.objectweb.org/fractaladl/index.html
- Pierre-Charles David and Thomas Ledoux. Towards a framework for selfadaptive component-based applications. In Jean-Bernard Stefani, Isabelle Demeure, and Daniel Hagimont, editors, Proceedings of Distributed Applications and Interoperable Systems 2003, the 4th IFIP WG6.1 International Conference, DAIS 2003, volume 2893 of Lecture Notes in Computer Science, pages 1–14, Paris, November 2003. Federated Conferences, Springer-Verlag.
- Alexandre Denis, Christian Perez, Thierry Priol, and Andr´e Ribes. Bringing high performance to the corba component model. In SIAM Conference on Parallel Processing for Scientific Computing, 2004.
- Jan D¨unnweber and Sergei Gorlatch. HOC-SA: A Grid Service architecture for Higher-Order Components. In International Conference on Services Computing, Shanghai, China. IEEE Computer Society Press, September 2004. ISBN 0-7695-2225-4.
- H. Fakih, N. Bouraqadi, and L. Duchien. Aspects and software components: A case study of the fractal component model. In International Workshop on Aspect-Oriented Software Development (WAOSD 2004), Beijing, China, September 2004.
- P. Parizek and F. Plasil. Model checking of software components: Combining java pathfinder and behavior protocol model checker. In Proceedings of 30th IEEE/NASA

Software Engineering Workshop (SEW-30). IEEE Computer Press, april 2006. to appear.

8 WORKFLOW – BUSINESS PROCESS

8.1 Vision

It is well-known that Grid workflow/business process is a critical part of the Grid environments, which captures the linkage of constituent services together in a hierarchical fashion to build larger composite services. It can enable collaborations between large-scale resources to support QoS-aware, secure guaranteed, optimized complex scientific applications and various business applications in the short-term vision.

For the moment, there are over twenty kinds of Grid workflow systems, and each is based on specific workflow description language and execution enviroment. Within five years, it is urgent to decouple the workflow systems with their specific Grid environments, and various legacy workflows can be incorporated despite the discriminations, for reducing costs as well as enabling dynamic collaboration and boundary-crossing problem solving.

Hopefully, in the long term, the business process decision-makers can use simulations seamlessly without problems supporting the high-level decision analysis. And this can be achieved by modeling the performance of detailed problem features by workflows and the integration of these workflows in a hierarchy of abstraction levels step by step, providing advanced workflow composition, execution and debugging tools support this integration at each abstraction level, and also matching the real-time events driven by the workflow with the simulation level parameters.

8.2 State of the Art & Limitations

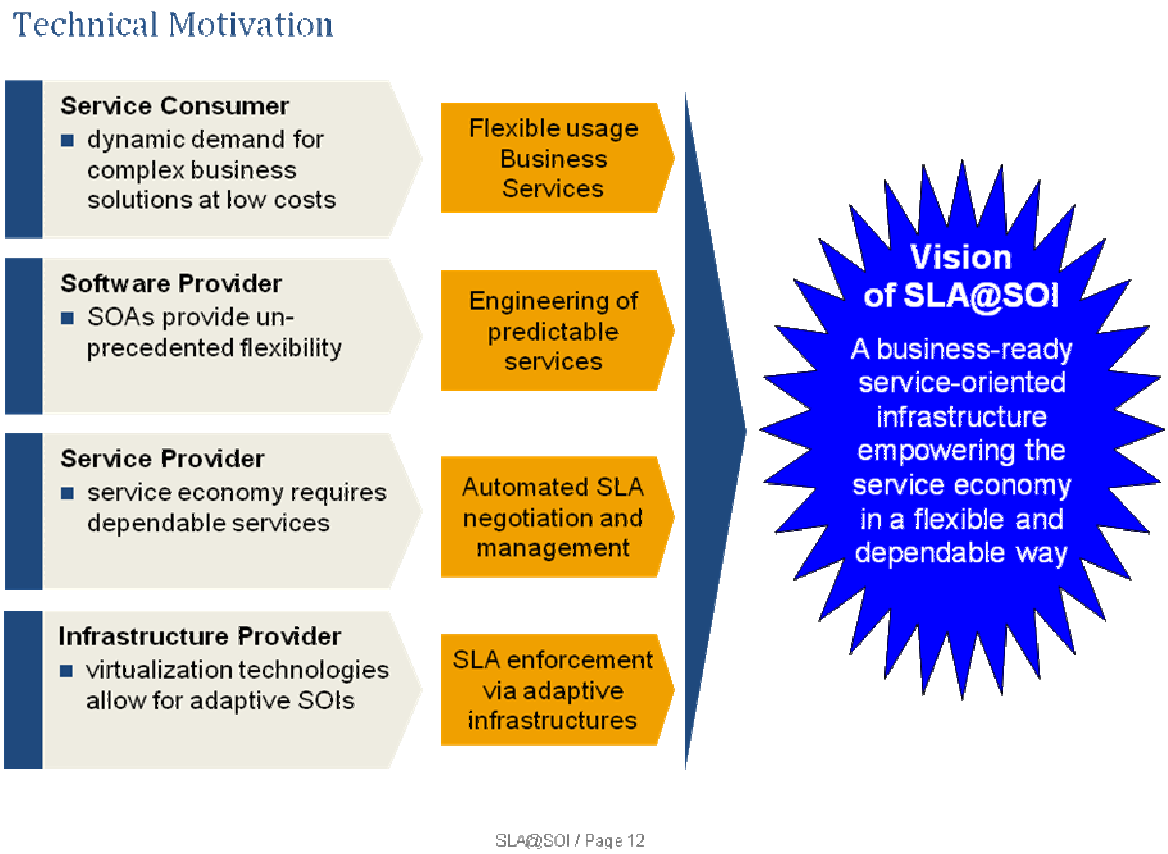
8.2.1 State of the Art

Some of research and engineering work has already taken place in related projects of both EU and China to achieve above visions. Typical research work and engineering practice involved Grid workflow/business process, funded by Ministry of Science & Technology (MOST), National Natural Science Foundation (NSFC) in China and vendors, are CNGrid GOS Grid Workflow, VINCA, CROWN and XServices Workflow, Protein Analysis Scientific Workflow, InforFlow etc.

At present, several workflow systems exist in the CNGrid, CROWN and XServices, such as BPEL4WS engines, JSDL processing engines, and others. From the perspective of interoperability, these legacy workflow systems should be candidates to be included into a Grid as a specific kind of resources providing orchestration capabilities, for lowering costs as well as enabling dynamic collaboration and boundary-crossing problem solving. GOS Grid Workflow extends previous work VINCA to a meta-workflow system, so that other workflow capabilities in several business and scientific communities can be incorporated. To this end, a concept model is proposed, in which the capabilities of legacy workflow applications and the enactment engines are virtualized as abstract services that can be further used to construct applications from business level. Crown workflow provides a suite of service based and graphic workflow modeling and executing environment based on WS_BPEL. it enables user to orchestrate services from Crown Grid nodes in the form of workflow in a visible developing environment and monitor the execution state in a browser. XService Workflow is a workflow management system for web services based workflow. It is composed of a set of tools which provide functions including modelling the workflow (XService Workflow Designer), running the workflow (XService Workflow Engine) and monitoring and managing the process (XService Workflow Manager). Protein Analysis Scientific Workflow project's main purpose is to deal with the distributed, dynamic Grid workflow collaboration problem. Not only can it provide business workflow function, but also deal with the data produce in the scientific workflow. InforFlow provides the Human workflow development, integrated workflow development, the development tools, management tools and operating environment support for the customers. Meanwhile, it combines with Web2.0 technology to package different service of the workflow engine into different widgets, and this function will support customers to add new workflow function in their system.

Correspondingly, the main representatives of workflow/business process evolvement in Europe are SIMDAT, BRIDGE, PoSSIS, ASKALON, etc. In SIMDAT and BRIDGE research on Workflows in the context of Grid is a major topic. As part of SIMDAT and BRIDGE or in close relation with the work of SIMDAT, the following workflow tools have been enhanced and partially Grid enabled: TAVERNA (IT Innovation), OPTIMUS (Noesis), Modelcenter, isight and KDE (Inforsense). Interoperability in the Grid contexts between these workflow tools have been demonstrated by several distributed workflows involving several tools in particular for applications in the development of airplanes. Initial work was performed on an advisor tools for workflow generation. The same cases are also in the PoSSIS, ASKALON and other projects.

SLA@SOI presented in last Shenzhen conference the following self-explanatory overview regarding the business vision and the QoS that the future Grid management will have to have.



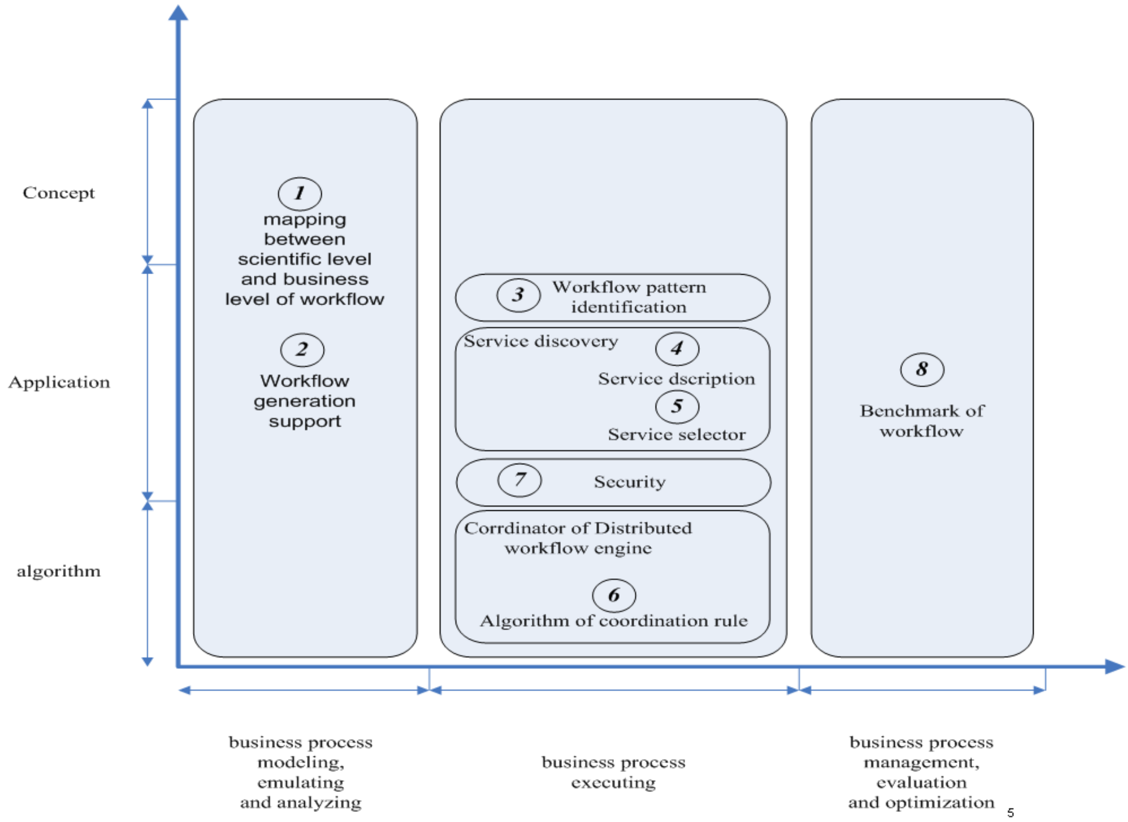
The dynamic composition of services by the means of SLA is considered a crucial component for a vivid service economy. SLA@SOI specifically considers the need of a multi-layered SLA management framework which supports the inclusion of different business layers ranging from technical provisioning of software, infrastructures to governance, sales and marketing. The different services will need to be bundled in workflows to unleash the full functionalities to consumers. The use of SLAs provides the basis to include all necessary constraints.

8.2.2 Limitations

Nevertheless, although current research and engineering progress have achieved important outcomes and many workflow/business process engines and tools occur now, there are still some remarkable limitations to be mentioned. First, the gap between the business process decision-maker and the application developer is obvious and wide. Secondly, it seems to be very difficult to precisely compare and evaluate the performance of different workflow engines; moreover, it is even hardly to improve the performance of execution of designed workflow engines. Thirdly, QoS-based resource scheduling, service selection and service composition are less considered, and the co-ordination of different distributed workflow engines needs to be considered and designed in more efficiency. Finally, security is still a vital element here, which needs to be well provided by all the workflow/business process engines, but few existed engines achieve this maturely now.

8.3 New ideas

From the state of art and limitations summarized above, it is seemed that there will be a long way and difficult to achieve the vision. Luckily, at the same time, as research and engineering practices on workflow/business process become in-depth, new ideas for solving these limitations and challenges occur gradually. In the figure below, it will orderly explain these new ideas from two directions. One direction is from the algorithm, application and concept levels, while the other is through the fields of business process modelling/emulating/analyzing, executing and management/evaluating/ optimization. Each idea is corresponding to a point of intersection (x, y) from two directions.



5

① **Mapping between scientific and business levels of workflows**: The gap between the business process decision-maker and the application developer is wide. While business decision maker usually use BPEL for the description of workflows, each engineering or scientific discipline has its own workflow language and established tools. BPEL itself lacks certain structural elements like loops, which are essential for optimization workflows, for example, to implement the vision for improving the decision making processes by including simulation related workflows into the decision process the gap between several abstraction levels have to be closed by support for seamless inclusion of low level workflows into high-level decision support workflows.

② **Workflow generation support**: The generation of workflows is a time consuming tasks, which need experiences about the workflow tool as well as the application domain. On the other hand, companies will develop and archive a huge number of workflows for different purposes. By enhancing the workflows with appropriate semantic information it will be possible to simplify the generation of workflows and make usage of the workflow repository. For a new problem the improved workflow environment will allow to find a set of workflows, which have been used already in a similar context. For prototypes of workflows, it will be possible to find similar ones, which might be a source for modifications or even alternatives. In a second step, the semantic information for workflows can be generated automatically from the workflow itself as well as from the actual usage of the workflows.

③ **Precisely identifying the pattern of the workflow**: As the different pattern of workflow (concurrent or sequential task), the performance of workflow engine is different. If the pattern of workflow is orderly, the engine execute the workflow is better. If the pattern of workflow is concurrent, engine compiles the BPEL to the middle language for high concurrent pipelines is better than sequential run. So identifying the pattern of business process will develop within pattern recognition evolution in the future.

④ **Improving the performance of semantic tagging of service description**: Service description language will contain the QoS and ontology information of the service. For the reason of performance of semantic tagging, before the add the semantic tag to the WSDL, create an inverted-index to search a similar WSDL set and tag them with semantic information, the methods which mixes the similarity search and semantic matching process.

⑤ **QoS and resource scheduling**: Selecting suitable service with global constraints by supporting flexible optimization objectives. Context information (including context of different engines) is focused on the services composition, respecting the global constraints, service selector choose the suitable solution reducing the re-optimization overhead to compose an optimal process from candidate service sets. Different usage scenarios expose varying cost functions and need for individual optimization algorithms during resource selection and scheduling.

⑥ **Coordinating the distributed workflows**: The coordinator cooperate the different engines according and follow the coordinate rules. Fuzzy logic of expert system affects the reasoning process, and now, Rete II algorithm of Forward-Chaining is applied in rules decision because considering the expected efficiency, the methods of Backward-Chaining in induction will create a precise pattern matcher in next years.

⑦ **Workflow Security Support**: Due to the security requirements in workflow and business process, it should provide special security services with different security level for such kind of applications. Moreover, a fine-grained and extensible security framework, trust federation and negotiation for resource sharing and collaboration are also needed to apply now and in the near future.

⑧ **Evaluating the workflow engine**: The Benchmark of workflow engine evaluates the performance of workflow engine. In the business process, the environment is often change in every time (such as the workflow of Grid, no a killer benchmark for evaluating the resource of VO (Visual Organization)), so the fair property of benchmark must be considered, the consumer must select the suitable workflow engine under the fair judgment methods.

From SLA@SOI the main innovations would pivot in the following issues, namely:

- **SLA management framework**
  - harmonizing perspectives of relevant stakeholders (software/service/infrastructure provider and customer)
  - standards for SLA specification and negotiation
  - providing a ystematic multi-layer SLA management (planning, optimization, and provisioning), monitoring and accounting
  - guaranteed QoS in a dynamic and end-to-end fashion via consistent SLA handling across IT stack
- **adaptive SLA-aware infrastructures**
  - ability to monitor SLA and infrastructure state and apply corrective measure according to predefined policies.
  - standardized interfaces for adaptive infrastructures with harmonized access to different virtualization technologies.
  - advanced technologies for SLA enforcement on infrastructure level
  - efficient resource usage w/ reliable SLA enforcement at infrastructure level
- **engineering methods for predictable service-oriented systems**
  - modelling techniques and prediction tools for SOA and SOI components
  - support for design-time and run-time application analysis for performance predictions
- **business management suite for e-contracting** covers complete business lifecycle of a service provisioning/delivery

8.4 Make it happen

From the state of art and limitations summarized above, it is seemed that there will be a long way and difficult to achieve the vision. Luckily, at the same time, as research and engineering practices on workflow/business process become in-depth, new ideas for solving these limitations and challenges occur gradually. In the figure below, it will orderly explain these new ideas from two directions. One direction is from the application and concept levels, while the other is through the fields of business process modeling/emulating/analyzing, executing and management/evaluating/ optimization. Each idea is corresponding to a point of intersection (x, y) from two directions. Mapping between scientific and business levels of workflows: The gap between the business process decision-maker and the application developer need diminish. A new mapping process translates the business process to a scientific formalized describing process.
The generation of workflows is a time consuming tasks, which need experiences about the workflow tool as well as the application domain. On the other hand, companies will develop and archive a huge number of workflows for different purposes. By enhancing the workflows with appropriate semantic information it will be possible to simplify the generation of workflows and make usage of the workflow repository. For a new problem the improved workflow environment will allow to find a set of workflows, which have been used already in a similar context. For prototypes of workflows, it will be possible to find similar ones, which might be a source for modifications or even alternatives. In a second step, the semantic information for workflows can be generated automatically from the workflow itself as well as from the actual usage of the workflows. Precisely identifying the pattern of the workflow: The execution of workflow engine is mainly based on the thread mechanism which is provided by underlying system such as OS or JVM etc. The execution of workflow process can be categorized into two different types: Interpret the tasks defined in the workflow process in a single thread is the widely adopted approach which is suitable for sequential task execution and easy to control the concurrent execution, but this approach may behave badly if the process itself is highly concurrent; Multiple thread based execution mechanism may solve the perform problem, but the tradeoff is the complexity of concurrent execution control. Actor mode is a share-nothing high concurrency support programming paradigm which is suitable for the construction of high concurrency workflow engine and can easily support sequentially and concurrently task execution. Many examples have demonstrated the successful of Actor mode such as Erlang, Scala etc. But as the different pattern of workflow (concurrent or sequential task), the performance of workflow engine is different. If the pattern of workflow is orderly, the engine execute the workflow sequentially is perhaps better; if the pattern of workflow is concurrent, engine compiles the BPEL to the middle language for high concurrent pipelines, i.e. an Actor mode compatible middle language, is better than sequential run. So identifying the pattern of business process will develop within pattern recognition evolution in the future. ④ Services of the workflow discovering and selecting process: Service description language will contain the QoS and ontology information of the service. For the reason of performance of semantic tagging, before the add the semantic tag to the WSDL, create an inverted-index to search a similar WSDL set and tag them with semantic information, the methods which mixes the similarity search and semantic matching process. Respecting the global constraints, service selector chooses the

suitable solution reducing the re-optimization overhead to compose an optimal process from the candidate service sets. Coordinating the distributed workflows:  Due to the complex interaction pattern between workflows, we are opinion of that traditional message-passing method like RPC or Web service couldn’t satisfy requirements of coordination of workflows. We suggest taking into account event-driven methods. Through making full use of achievements in the field of event streams (data streams) management system, on the one side, we could initiate more complex interactions between workflows. On the other side, benefited from the property of real-time in event streams (data streams) management system, efficient performance is also guaranteed. Since workflow systems may derive from different fields or domains, the differences of interfaces may exist. Coordination forces any workflow to understand all other interfaces exposed by other workflows, and that is impossible or at least very difficult in the large-scale environment especially for dynamically joined or changed workflow system. So we suggest adopting the method of ontology to express corresponding domain conceptions. Combined with event-driven structure, workflows could coordinate each other more flexibly and naturally. Workflow Security Support: Due to the security requirements in workflow and business process, it should provide special security services with different security level for such kind of applications. Moreover, a fine-grained and extensible security framework, trust federation and negotiation for resource sharing and collaboration are also needed to apply now and in the near future. Evaluating the workflow engine: The Benchmark of workflow engine evaluates the performance of workflow engine. In the business process, the environment is often change in every time (such as the workflow of Grid, no a killer benchmark for evaluating the resource of VO (Visual Organization)), so the fair property of benchmark must be considered, the consumer must select the suitable workflow engine under the fair judgment methods.

8.4.1 Three years roadmap

Although current research and engineering progress have achieved important outcomes and many workflow/business process engines and tools occur now，in the short term, there are still some remarkable limitations and challenges of Grid workflow technologies, such as the gab between the business process decision-maker and the application developer, evaluating the workflow engine, QoS and resource scheduling, workflow security support, etc. A particular emphasis has to be made to overcome the limitations and deficiency of current technologies and policy.

8.4.2 Five years roadmap

From the perspective of the medium-term, we have to face the problem of standardization of Grid workflow/business process engines, which will promote the integration, coordination and inter-operation among workflow systems and legacy workflows. Standardization will play a crucial role in decouple the workflow systems with their specific Grid environments, and various legacy workflows can be incorporated despite the discriminations, for reducing costs as well as enabling dynamic collaboration and boundary-crossing problem solving. It seems that high-level of abstractions are required but the poor efficiency come with it must be carefully considered. All these technologies, policies and standards must be verified in real applications.

8.4.3 Ten years roadmap

In the long term, the business process decision-makers can use simulations seamlessly without problems supporting the high-level decision analysis. Grid workflow/business process engines will be widely used in the production environment of research and business, and become one of the mainstream development environments and methods.

8.5 How it is covered in existing EC and Chinese R&D Programmes

In general, both China and EU are aware of strategic essentiality to the research of workflow and business process. Up to now, they have already set down some Work Programmes and funded several projects for achieving short-term and middle-term vision.

8.5.1 EU Work Programme Initiative Status

Within the European Union, according to the research priorities highlights of NESSI during 2009-2010, Research area of business process modelling is included. To meet the demands of paving the way towards the collaborative executable enterprises, the researchers will focus most of their attention on how to support dynamic formation, formalization management and interaction of business processes implemented through services, and how to provide long-term transactional business collaboration. In the Service and software architectures, infrastructures and engineering Theme, ICT-1-1.2 of FP7, Project NEXOF - Reference Architecture (FP7-216446)'s WP2 - Service-centric systems engineering, is to define and describe architectural components for creating, describing, managing services and composing them into workflows in order to meet specified business requirements. This is an exemplification to show that, fulfilling the projects (e.g. TRUSTCOM, GRIA, NESSI-Grid, BEinGRID, etc.) of FP6, EC will continue to supporting the research of workflow and business process in FP7.

8.5.2 China Work Programme Initiative Status

The National High-tech R&D Programme (863), sponsored by the Ministry of Science and Technology (MOST) of China and National Natural Science Foundation of China (NSFC) are the two major sources of government funding on Workflow and business process research in China. Many efforts have been put into research projects. As mentioned above, CNGrid GOS Grid workflow, VINCA and VINCA4Science, etc., are all illustrations for this issue. For the current 5-year programme (2006-2010) of 863, there are a number of projects for supporting to reach the short-term and middle-term vision. Workflow enabled Grid applications have been developed under the support of CNGrid, e.g., Drug Discovery Grid (DDGrid) uses workflow extensively to organize the docking process. Aviation Grid (AVIGrid) also uses workflow as the basis in organizing and scheduling the aircraft optimization process.

8.6 Recommendations for EC and China

In order to come closer to the short-term (three years), middle-term (five years) and long-term (ten years) vision mentioned above, it is recommended that the research and engineering communities from both EC and China pay more attention to the following important issues:

First, to make the composition of services an explicit research topic in the revised EU Work Programme is a way to match its emphasis on the Chinese Research Programme, for example, to close the gap by mapping between the scientific level and the business level of workflows, support workflow generation by mining workflow repositories for similar workflows and workflows solving similar tasks, etc.

Secondly, advanced workflow technology is central to bring Grid technology to decision maker's workbench. Moreover, research programme should not only care for Grid architecture and typical Grid applications themselves, but also pay more and more attention to the workflow/business process technology.

Thirdly, like solution partitioning, subprocess and workflow implementation, management, workflow pattern reusability, system scalability and resilience options, and others also need to be discussed in the future.

Fourthly, within the distributed environment, event infrastructure, event routing, event correlation, event template customizing and other methods for event processing are essential to workflow and application integration.

Research should not focus on the development of new workflow tools, but concentrate on far-reaching advances, such as integration of these workflows from different workflow tools in a hierarchy of abstraction levels, designing advanced workflow composition, execution and debugging tools which support this integration at each and between abstraction levels, matching between the real-time events driven by the workflow and the simulation level parameters.

Finally, continually connection and exchange of knowledge and experience, and projects-driven collaboration on Grid workflow/business process engines should be preserved to drive the implementation of short-term, middle-term and long-term common vision.

8.7 References to the elements that lead us to the conclusions and roadmap

Website of CNGrid: http://www.cngrid.org/
http://vega.ict.ac.cn/gos/
http://xservice.ow2.org/xwiki/bin/view/Main/WebHome

Taverna project website: http://taverna.sourceforge.net/
http://www.noesissolutions.com/index.php?col=/products&doc=drivers
http://www.gridworkflow.org/snips/gridworkflow/space/SIMDAT
Bridge Grid: http://www.bridge-grid.eu/
The Kepler Project: https://kepler-project.org/
http://www.oasis-open.org/committees/wsbpel/
http://echogrid.ercim.org/component/option,com_contribution/task,view/

9 Conclusion

The recommendations to the European and Chinese research funders, and research planners for each of the six topics are stated at the end of the each of the preceding six chapters. A very brief summary of these recommendations is:

- Programming paradigms research activities should be put in the main stream of R&D projects instead of being left peripheral.
- Put effort in dynamism (such as dynamic service selection, composition, business models support, etc).
- Emphasize on the development of WS standards for Grids that will also support backwards compatibility.
- Develop open source competitors for tools that support distribute SOA infrastructures (e.g. WebSphere).
- Grid Management should be an explicit focus of research in both EU and Chinese research initiatives and focus should be on SMEs and industry to experiment with innovative business models upon them.
- Grid Management should take considerable input from the Telco standard fraternity. ETSI could take the role for Grids in Europe, the Ministry for Information in China that ITU does for Telcos, as Regulator and Standardization body.
- For the first 3 year period, a clear definition of VO framework on the functionalities and management infrastructures should be proposed.
- For the middle-term aim, advanced functionalities of VO except for semantic support, cross platform VO and high efficient VO should be provided to the users.
- For the 10 year goal, semantic support in VO should be realized, and relevant standards should be raised. Also, autonomic capabilities for full life-cycle management of VO should be provided.
- Clarify the relationships among component-based and service-based models, environments and tools, in particular concerning the possibility to migrate the innovative component features of Grid component model (such as GCM) to WS/SOA platform (such as X-Service).
- Both the EC and China need to put together research groups that will tackle these important issues.
    ① New component models capable of supporting new properties of future Grid software, such as autonomic execution, dynamic roles, adaptive behaviours, QoS etc;
    ② Collaborations should be funded to speedup the development of programming framework fitted for both scientific and business applications;
    ③ Interoperability of different component models, at different layers.
- To make the composition of services an explicit research topic, for example, to close the gap by mapping between the scientific level and the business level of workflows,

support workflow generation by mining workflow repositories for similar workflows and workflows solving similar tasks, etc.

- Research programme should pay more and more attention to the workflow/business process technology.
- Research should not focus on the development of new workflow tools, but concentrate on far reaching advances.

10 References

[7] The MAC system : The computer utility approach, R. M. Fano, in IEEE Spectrum, vol. 2, pp. 5644, January 1965.
[8] Time-sharing on computers, R. M. Fano and F. J. Corbato, in Scientific American, September 1966.
[9] What is the Grid? A Three Point Checklist, Ian Foster, 2002, http://www-fp.mcs.anl.gov/~foster/Articles/WhatIsTheGrid.pdf
[10] The Globus Toolkit
http://www.globus.org/toolkit/
[11] The Open Grid Services Architecture
http://www.globus.org/ogsa/