Project no. FP6-004265

CoreGRID

European Research Network on Foundations, Software Infrastructures and Applications for large scale distributed, GRID and Peer-to-Peer Technologies

Network of Excellence

GRID-based Systems for solving complex problems

**D.IA.09 – Annual Scientific Roadmap**

Due date of deliverable: April 30, 2006
Actual submission date: June 16, 2006

Start date of project: 1 September 2004
Duration: 48 months

Organisation name of lead contractor for this deliverable: INRIA

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

Keyword List: Joint Research Program, Scientific Annual Roadmap, Research Activities
Table of content

1 Introduction .................................................................................................................. 6

2 Research Roadmap ..................................................................................................... 6
   2.1 Knowledge and Data Management (KDM) ............................................................... 6
   2.1.1 Challenges ........................................................................................................ 6
   2.1.2 Research activities .......................................................................................... 7
   2.1.3 Partners contribution to the research Roadmap ................................................ 9
   2.2 Programming Model (PM) ................................................................................... 9
   2.2.1 Research Challenges ....................................................................................... 9
   2.2.2 Research activities .......................................................................................... 10
   2.2.3 Partners contribution to the research Roadmap ................................................ 11
   2.3 System Architecture (SA) .................................................................................... 12
   2.3.1 Research Challenges ....................................................................................... 12
   2.3.2 Research activities .......................................................................................... 12
   2.3.3 Partners contribution to the research Roadmap ................................................ 15
   2.4 Grid Information, Resource and Workflow Monitoring Services (IRWM) .......... 15
   2.4.1 Research Challenges ....................................................................................... 16
   2.4.2 Research activities .......................................................................................... 16
   2.4.3 Partners contribution to the research Roadmap ................................................ 18
   2.5 Resource Management and Scheduling (RMS) .................................................... 18
   2.5.1 Research Challenges ....................................................................................... 18
   2.5.2 Research activities .......................................................................................... 19
   2.5.3 Partners contribution to the research Roadmap ................................................ 21
   2.6 Grid Systems, Tools and Environments (STE) ....................................................... 22
   2.6.1 Research Challenges ....................................................................................... 23
   2.6.2 Research activities .......................................................................................... 23
   2.6.3 Partners contribution to the research Roadmap ................................................ 24

3 High-level scientific positioning .................................................................................. 25
   3.1 Knowledge and Data Management ....................................................................... 25
   3.2 Programming Model ............................................................................................ 25
   3.3 System Architecture ............................................................................................. 26
   3.4 Grid Information, Resource and Workflow Monitoring Services ....................... 27
   3.5 Resource Management and Scheduling ................................................................. 28
   3.6 Grid Systems, Tools and Environments ................................................................ 29

4 Conclusion .................................................................................................................. 30
Executive Summary

This deliverable describes the annual roadmap for the six research areas associated with the CoreGRID Joint Research Programme. For each area we identified the objectives, the planned research activities, the involved researchers and the planned meetings. Overall, this is a refinement and the implementation of the Programme for jointly executed research activities as described in the CoreGRID Annex I “Description of Work”. This second annual roadmap has been established on the basis of the recommendations made by the reviewers. At the time this annual scientific roadmap has been written, the Network has run for more than 18 months. The Network has increased its level of integration presenting itself as a European research laboratory structured in six Institutes. These Institutes are themselves organized into a set of well-focused research groups. Compared to the previous annual scientific roadmap, we slightly changed the name of two research areas. “Grid Information and Monitoring Services” has been renamed “Grid Information, Resource and Workflow Monitoring Services” to extend its scope to Workflows, which is an important topic in Grid research. “Problem Solving Environments, Tools and Grid Systems” has been renamed “Grid Systems, Tools and Environments” since this Institute is mainly concerned with the design methodology for a generic component system that leads to the development of PSEs and not the development of PSEs per se. The six research roadmaps can be summarized as follows.

Research Roadmaps

In the area of Knowledge and Data Management (KDM), the objective is to design and develop specific and common solutions and services for data management and knowledge discovery and management on Grids promoting a wide diffusion and use of knowledge-based Grid services for the Semantic Grid and the Knowledge Grid. The main scientific challenges are the unified access of numerous heterogeneous distributed storage devices, the cost-efficient administration of the hardware and software resources involved, and the offering of quality-of-service guarantee to service users. More precisely, the following basic challenges are addressed: data heterogeneity, algorithm integration and independence, compatibility with Grid infrastructure and Grid awareness, openness, cost-effective scalability, distributed and streamed nature of data sources and security and data privacy. To address these challenges, the research roadmap is structured along three main directions: distributed storage management, information and knowledge management, and data mining and knowledge discovery. The KDM Institute involves 11 CoreGRID partners including 41 researchers and PhD students. The roadmap is implemented thanks to 7 research groups.

Concerning Programming Model (PM), the objective is to raise the level of abstraction in such a way that the programmers are provided with higher level programming models. Such programming models must actually be able to relieve the programmers from most of (possibly all) the burden involved in the direct management of the specific, demanding and low-level issues of Grid infrastructures. The research challenges is the definition of a suitable programming model based on software components, how they can be composed, their associated semantics, the definition of performance/cost models for reasoning about components and security. The research programme of the PM Institute is organized in three related research directions: the definition of a basic component model, the capability of making new components from the composition of existing components and a more advanced programming model taking into consideration theoretical and performance models and properties such as autonomy and security. The PM Institute involves 12 CoreGRID partners including 60 researchers and PhD students. The roadmap is implemented thanks to 8 research groups.

The main objective of the System Architecture (SA) Institute is to achieve a significant improvement of architectural designs of future Grids by focusing on specific issues of three particular key aspects: scalability, adaptability, and dependability of Grid architectures and Grid services. The research challenges are how to make Grid system architecture scalable avoiding centralizing approaches, to let them adapt to the change in internal and the external system state and last but not least how to manage faults that have to be seen as ordinary events. To take up these challenges, the Institute has identified six focused research directions: P2P based services and resource discovery, self-organizing Grid services using P2P technology, dependability mechanisms for computational and data Grids, fault-tolerant and robustness and adaptive management of systems and resources. The SA Institute involves 12 CoreGRID partners including 65 researchers and PhD students. The roadmap is implemented thanks to 12 research groups.

In the area of Grid Information, Resource and Workflow Monitoring Services (IRWM), the focus is on monitoring of Grid resources, support for checkpointing, and management of complex workflows. The associated research challenges are the following: monitoring of large scale systems that grow with the square of the dimension of the system, capturing the state of Grid applications to allow checkpointing in a large scale.
distributed and heterogeneous environment, monitoring the execution of complex workflow and the definition of accounting mechanisms. The IRWM Institute defined four main research directions: information and monitoring, checkpointing, workflow and accounting and user management. The IRWM Institute involves 10 CoreGRID partners including 43 researchers and PhD students. The roadmap is implemented thanks to 7 research groups.

The key objective of the Institute on Resource Management and Scheduling (RMS) is the development of common and generic solutions for Grid scheduling and management in Next Generation Grids. The research challenges that have to be taken up are the following: creation of a common model for Grid scheduling, definition of specific Grid scheduling strategies, support of Grid business models, management of various resources types in a common management and coordination framework, automatic task reallocation in the occurrence of faults and benchmarking and performance evaluation. Seven research directions have been identified: definition of a Grid scheduling architecture, multi-level scheduling strategies, workflow scheduling strategies, evaluation and benchmarking of Grid scheduling systems, mapping and scheduling of HPC applications, coordination of Grid scheduling and data management and performance prediction. The RMS Institute involves 20 CoreGRID partners including 82 researchers and PhD students. The roadmap is implemented thanks to 20 research groups.

The research activities of the Grid Systems, Tools and Environments (STE) Institute are focused around the development of the design methodology for a generic component system that integrates application components, tools/system components, PSE/portal components, and infrastructure components. The adopted approach encompasses both the application of the component model, and its integration into a tools framework. The main research challenge is the design specification of a single, seamless “invisible” Grid software infrastructure. This flexible and lightweight Grid platform will be suitable for a wide range of devices from portable PDAs to parallel supercomputers, providing support for both the client-server and the peer-to-peer paradigms. In its research activities, the STE Institute also gives specific attention to assessing and reducing the various overheads associated with the components-based approach. The Institute has identified four research activities: definition of a generic platform, mediator components, integrated toolkit and advanced tools and environments for problem solving. The STE Institute involves 13 CoreGRID partners including 80 researchers and PhD students. The roadmap is implemented thanks to 9 research groups.

Positioning

The research activities described above are addressing challenges that have also been identified by other internationally well-know research groups. Therefore CoreGRID is involved in a very competitive research field.

For example, in the area of Knowledge and Data Management, the USA and Asia currently strive to address issues such as building scalable storage systems that can hold petabytes of storage in a cost-effective manner and making the storage infrastructure location-independent and client-agnostic in an efficient manner. However, there are no common methods for describing storage services realized at various levels of abstraction by different elements of the storage infrastructure, and there is no common approach to the problem of virtualizing the storage resources without losing their specific, interesting features.

In the context of Programming Model, the CCA initiative, mainly contributed from US researchers, is aimed at “defining a standard component architecture for high performance computing”. The initial specification did not involve directly Grids. However a consistent part of the recent works in the CCA community has been dedicated to target Grid architectures. Our Grid Component Model has been designed for Grids and addresses many issues not yet addressed by CCA.

In System Architecture, most of the current practices in the design of Grid middleware both in the USA and Asia, follow a client/server approach while our research roadmap aims at combining Grids and P2P technologies to allow next generation Grid middleware to be scalable, adaptable, dependable and self-managed.

In the area of Grid Information, Resource and Workflow Monitoring Services, current Grid monitoring architectures are not yet able to provide reliable information and monitoring systems for general heterogeneous large scale Grids. Concerning Checkpointing, which is an important functionality that a Grid middleware system should provide, current implementations are very embryonic. Our effort will go toward the definition of a generic Grid Checkpointing architecture. Workflow composition is a challenging task because the system has to deal with resource unreliability and unpredictability, which are closely related to fault tolerance and scheduling. Unfortunately, these issues have not been addressed to by most of the related work on Grid-based workflows. Most of this work assumes no faults on the Grid and poor or no information provided to the system to deal with...
Grid dynamics. For the management of user accounts, current research efforts do not really address the large-scale dimension of Grids (numbers of virtual organizations and number of users).

In Resource Management and Scheduling, there are currently not many accepted solutions in this area. Instead there are several implementations for specific projects. The lack of solutions is caused by the complexity of resource management and coordination and its reliance on the availability of many other Grid functionalities and services (information, monitoring, resource/job description, execution management etc.). CoreGRID contributes to the design of interoperable scheduling systems and to provide the foundation for efficient resource management solutions.

Finally, in the area of Grid Systems, Tools and Environments, our generic Grid platform will advance significantly the state-of-the-art enabling the design of component-based Grid systems for both applications and tools/systems/environments to have a single, seamless “invisible” Grid software infrastructure. Our adopted approach enables the design of re-configurable Grid systems with dynamic properties guaranteed by our component-based framework. Thus, our flexible lightweight Grid platform is expected to be suitable for a wide range of devices from portable PDAs to parallel supercomputers while other research efforts only address single type of devices.
1 Introduction

This document describes the annual roadmap of the CoreGRID Network of Excellence. It is based on the roadmaps version 2 that have been prepared by the six Institutes. The objective of this document is to give a unified vision of the research roadmaps as well as a single gateway as recommended by the reviewers after the evaluation of the first annual roadmap. It gives also a high-level scientific positioning of the whole Network within the context of the Grid research landscape.

The second annual roadmap reflects the increasing level of integration within the Network. CoreGRID is now established as the European-wide Grid research laboratory organized in six Institutes corresponding to the six research areas identified at the beginning of the project. Each of these six Institutes is further split in research groups that are dedicated to solve specific research challenges. This structuring was not planned when the Network was set up, but considering the size of each Institute as well as the increasing number of collaborations between partners, it was necessary to add this new layer in order to efficiently implement the scientific roadmap and to make the research results more visible outside the Network.

2 Research Roadmap

This section describes the research roadmap of the six Institutes. For each of them, their research challenges are identified and the list of research activities are presented. We also show the participation of partners to the research groups.

2.1 Knowledge and Data Management (KDM)

Data and knowledge are going to play a more important role in current and future Grids. The issues surrounding the representation, discovery, and integration of data and knowledge in a dynamic distributed environment have to be addressed. The key KDM objective is to design and develop specific and common solutions and services for data management and knowledge discovery and management on Grids promoting a wide diffusion and use of knowledge-based Grid services for the Semantic Grid and the Knowledge Grid. To this end, the partners of the KDM Institute are focusing on the problems of providing commodity-based connectivity across heterogeneous distributed storage devices, management automation of administration tasks traditionally handled manually, and storage virtualization for serving well-defined requirements from multiple users. Furthermore, the focus is also on developing knowledge techniques and tools for supporting data intensive applications and the integration of data analysis techniques for information and knowledge extraction on Grids. Another area that will be investigated is the need to develop autonomous security mechanisms in the context of decentralized data management involving self-organizing behaviour, so that no human intervention is needed to ensure security.

2.1.1 Challenges

The main scientific challenges in distributed data management are the unified access of numerous heterogeneous distributed storage devices, the cost-efficient administration of the hardware and software resources involved, and the offering of a quality-of-service guarantee to system users. In the following, we identify some basic challenges that must be faced in the development of data and knowledge management systems and applications in Grids.

- Data heterogeneity and large data sets. Systems must be able to cope with very large and high dimensional data sets that are heterogeneous in structure, geographically distributed and stored in different types of repositories as structured data in DBMS, text in files or semi-structured data in Web sites.
- Algorithm integration and independence. The architectural solutions must allow the integration of different algorithms and tools and must be as independent as possible from the data sources.
- Compatibility with Grid infrastructure and Grid awareness. The designed services must be interfaced with the lower levels of the Grid infrastructure. The interface must be aware of the Grid services and access them for supporting applications.
- Openness. Solutions must be open in order to be integrated with data management tools and knowledge oriented systems.
• Cost-effective Scalability. Designed systems must be scalable both in terms of number of nodes used for performing distributed tasks and in terms of performance achieved by using large Grid configurations.
• Distributed and streamed nature of data sources. Nowadays continuous and unbounded streams of data are naturally collected or summarized in distributed sites, due to either plural data ownership or geographical distribution of the processes that produce data. In many real cases such data cannot be moved and centralized, due technical, policy or privacy reasons. Therefore knowledge discovery algorithms and systems that need to access many of these data sources, have to cope with the issues deriving from the distributed and streamed nature of such data.
• Security and data privacy. Security and privacy issues are vital features in wide area distributed systems. Grid services must offer valid support to systems to cope with user authentication, security and privacy of data. Basic Grid functionality (e.g., Globus security infrastructure - GSI) must be exploited to support secure client-server interactions without impacting on the usability of the Grid infrastructure and services. Moreover, models, algorithms and techniques are needed to implement privacy preserving distributed data management and data analysis on Grids.

2.1.2 Research activities

To achieve the objectives and to contribute to solve the research challenges described above, the activities of the KDM Institute are coordinated around three main directions, which are presented in more details in the following sections.

2.1.2.1 Distributed Storage Management

Demand for on-line storage has been growing dramatically over the last few years. The activities undertaken by project partners in this area are related to research performed currently both in leading industrial and academic organizations for reducing the cost of ownership of storage facilities. All efforts in this direction, essentially aim at replacing existing centralized solutions (e.g. large, centralized storage controllers) with more scalable distributed solutions, at varying degrees of distribution and heterogeneity, depending on the target applications. For example, the Internet is currently used not only by people, but also by agents, bots, and spiders that search, retrieve and process information on behalf of their owners. It is speculated that within a few years, a human will never see most of the information on the network; instead it will be entirely created and consumed by computer applications, which will skyrocket the amount of stored information.

The research roadmap in this area aims at exploring how next generation storage systems may be built by using generic commodity components and by developing the required technology to build future storage infrastructure. These next generation systems will be location-independent and client-agnostic to reduce access costs and to enable new applications. Key factors to reduce management complexity and cost will be identified and mechanisms for policy-based storage management will be provided. Particular features such as data reliability and performance when accessing remote data are required. Such functionality has been investigated for several years in the storage systems community, and remains an open problem to a large extent. Part of the research roadmap will be performed in cooperation with two other CoreGRID Institutes (SA Institute for dependability and IRWM for performance monitoring).

Another important issue is how to ensure and to manage security of stored information. Currently, protection of stored information is enforced by access control at the file system level. However, as storage is becoming networked, where disks are detached from the application and file servers are connected to a datacenter-type network, this type of protection is not adequate. The Institute will study new methods on how to be able to provide fine-grain security at the block level for large-scale storage systems at affordable cost and without compromising system performance. It also plans to investigate possible techniques for delegating, mapping and verifying the credentials of the client systems in particular layers of the storage services.

2.1.2.2 Information and Knowledge Management

Information and services have to be given a well-defined meaning to better enabling Grid resources and people to work in cooperation. Ontologies are among the key building blocks to implement the vision of a semantic Grid. They define and determine the concepts, vocabularies of entities, resources, capabilities and the relationships between them, with which any kind of content can become meaningful by the addition of ontological annotations. The main problem for building an ontology for Grids is that there is currently a multitude of proposed Grid architectures and implementations, which are comprised of thousands of entities,
services, components, and applications. It is thus very difficult, if at all feasible, to develop a complete ontology that will include all aspects of Grids. Within CoreGRID, the KDM Institute develops a Core Grid Ontology (CGO) that defines fundamental specific concepts, vocabularies and relationships. One of the key goals is to make this CGO generic enough and easily extensible to be used by different architectures or middleware, so that the CGO can provide a common basis for representing Grid knowledge about resources, middleware, services, applications, and users. The role of the CGO is to provide a higher-level framework in which all concepts of Grids can be given a consistent and semantically coherent representation. Thus it is designed as an upper-level ontology, which captures and models the basic concepts and knowledge of Grids.

Managing and discovering information about Grid resources/services are the key aspects of Grid middleware. However, resource information services tend to be centralized and thus have a limited scalability. For next generation Grids, it is mandatory to make scalable such services. One way is to follow a P2P approach to avoid any contention. Several CoreGRID researchers have started research activities investigating the use and the adaptation of existing P2P system to design scalable information service. In particular, they are developing the Tree Vector Indexes, a P2P system able to efficiently implement range queries for dynamic contents identified by multiple numeric attributes. They plan to extend their work, by considering the possibility of using more unstructured P2P networks, other than simple trees. In addition, they will investigate the exploitation of indexes more sophisticated than simple bitmap ones, in order to limit the network flooding, and also the adoption of queries (like top-k) that can be solved by limiting the amount of retrieved data. Another research activity will focus on the development of a coherent framework in which Grid resources and their associated features/metadata can be organized according to various levels of dynamicity (related to their rates of change). In this context, the Institute is studying and evaluating how emerging P2P models and protocols can be used and extended to support the realization of novel information discovery services, with different mechanisms for different level of dynamicity. The levels of information dynamicity will be used to decide how and where the same information will be indexed in the distributed system, and how such indexes will be maintained up-to-date.

P2P systems and service oriented Grids have been quite successful in providing an easy way for users to share their data. They enable decentralized collaboration by integrating computers into networks in which each can utilize and offer services. However, both of these systems lack an integrated and efficient approach to semantic data sharing. Filling this gap is the main objective of the ongoing research efforts within CoreGRID. Although P2P tools and algorithms help leveraging the heterogeneous, dynamic and large-scale properties of the environment, a serious limitation of most of these approaches is that they only support keyword queries. These techniques remain at low, system level and do not take into consideration data management issues. The KDM Institute is working on a P2P data integration approach to efficiently share large scale and heterogeneous data distributed over a large set of networked nodes. The Institute will propose, in this context, state-of-the-art data integration algorithms and techniques available on top of peer-to-peer overlay. The combination of decentralized data integration with gossip-based (unstructured) overlay topology management and (structured) distributed hash tables (DHT) provides the required level of flexibility, adaptability and scalability, and allows rich queries to be performed on a number of autonomous data sources.

Query mechanisms and intelligent agents for query formation are investigated in the context of the OGSA-DAI and OGSA-DQP systems and services. The Institute proposes to design a framework for defining and constructing adaptive query processing (AQP) systems both for Grids and for more traditional environments. Moreover, a generic mechanism for extracting monitoring information from the query execution to support multiple AQP techniques is also considered. The framework being developed tackles two important problems: software reuse and adaptability to changing resources.

2.1.2.3 Data Mining and Knowledge Discovery

Knowledge discovery processes and data mining applications generally need to handle large data sets and, at the same time, are compute intensive tasks that in many cases involve distribution of data and computations. This is a scenario where the use of parallel and distributed computers can be effective for solving complex data mining tasks. Knowledge discovery services on Grids allow professionals and scientists to create and manage complex knowledge discovery applications composed as workflows that integrate data sets, mining tools, and computing and storage resources provided as distributed services on a Grid. Services orchestration allows users to compose, store, share, and execute these knowledge discovery workflows as well as publish them as new components and services on the Grid. CoreGRID researchers will work in this area to define high-level models for data mining workflow composition.

The Institute will also investigate algorithms, systems and services for distributed data mining in scalable systems such as Grids and P2P systems. Software technologies for the implementation and deployment of
knowledge Grids will provide important elements to build up knowledge-based applications on a local Grid or on a large-scale Grid. These models, techniques, and tools can provide the basic components for developing Grid-based data analysis systems. The design and implementation of Grid-based distributed data mining services that are leveraging the OGSA and WSRF standards will be investigated and mining algorithms will be integrated in OGSA services to provide knowledge discovery services on Grids. Related issues such as metadata management for supporting data mining and resource/services discovery will be also considered.

2.1.3 Partners contribution to the research Roadmap

The KDM Institute involves 11 CoreGRID partners including 41 researchers and PhD students. The roadmap will be implemented thanks to 7 research groups as listed in the following table.

<table>
<thead>
<tr>
<th>Knowledge and Data Management</th>
<th>CETIC</th>
<th>ISTI-CNR</th>
<th>ICST - FORTH</th>
<th>INFN</th>
<th>PSNC</th>
<th>RAL - CLRC</th>
<th>SZTAKI</th>
<th>UCY</th>
<th>UNICAL</th>
<th>UNCL</th>
<th>UoM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid Data Storage Access and Data Management Architecture</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Storage Security</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grid Data Integration Models and Architectures – Scheme Integration in OGSA-DQP</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resource Description and Discovery in Large Collaborative Networks (inc. core GRID ontology)</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Methods for Deriving Grid trust and Security Policies for Managing Virtual Organization</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distributed Data Mining in Grids and P2P Systems</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distributed Query Processing in the Grid</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.2 Programming Model (PM)

Substantial research efforts have to be invested to raise the level of abstraction of future generation Grid systems. In particular, it is necessary to raise the level of abstraction in such a way that programmers are provided with higher level programming models. Such programming models must actually be able to relieve the programmers from most of (possibly all) the burden involved in the direct management of the specific, demanding and low-level issues of Grid infrastructures. Current programming practices for the development of Grid applications share many similarities with what the programmers faced forty years ago with the programming of computers using assembly languages. With the availability of Grid infrastructures, it becomes pressing to develop a high-level programming model to allow the independence of the applications vis-à-vis the Grid infrastructures and to raise the level of abstraction to increase programmers’ productivity. The key objective of the PM Institute is to define a component-based programming model that can be usefully exploited to design, implement and run high performance, efficient Grid applications. The research programme of the PM Institute is organized in three related research directions: the definition of a basic component model, the capability of making new components from the composition of existing components and a more advanced programming model taking into consideration theoretical and performance models and properties such as autonomy and security.

2.2.1 Research Challenges

The Grid poses new challenges in terms of programmability, interoperability, code reuse and efficiency. These challenges arise mainly from the features that are peculiar to Grid, namely heterogeneity and dynamicity. Grid programmability, in particular, represents a substantial challenge. Grid programs cannot be readily expressed using conventional programming models and tools, unless the programmer is prepared to pay a high price in terms of programming, program debugging and program tuning effort. New programming models are required that exploit a layered approach to Grid programming and which offer user-friendly ways of developing efficient, high performance applications. This is particularly true in cases where the applications are complex and multidisciplinary. Within CoreGRID, the challenge is to design a component based programming model that overcomes the major problems arising when programming Grids:
A suitable programming model (that is user-friendly and efficient) to program individual components is needed.

Component definition, usage and composition must be organized according to standards that allow interoperability to be achieved.

Component composition must be defined precisely, in such a way that complex multidisciplinary applications can be constructed by the composition of building block components, possibly constructed by suitably wrapping existing code. Component composition must guarantee scalability, i.e. the implementation of component composition and the composition model itself must be “bottleneck free”.

Semantics must be defined, precisely modelling both the single component semantics and the semantics of composition, in such a way that provably correct transformation/improvement techniques can be developed.

Performance/cost models must be defined, to allow the development of tools for reasoning about components and component composition programs.

Security issues must be investigated and proper implementation techniques must be adopted to allow safe and secure component based Grid computing.

All of these problems must be dealt with, taking into account that improvements in hardware and software technology require new Grid systems to be transparent, easy to use and to program, person-centric rather than middleware, software or system-centric, easy to configure and manage, scaleable, and suitable for use in pervasive and ubiquitous contexts.

2.2.2 Research activities

To achieve the objectives and to contribute to solve the research challenges described above, the activities of the PM Institute are coordinated around three main directions, which are presented in more detail in the following sections.

2.2.2.1 Basic Programming Models

The first research direction aims at defining suitable models and tools to support the programming of single components that will constitute component based Grid applications. Several on-going research activities inside the Institute have been already set up, related to programming models for the implementation of single components of a GCM (Grid Component Model, the component model currently investigated in the Programming Model Institute). Single components can be either sequential or parallel. They can include legacy codes and can be very demanding with regard to optimal use of the underlying hardware. Finally a component can interact with non-GCM based applications. Based on on-going research activities in the Institute, this research direction aims to produce the definition of a precise range of programming models suitable for use in single components. Beside the numerous programming models already available for implementing parallel and distributed single components (such as MPI, OpenMP, HPF, RMI,...), several CoreGRID partners are working on original approaches trying to combine parallel and distributed programming paradigms as, for example, PaCO++, Pro-Active, Ibis, Superscalar, ASSIST or POP-C++. If this can be seen as a fragmented approach pursuing the goal of defining programming models for the Grids, it can be also a think tank to define a more ambitious programming model. One of the main activities will be thus to survey existing approaches and those under development, to compare them and to identify suitable primitives and mechanisms for programming single parallel components. Chosen models should support programming parallel components based on SPMD and MPMD programming styles.

In addition, Institute members investigate within this research direction suitable techniques to optimize the communications (point to point as well as collective) occurring between and within components, i.e. inter- and intra-component and with non-GCM based applications. Communications are used to transfer data between tasks and to synchronize them. Two types of communications have been identified: point-to-point communications and global communications. The way in which a programmer is exposed to communications (communication model) depends on the programming model he is using. It can be explicit messages transfers, as in MPI, or implicit communications as in Pro-Active or POP-C++. However, in both cases, communication mechanisms may greatly influence the performance of both the single component and the interaction between components. Thus better co-design of the communication model and the communication mechanism may significantly improve the efficiency of the computation. Researchers in the Institute will make explicit (and compare) the communication mechanisms used by the various CoreGRID teams implementing parallel and distributed programming models in order to identify the most suitable one for our GCM.
2.2.2.2 Component and Hierarchical Composition

This research direction aims at defining a component model suitable for Grid programming, and allowing components to be composed, either sequentially or in parallel, to construct new components. During the first year of CoreGRID, the Institute members made a proposal for the GCM. It highlights the main features to be included in a component model suited to Grid programming. This proposal needs to be further refined and assessed in order to produce a generalization of the component model. The current definition needs agreement on the following issues: communication mode among components (synchronous/asynchronous, one way/two ways, 1-to-1/1-to-N/N-to-M, etc.), support for parallel components (how to define them, how and where to launch them, how to synchronize them, how to enable them to communicate, etc.), introduction of controllers to control Grid specific features or to provide adaptive behaviour at the level of the components themselves (such as adaptive and autonomic behaviour). Another research direction has been identified in the concrete definition of the component model. The Institute's plan is to specify a component using XML schema or DTD (Document Type Definition) (inspired by existing ADLs), to define runtime API defined in several languages (inspired by and extending the current Fractal API spec) as well as packaging of components described as an XML schema. Finally the last identified research direction individuated concerns the definition of several conformance levels. They should allow any CoreGRID partner to relate existing component models to the GCM, by determining the conformance levels that are reached. Conformance levels will be defined according to several distinct features. The basic conformance levels will encompass basic characteristics upon which more sophisticated ones will be built (hierarchical features, distribution and definition of distributed components, introspection, dynamic reconfiguration, type definition for components, etc.).

2.2.2.3 Advanced Programming Model

The third research direction aims at defining higher-level models that permit programmers to use components and component compositions in more efficient and user-friendly ways. Several research directions have been identified and will have to be coordinated. Institute partners will contribute to the definition of “theoretical” computation models that can be used to support the Grid component model as well as performance models developed to cover major Grid computing parallel patterns. Institute partners will also study some program transformation techniques to improve the performance features of the user-supplied code via automatic program rewriting techniques. Since Grid infrastructures are intrinsically dynamic (resources may disappear at any time), the PM Institute will propose extensions to let components be autonomous taking care of all the details related to the management of component interaction with the underlying Grid middleware. Another important issue is related to security. In this context Institute partners will contribute to enhance security of the deployment of component codes, the communication between components and access to components themselves.

2.2.3 Partners contribution to the research Roadmap

The PM Institute involves 12 CoreGRID partners including 60 researchers and PhD students. The roadmap will be implemented thanks to 7 research groups as listed in the following table.

<table>
<thead>
<tr>
<th>Programming Model</th>
<th>IS</th>
<th>IC</th>
<th>INRIA</th>
<th>UQ</th>
<th>WW</th>
<th>UCHILE</th>
<th>UC</th>
<th>UNIPASSAU</th>
<th>EP</th>
<th>EO</th>
<th>UP</th>
<th>VA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Component</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Communication</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Programming</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>model for the</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>single component</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Component</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>definition</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Advanced</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>programming</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>models</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Performance</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>models, adaptivity</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Component</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>runtime support</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>❑</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

1 See deliverable D.PM.03 “Roadmap version 2 on Programming Model” for a complete list of characteristics.
2.3 System Architecture (SA)

Every Grid system, as every non-trivial system in general, features an architecture – the definition of its individual building blocks, interconnections between them, and the general principles that govern their composition and interoperability. Grid systems will be composed from many components, which number and diversity will only increase over time. Such complexity of the Grid has to be fully hidden from users and developers through complete virtualization of Grid resources. Different Grid systems can be composed from the same reusable, usually pre-existing resources and components. Virtualization of resources demands certain uniformity and standardization, which further increases the role of architecture in the Grid. The SA Institute focuses on the architectural principles of Grid applications and infrastructure that meet the mandatory properties on Next Generation Grids. The main objective of this Institute is to perform a significant improvement of architectural designs of future Grids by focusing on specific issues of three particular key aspects: scalability, adaptability, and dependability of Grid architectures and Grid services.

2.3.1 Research Challenges

The scale, dynamism and openness of the Grid, together with demands on its reliability, security and manageability, poses new, unique challenges on software architecture. Next Generation Grids will have to be open, large-scale, pervasive and heterogeneous, and will have to deal with diverse types of resources. Yet, in order to exploit their full potential, Grids have to be simple, transparent, reliable, persistent, secure, and easily configurable and manageable. This is clearly a novel combination of demands on software architecture. Guiding principles for building such systems are not known and represent research challenges. The SA Institute identifies three main challenges in the design of Grid system architectures:

- Classical approaches to Grid resource location are either centralized or hierarchical and will prove inefficient as the scale of Grid systems rapidly increases. On the other hand, the Peer-to-Peer (P2P) paradigm emerged as a successful model that achieves scalability in distributed system. While the Grid is currently distributed and semi-decentralized, individual services are still highly centralized, static, and not self-organizing. If this trend continues, substantial amount of administration and management will be required to setup and maintain a Grid infrastructure, which is an obstacle if the Grid is going to be ubiquitously deployed. The challenge is thus how to introduce P2P techniques in the design of Grid system architectures.

- Adaptability is understood as the ability of automatic adaptation of Grid systems to the changes in internal and the external system state. Also this keyword is frequently connected to the notion of self-management which is understood as a set of system abilities to perform configuration, performance tuning, recovery and other management tasks automatically. Adaptability requires the monitoring and the prediction of system behaviour. Currently, no major Grid middleware utilizes tools for modeling and prediction of system characteristics. This situation is partially due to the lack of implementations of suitable algorithms. A further impediment comes from the fact that in current Grid architectures, tools for demand modeling and prediction have not been foreseen as a part of the middleware, and so other components such as schedulers etc. cannot take advantage of them.

- One of the important issues to be solved in Grid Computing infrastructures is the support for fault-tolerance. Due to the complexity and heterogeneity of Grid elements, there is a need to devise new fault-tolerance mechanisms that should be able to adapt to the scalable and dynamic environments of the Grid. The field of dependability has gained notable advances in the past decades in the areas of distributed computing, parallel processing and clusters of computers. However, the fault-tolerant schemes that have been devised for those environments are mainly targeted to small-scale systems. With the advent of Grid computing there is a clear need to adapt the fault-tolerance schemes to scalable, dynamic and wide-area environments that may comprise heterogeneous modules and different Grid middleware.

2.3.2 Research activities

To achieve the objectives and to contribute to solve the research challenges described above, the activities of the SA Institute are coordinated around five main directions, which are presented in more details in the following sections.
2.3.2.1 P2P based services and resource discovery

The general requirements for resource discovery are scalability, reliability, and support for dynamicity. Supporting very dynamic environments is fundamental, since the availability and status of resources within each node change dynamically over time. Another fundamental requirement in Grid systems is the ability to perform multi-attribute and range queries. P2P systems may help designing and implementing scalable, reliable and dynamic resource discovery services. Such P2P systems adopt different models and solutions, including structured or unstructured overlay networks, fully decentralized or super-peer architectures, and diverse strategies for improving routing performance and search precision. Moreover, they provide very different search capabilities, ranging from single-attribute search to multi-attribute and range queries. These different models and solutions have to be adapted in the context of resource discovery in Grid computing systems. The SA Institute will first perform a comparative study of various approaches that currently exist in P2P technology (structured vs. unstructured) and how these can provide benefit to Grid research. If P2P systems have been shown as scalable, it is at the expense of an increasing number of messages exchanged between peers. Lookup in unstructured P2P systems is mainly performed by having each peer forward each incoming query message to all of its neighbours, a process called flooding. Although this algorithm has excellent response time and is very simple to implement, it creates a large volume of unnecessary traffic in today’s Internet because each node may receive the same query several times through different paths. The Institute will design a method for the reduction of the duplicate messages that are produced during the flooding process in unstructured P2P systems in order for flooding to become scalable.

2.3.2.2 Self-organizing Grid services using P2P technology

Grid systems aggregate heterogeneous resources for specific tasks. However, Grid’s existing centralized architectural solutions limit the scalability of Grid systems, and have to be abandoned for scalable, self-organizing designs that would allow these systems to scale. Grid systems can benefit from the P2P paradigm. P2P systems have emerged as self-organizing, adaptive, autonomous, and scale-free distributed resource-sharing environments. Structured P2P overlay networks have sufficiently matured in the last few years that they can be considered as a basic part of an infrastructure for scalable distributed applications. What is lacking is that they do not provide any support for managing applications, e.g., deployment, versioning, and other kinds of configuration management. The SA Institute will conduct a research activity on mechanisms for self-organization and self-management of reconfigurable self-managing services (components) built on top of P2P networks that show a potential for building large-scale systems with autonomic management. The outcome of this activity is expected to be a survey of existing approaches to building Grid services with self-* properties on top of P2P networks. Furthermore, research will be conducted on a programming component-based model for composition of distributed services with self-* properties. The contribution of this activity is expected to be a mechanism for self-organization of reconfigurable self-managing services (components) built on top of P2P networks. Finally, the last activity in this research direction is to deal with the architectural and programming issues of desktop Grids and to develop an architectural framework that interconnects desktop Grids both in a hierarchical and symmetric way. This will enable large number of users to use the resources available on the desktop Grid system as well as, to build a large scalable desktop Grid from smaller organizational level desktop Grids as building blocks. This new architectural concept will significantly extend the current usability of desktop Grid systems.

2.3.2.3 Dependability mechanisms for computational and data Grids

The main focus of this research direction is to study dependability for Grid systems, in particular Desktop Grids. One of the major problems of desktop Grids is the volatility of computing nodes. A common solution to cope with the limitation imposed by resource volatility is checkpointing. An important issue regarding checkpoint lies in the place where it will be stored. An actual limitation of Desktop Grid middleware is the fact that checkpoints are saved in the local disk of each computing node and they can only be used to resume the application in that same machine. This limitation has to be solved and the researchers of the Institute will study some techniques to improve the scalability and survivability of checkpoint data in large-scale systems, how to deal with the heterogeneity of systems and middleware, and how to adapt the checkpoint-and-recovery schemes to the paradigms of Grid applications. An avenue of research that will be exploited will be the distribution of checkpoint images among the network for the sake of availability and scalability. In this case it seems promising to exploit the use of DHT-based and P2P techniques and to compare them with traditional hierarchical schemes for distributed storage.
Desktop Grid must deal with sabotage and malicious failures that may undermine completely the results of a long-running computation. Thereby it is of paramount important to devise techniques and strategies to cope with malicious participants and provide “sabotage tolerance” to Desktop Grid Middleware. Together with the techniques for sabotage-tolerance it is mandatory to devise some protocols for trust management that should be adapted to these environments. If the computations that are performed in open environments are not trustable then Grid Computing will never be performed in those environments. The Institute members will identify problems and will devise novel strategies for sabotage tolerance and trust-management in volunteer-based desktop Grid computing.

The main trends today is to design Grid systems using a SOA approach based on a set of distributed Grid services. The underlying technology, behind a SOA approach, is often based on Web Services and the SOAP protocols. SOAP implementations are prone to the problem of software aging, a phenomena that is observed in long-running applications where the execution of the software degrades over time leading to expensive hangs and/or crash failures. We do believe that SOA-based tools are highly prone to this problem due to the immaturity of the software and the inherent complexity. Software aging happens due to the exhaustion of systems resources, like memory-leaks, unreleased locks, non-terminated threads, shared-memory pool latching, storage fragmentation, data corruption and accumulation of numerical errors. The natural procedure to combat software aging is to apply the well-known technique of software rejuvenation. The research goal of this Institute is to include some techniques inside a SOAP implementation that will be able to predict software aging problems and will take the correct actions (using selective software rejuvenation) to prevent failures to happen. This corresponds to the vision of self-healing systems.

2.3.2.4 Fault-tolerance and robustness

Long-running applications that will execute in Grid infrastructures are easily affected by the occurrence of partial failures in some components of the system due to the increased complexity and the distribution of computing resources. Fault-tolerant techniques and mechanisms should be investigated taking into account some of the intrinsic properties of Grid systems (scalability, dynamicity and heterogeneity). The SA Institute will evaluate the effectiveness of fault-tolerance mechanisms and the reliability level of Grid middleware. It will review several software fault-injection tools and workload generators for Grid Services that can be used for dependability benchmarking in Grid Computing.

Job-batching systems like Condor already provide support for system-level checkpoints for job-migration and recoverability. However, they are not targeted to parallel programs but rather limited to embarrassingly parallel applications. The standard for executing message-passing parallel applications is MPI. The way to deal with faults in MPI programs is still an open research issue of research. Basically there are two main options: (i) the MPI implementation provides some API for fault-tolerance that should be used by the application programmer; (ii) or the MPI implementation provides some logging and checkpointing protocols for automatic rollback-recovery. While this last approach has the potential advantage of transparency it still has some issues to be addressed like the higher performance overhead and the lack of portability.

2.3.2.5 Adaptive management of systems and resources

Modeling and prediction of system characteristics such as resource demand of applications, workloads of servers or machine performance degradation has a multitude of applications within Grids. One of them is increased efficiency of resource sharing by allowing long-term capacity planning and by providing support for scheduling. Another exemplary application area is dependability, where prediction of performance degradation enables adaptive software rejuvenation, anomaly detection, or preventive migration of applications. Many possible approaches exist for modeling and prediction. Some of these methods need further development and partially require “adjustment” for the particular application scenarios like predictive software rejuvenation. Furthermore, a comparative analysis in different application scenarios is required. In addition, supportive techniques are required, for example automatic classification of traces into predictable or “random” for early assessment of modeling benefits.

Lack of automatic configuration and management of systems such as Grid infrastructures has negative impact on both the cost of the operations and the dependability. A partial remedy to this problem can be achieved by the approach of automated and adaptive generation of workflows composed of configuration, management and recovery activities. The research issues involved in this problem will be studied (among others) on the example of automated construction and configuration of complex resources, such as multi-tier applications or virtual data centers. Within this research domain, the Institute will also attempt to develop a user-friendly formalization...
"language" for describing system state (global "services" state and resources), possible configuration and management actions enabling system adaptation and fault resilience and target system state, including both performance and dependability requirements.

One of the key architectural characteristics of open large-scale Grid infrastructures is the heterogeneity of resources, which is due to the distributed ownership of Grid resources and the incremental approach that most institutions take when purchasing or upgrading their clusters. Nevertheless, the architectural paradigms used so far by the Grid community to develop mechanisms for resource selection, job scheduling etc., ignore this fact and follow the more traditional approach of massive-scale parallel system research, where algorithms and tools assume a model comprising many homogeneous processors that have identical characteristics in terms of performance, reliability, maintenance, etc. The Institute plans to investigate the effects that different aspects of heterogeneity have on the quality of service (functionality and performance) provided by Grids to end-users. The Institute will explore the development of high-level metrics of heterogeneity and apply its findings for the development of new algorithms for resource selection and job scheduling in heterogeneous Grids. The goal of the work will be to investigate quantitatively the potential impact that heterogeneity of Grids has on the performance and reliability provided to Grid end-users; furthermore, to come up with algorithms for the matching of Grid jobs to heterogeneous resources.

2.3.3 Partners contribution to the research Roadmap

The SA Institute involves 12 CoreGRID partners including 65 researchers and PhD students. It has to be noticed that 2 other CoreGRID partners (INFN and PSNC) will contribute marginally to the SA research activities and thus are not listed here. The roadmap will be implemented thanks to 12 research groups as listed in the following table.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>P2P techniques for resource discovery in Grids</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scalable resource location in P2P systems</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Building scalable self-organizing services in P2P technologies</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Self-management for applications built on structured overlay</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scalability for desktop Grids</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dependability mechanisms for desktop Grid computing</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sabotage tolerance in desktop Grids</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Self-healing SOA and Grid architectures</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fault-injection and robustness assessment for Grid services</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fault-tolerant MPI</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance-aware Grid resource selection</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Modeling and prediction of workloads and system behaviour</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.4 Grid Information, Resource and Workflow Monitoring Services (IRWM)

Grid infrastructures are subject to frequent changes and disruptions in the service of a huge number of services they are built of. To manage such a dynamic system and its resources, online monitoring of the resources to determine their availability is required. Monitoring is an essential, even critical, part of any Grid infrastructure, helping to hide its complexity. Another important topic for supporting dynamic systems is checkpointing,
overcome hardware and/or software failures. Complex job workflows represent another challenge, as the monitoring information must be almost synchronously gathered from many different sources and appropriately processed to provide a coherent view (state information) of the whole workflow and its components. The problem of managing user accounts becomes a non-trivial one in a distributed environment (especially in a Grid environment) that includes many independent sites and virtual organizations with hundreds or even thousands of nodes with allocated user accounts. Therefore the complexity rises significantly in dynamic systems.

2.4.1 Research Challenges

The main research challenges in the design of Grid Information, Resource and Workflow Monitoring Services are the following:

- Monitoring the network infrastructure of a Grid has a vital role in the management and the utilization of the Grid itself. Network monitoring, as an instance of resource monitoring should produce observations that are used by brokers and other Grid agents in order to optimize Grid applications. The challenge of network monitoring is that it must face a task that, in principle, grows with the square of the dimension of the system. In addition, Network Monitoring is characterized by peculiar security problems since they require cooperation between two “peer” testing applications, thus requiring the existence of an appropriate authentication mechanism.

- The current state-of-the-art are nowadays distributed heterogeneous computing environments, which are very complex. But the requirements are still the same, e.g. high level of reliability. The checkpointing is used to minimize site-effects of hardware or software failures, and finally to support dynamic job migration, demanded by Grid scheduling systems. A kernel level checkpointing is supported only by few operating systems. An application checkpointing is done during the application process development. The research challenge is to deliver a functionality which supports checkpointing in distributed and heterogeneous systems, what increases the flexibility of resource management systems and makes better reliability on a kernel (hardware) and application level.

- Complex job workflows represent another challenge, as the monitoring information must be almost synchronously gathered from many different sources and appropriately processed to provide a coherent view (state information) of the whole workflow and its components. The job workflow itself must be extracted from programming models, the monitoring and information services must be tightly coupled with job checkpointing and migration support to provide an environment where even complex job workflows could be easily deployed, executed, and monitored. Models and methods to provide a virtualized end-user account system are a specific part of combined job flow support and information services.

- The last important challenge is accounting. Existing solutions allow the accounting of resources used for only one system or at most for local and homogeneous clusters. The problem arises in Grids when the environment is more complex by adding virtual organizations and dynamically assigned accounts. Demand for a solution will significantly increase in the near future, especially in context of production Grids and Grid economy. Obviously, the accounting mechanism should provide with some information independently of the service scale and range (a single system, local cluster or Grid environment). The next important issue is scalability and how to reduce overheads generated by accounting mechanism. To ensure a real practical accounting mechanism for the Grid, it should have a decentralized, scalable and flexible structure. It should interfere with local domain policies as little as possible. To meet these requirements, distributed resource allocation accounting models, which would properly work at various sites with different administrations and resource management policies, would be necessary.

2.4.2 Research activities

To achieve the objectives and to contribute to solve the research challenges described above, the activities of the IRWM Institute are coordinated around four main directions, which are presented in more details in the following sections.

2.4.2.1 Information and Monitoring

This research direction is intended to provide a scalable architecture for network monitoring, trying to cope with its quadratic complexity: although we understand that this is intrinsic in its nature, the Institute tries to keep its growth under control, and provide ways to avoid resource saturation. The architecture should provide a clear interface to the outside, taking into consideration that the environment is dynamic, due to progress in the design
of Grid systems. Flexibility and scalability should be a primary concern in the design of the interface of the Network Monitoring Architecture with the rest of the Grid. Scalability issues are seldom taken into account when considering Network Monitoring. There is evidence that they can compromise the availability of Network Resources observations. The IRWM Institute's primary concern is scalability: to this end, the Institute proposes a Network Monitoring Architecture which combines several known concepts: passive monitoring techniques, a domain-oriented overlay network, and attitude for demand-driven monitoring sessions. In order to keep into account the demand of extreme scalability, we introduce solutions to two problems that are inherent to the proposed approach, and well documented in the literature: security and group membership maintenance.

An example of scalability problem is bound to the existence of a shared knowledge, here represented by the overlay topology. The management of such shared knowledge is a potential scalability limit, if based on any kind of centralized control. Therefore the researchers of this Institute have defined a distributed algorithm that keeps an instance of the whole database reasonably updated, describing the composition of the overlay domain partitions, on each Network Monitoring Element. The algorithm, as well as the size of the database, has a complexity that linearly increases with the size of the system, and is based on a random gossip scheme.

### 2.4.2.2 Checkpointing

This activity deals with the development of a Grid Checkpointing Architecture (GCA), which defines a uniform interface of checkpointing services on both levels: kernel and application. The GCA integrates existing functionality with any new developed in the future. The user can expect that the job that is running for a really long period of time will not have to repeat the whole computation once again because of some sort of failure. From the scheduler point of view, checkpointing and migration allows to use dynamic scheduling algorithms because application is no longer required to run from start to end on the same node. Implementation of checkpointing and job migration is justified from the economical point of view as it allows better utilisation of the Grid resources. The current main direction is to integrate GCA with other Grid-related services (and especially with the metascheduler or broker). This activity is carried out through a close cooperation with the members of the RMS Institute.

### 2.4.2.3 Workflow

Another research direction is the study and development of services able to coordinate the reliable execution of vastly complex compound Grid jobs and realize middleware support for complex job workflow execution. This will also include an adequate description and modeling of workflows, mapping abstract onto concrete workflows and providing services for the monitoring of workflows, thus adding support for dynamic workflows on non-reliable Grid resources. Current state-of-the-art Grid workflow management solutions still show a lack of interoperability with other workflow systems, due to incompatible and informal workflow description languages. Therefore, an important objective is to study and to overcome compatibility and conversion issues between commonly available workflow description languages, such as BPEL, Directed Acyclic Graphs, and Petri Nets. Future Grids must support the management of complex jobs and service-level agreements. Those jobs have workflows with co-allocation constraints and dependencies that must be considered for a wide range of diverse resources. Present systems have architectural and design limitations that make them usable in a productive manner only for simple workflows. This is not sufficient for highly complex Grid applications to be expected in important application domains such as industrial design, engineering, drug design and bioinformatics. The current limitations must be overcome by a common set of job management and execution services based on a powerful model.

### 2.4.2.4 Accounting and User Management

Security within Grid infrastructures requires authentication of the user and authorization based on combined security policy from the resource provider and the virtual organization of the user. The second important thing is the possibility of logging user activities for accounting and security reasons and then gathering these data both by the resource provider and virtual organization of the user. From the user point of view, an important feature is single sign-on. The problem of user management is a non-trivial one in an environment that includes a bulk number of computing resources, data, and hundreds or even thousands of users participating in lots of virtual organizations. The complexity rises from the point of view of time required for administration tasks and automation of these tasks. There are many solutions that attempt to fulfil these basic requirements and solve the mentioned problem, but none of them, according to our best knowledge, solve the problem in a complex and satisfactory way. The IRWM Institute will thus focus on supporting full virtualization of user accounts on the heterogeneous Grid, investigating approaches for real-time on-demand user account creation and management,

CoreGRID – Network of Excellence
supporting hierarchical VOs, with a possibility of user and job separation, correct data protection (including failure recovery) and accountability.

2.4.3 Partners contribution to the research Roadmap

The IRWM Institute involves 10 CoreGRID partners including 43 researchers and PhD students. It has to be noticed that one other CoreGRID partner (UNIDO) contributes to the IRWM research activities (collaboration between the IRWM and the RMS Institutes) and is not listed here. The roadmap will be implemented thanks to 8 research groups as listed in the following table.

<table>
<thead>
<tr>
<th>Grid Information, Resource and Workflow Monitoring Services</th>
<th>FHG</th>
<th>FORTH IC</th>
<th>INF</th>
<th>MU</th>
<th>PSNC</th>
<th>SZTAKI</th>
<th>UNICAL</th>
<th>UO</th>
<th>WW</th>
<th>UC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Monitoring System</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integration of TCKPT and PSNC checkpoint</td>
<td></td>
<td></td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Storage functionality for distributed checkpointing</td>
<td></td>
<td></td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Workflow description languages using high-level Petri nets</td>
<td></td>
<td></td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compatibility and conversion of different Grid workflow description languages</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fault tolerance in Grid workflow execution</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Extending the SEAGRIN semantic overlay Grid infrastructure with the collaborative workflow management support of the P-GRADE portal</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>User management and accounting framework</td>
<td></td>
<td></td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.5 Resource Management and Scheduling (RMS)

The management and coordination of Grid resources is a central topic for Next Generation Grids. Many of the anticipated features of Grids include automatic management, reliability, support for business scenarios and especially ease-of-use and transparency for the end-users. Most of these features require a powerful and feature-rich resource management that hides the complexity of Grids like leveraging resource selection, orchestration under consideration of several objectives.

Currently, there is neither a coherent and generally accepted infrastructure to manage and schedule resources nor are there efficient coordination algorithms that suit the complex requirements of a large scale Grid environment with different resource types. While there are several approaches to Grid resource management and scheduling, most of them consider only parts of the problems. A general vision and a coherent approach to integrate different models into a viable solution are still missing. The key objective of the Institute on Resource Management and Scheduling (RMS) is the development of common and generic solutions for Grid scheduling and management in Next Generation Grids. This includes the architectural perspective which has to support scalability and cooperation across different administrative domains. Such architecture requires new scheduling algorithms and scheduling policies for coordinating the access to resources and for supporting complex job requirements. In addition, the actual management of resources and jobs needs several services that provide the necessary functionality e.g. to manage the scheduling of complex job workflows.

2.5.1 Research Challenges

The main research challenges in Resource Management and Scheduling are the following:
• **Creation of a common model for Grid scheduling** that can be used as a framework to integrate different algorithms and strategies while providing interoperability and consistency between different implementations. Such a model supports theoretical analysis as well as the advancements in the practical development of Grid management solutions.

• As there are several types of Grid scenarios, there is need for specific Grid scheduling strategies. These strategies have to take into account the special requirements in Grids. That is especially the consideration of multi-domain Grids with independent service providers. In this scenario no Grid broker or scheduler can assume full control about all resources. Thus, strategies are required that are suitable for multi-level scheduling, integrating Grid scheduling, resource scheduling and application scheduling.

• **Support of Grid business models** is a key requirement for future Grid systems. Here, the Grid resource management has a vital role to facilitate such business models. This includes the consideration of service level agreement as a basis for reliable resource use as well as a pre-requisite for later cost modeling. The integration of costs and agreements adds another dimension in the planning and managing process as now different scheduling objectives have to be considered. Users will require cost consideration as an important aspect in the resource selection strategy that does not allow the simple consideration of performance issues, like response time optimization, but now includes multi-criteria optimization including cost, performance, or quality. In general, Grid economics is considered a crucial field for the success of Grids in business related environments. With the inclusion of Grid economics, cost modeling as well as reliable accounting become important aspects of Next Generation Grids.

• Most of Grid resource management systems focus on particular resource types, mostly only compute resources, or data or networks. Even for these systems, the different resources are independently considered. However, it will be essential for the success of Grids to provide common solutions to integrate the management of different resource types in a common management and coordination framework. Here, it is clear that no single strategy will suit all application scenarios. However, there is a need to support the coordination of different resources. Therefore, suitable models and strategies are required that provide the integration of the different resource types. A currently important example is the coordination of data management with compute and network resources.

• While first Grid solutions focused on the simple allocation of single resources, it is clear that due to the distributed nature a more sophisticated workflow management is essential. Many Grid related activities require the coordination of several resources in complex workflows (e.g. combination of data, storage, network and computational resources) with resource and control dependencies. Here suitable management solutions are required to plan the resource utilization and orchestration of the resource allocation for the tasks.

• As Grids are dynamic by nature and the availability of resources may change constantly, reliability and fault-tolerance are important aspects that need to be addressed for several application scenarios. To a major part, these issues will have to be managed automatically as it cannot be expected that the end-user or the application itself can alone cope with the constant monitoring and dynamic adaptation to changes. Here, the Grid resource management has to provide suitable means to facilitate automatic reallocation of tasks to resources if the Grid situation changed or the application does not behave as expected. This task needs suitable monitoring functionality but also planning and orchestration logic to analyze the impact on the Grids performance and suitable counter-measures.

• **Benchmarking and performance evaluation** are important to estimate the suitability of models, algorithms and applications in practical Grid scenarios. While there are some existing solutions for specific aspects in parallel and distributed systems, there are yet no established solutions that cope with the different aspects of Grids. This is partially caused by missing data about practical Grid use in real-world scenarios. Thus, suitable models need to be identified and evaluated for their validity and practicability. Moreover, suitable tools are necessary to perform such benchmarking to gain information on system performance. Such information are also necessary to support performance prediction which can be useful for the scheduling and management process if no detailed data about resources and applications are available.

### 2.5.2 Research activities

To achieve the objectives and to contribute to solve the research challenges described above, the activities of the RMS Institute are coordinated around seven main directions, which are presented in more details in the following sections.

#### 2.5.2.1 Definition of a Grid Scheduling Architecture

CoreGRID – Network of Excellence
Next generation Grids will provide a large variety of complex services and resources. Coordination of those services requires an extensible and integrated scheduling concept. Such a coordinated scheduling of services is currently not available. However, there are some specialized solutions for dedicated application scenarios. It requires a different approach than known from present distributed systems as the resources in Grid systems (often) do not belong to the same administrative domain. Consequently, the individual demands of the resource owners need to be observed. Taking them into account requires a new technological approach. Therefore, Next Generation Grids need a scheduling architecture that supports the interaction of the independent local management services with higher-level scheduling services. Such architecture manages access to the various Grid resources, which are typically subject to individual policies of the resource owner for access, accounting, priority, and security. In addition, the user of a Grid may also establish individual scheduling objectives. The scheduling architecture must thus support negotiation of resource usage based on these policies to allow the implementation of various different business models. Within this research direction, the RMS Institute plans to review existing Grid scheduling approaches before designing a common CoreGRID Grid scheduling architecture. This scheduling architecture will later integrate some of the research results from CoreGRID partners to build or to support this common architecture. For instance, the Institute plans to explore the use of P2P techniques in its design. Another issue to be explored is the integration of service level agreements within the scheduling architecture. A gap analysis will then be performed to identify the necessity for future research and development activities.

2.5.2.2 Multi-level Scheduling Strategies

The Grid scheduling structure requires scheduling algorithms to be developed at a higher level of abstraction to take the different system levels into account. Most resources are controlled by local resource management systems. Higher-level Grid scheduling instances will have to interact with these local scheduling systems. That is, scheduling for Grids will significantly differ in comparison to parallel job scheduling. For instance, only limited information may be available from the local resource management systems. In addition, the consideration of individual resource policies and user demands will lead to complex multi-criteria scheduling objectives. From the user’s point of view, the optimization objective might deviate from the common response time minimization. In a Grid the user will also have individual preferences about service-quality, cost, etc. To this end, new scheduling models will be explored. Reservation and negotiation models are identified as important components in the scheduling process and will thus be studied.

2.5.2.3 Workflow Scheduling Strategies

Future Grid jobs will consist of complex workflows with temporal dependencies between different resource requirements. That is, scheduling will not only include sequential and parallel jobs but also dependencies between subtasks/sub-jobs. These dependencies can indicate parallelism, control or data dependencies. The efficient execution of such workflow will eventually require advanced job scheduling for such complex workflows. The Institute will examine new job scheduling strategies to e.g. estimate the deadlines of job parts and maintain resource reservations to guarantee efficient job execution. Current scheduling strategies do not exploit all available workflow information. Many different strategies and formalisms exist for the different kinds of workflow models. This includes resource orchestration or choreography, as well as graph vs. script-based workflow models. Similarly there exist several different specification languages for the description of workflows. The Institute will review existing Grid formalisms and workflow scheduling approaches and will propose to integrate some of them within the common scheduling model.

2.5.2.4 Evaluation and Benchmarking of Grid Scheduling Systems

The efficiency of new Grid scheduling protocols and strategies must be evaluated and compared. To this end, the creation of a simulation environment and the definition of benchmarking methods are planned. The evaluation requires realistic conditions and the examination of the complete scheduling process. As the user demand for Grids and subsequently the Grid workload is not yet known, existing models for workload models are evaluated and extended to the Grid environment. Available applications and performance data from Grid projects will be examined to provide us with a good benchmarking suite and data. Moreover, the results are verified in Grid testbeds that are available to CoreGRID. As there exist different simulation environments, it will be reviewed whether it is possible and sensible to work towards a combined Grid simulator.

2.5.2.5 Mapping and Scheduling of HPC Applications
Next generation Grids will have to support many different application scenarios and policies. These application scenarios will not be limited to computational resources. Thus, the creation of a Grid scheduling architecture and corresponding strategies need to be designed very generic and extensible. This includes support for cost management or the co-allocation of different resource types which may not occur in all Grid scenarios. However, high performance applications, and especially parallel applications, are and will be a very important class of applications that are executed on Grids. These applications have special characteristics and requirements. This applies for sequential jobs with high-throughput characteristics but even more for parallel jobs. Therefore, special scheduling and management mechanisms are required which provide high efficiency for these applications. Mapping and scheduling methods should be customized with respect to classes of structurally similar parallel applications. After reviewing existing scheduling methods, the RMS Institute will develop a set of robust scheduling mechanisms to improve reliability and fault-tolerance. It will also study the integration of these mechanisms within the common scheduling architecture.

2.5.2.6 Coordination of Grid Scheduling and Data Management

Data constitutes a central resource in Grid environments. Most Grid jobs require access to data during their execution, and the data sets used by current Grid applications can be huge. However, data as a Grid resource is different from most other resource types as data can be transferred, replicated and altered. The use of data results in storage, replication, transport, and network traffic problems. Therefore, the management of data must be coordinated with the management of other physical resources such as processors. In addition, a wide range of constraints must be specifically considered in data management, such as data locality, data persistence, and data transfer costs. Moreover, a Grid will have no centralized control and will be subject to frequent configuration changes. All of these aspects must be considered in the general design of the scheduling algorithms as well. To this end, data management has to be an integrated component of a Grid scheduling architecture and of Grid scheduling algorithms. A Grid scheduler needs to consider data locality into his resource selection and to analyze the benefit of transferring some data to and from a particular resource. The plan of the researchers in this Institute is to survey existing coordinated job and data management functions and to identify required interfaces to data and network management. The Institute will propose a set of coordinated job and data scheduling strategies to be integrated into the common scheduling architecture.

2.5.2.7 Performance Prediction

The performance prediction of Grid jobs is an important component for efficient scheduling as information about the execution time of a Grid job is required to plan resource allocations in advance. With different types of resources and different kinds of service quality, the performance analysis is a complex task. While many information about the current status of the Grid can be deduced or statistically forecasted from a Grid information and monitoring system, many information about the prediction of a job rely on tight interaction with the higher-level Grid scheduling strategy and the lower-level scheduling of the local resource management system. Therefore, it is the objective of this research activity to analyze and develop scheduling related prediction mechanisms. To this end, the Institute will define new performance models for Grid jobs. Based on these models, strategies for predicting the application performance with respect to the available resource situation will be examined.

2.5.3 Partners contribution to the research Roadmap

The RMS Institute involves 20 CoreGRID partners including 82 researchers and PhD students. It has to be noticed that 2 other CoreGRID partners (INRIA and UNIPD) will contribute marginally to the RMS research activities and thus are not listed here. Within the RMS Institute there is also one institution, which is not a CoreGRID partner (Wroclaw Institute on Networking and Supercomputing – WCNS), that is involved in three research groups. This partner has been invited by CoreGRID partners to participate to these research groups. The roadmap of the RMS Institute will be implemented thanks to 20 research groups as listed in the following table:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Research Groups</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CoreGRID – Network of Excellence
### 2.6 Grid Systems, Tools and Environments (STE)

The research activities of the STE Institute are focused around the development of the design methodology for a generic component system that integrates application components, tools/system components, PSE/portal components, and infrastructure components. The adopted approach of the Institute encompasses both the application of the component model, and its integration into a tools framework. The specification of the Institute's generic Grid platform will advance significantly the state-of-the-art enabling the design of component-based Grid systems for both applications and tools/systems/environments to have a single, seamless “invisible” Grid software infrastructure. The results from this research are expected to be equally valid for both the client/server and the Peer-to-Peer paradigms.

The adopted approach of the STE Institute enables the design of re-configurable Grid systems with dynamic properties guaranteed by our component-based framework. Thus, our flexible Grid platform is expected to be suitable for a wide range of devices from portable PDAs to parallel supercomputers. Beyond the contributions of the partners of this Institute, there exist many individual approaches for building Grid systems, tools and environments by researchers from all around the world.
2.6.1 Research Challenges

The main research challenges in the design of Grid systems, tools and environments are the following:

- One of the shortcomings of the existing and contemporary Grid systems is their inflexible support for scalability and adaptation; being purpose specific or being bound to infrastructures, they prevent immediate adaptation for a given context.
- Another problem the STE Institute is addressing is the combination of applications and system software to integrated software assemblies. The aim of this integration is twofold: (1) allow applications to better and more flexibly use Grid resources, and (2) allow system services to steer and control their applications.
- The development of applications to be run on the Grid is difficult and this prevents the major adoption from non-expert users. The challenge in this case is to provide programming environments for Grid-unaware applications. By Grid-unaware, the researchers in this Institute understand applications where the Grid is transparent to them but that are able to exploit its resources. Furthermore, the challenge is to increase the performance of these applications when possible.
- Frequently, applications have to be adapted to be run on specific hardware platforms. Even more, to improve performance figures the applications have to be specifically tuned for each hardware platform. This situation is not feasible in Grid computing environments, since the execution platform can change in a per-job basis, and even at run time. In this kind of environments, the applications have to adapt to the systems and resources available.
- This Institute intends to design and build an integrated component system that has a single glue layer between all forms of components. Thus, most of the complexity of integrating the many individual software items will be removed.

2.6.2 Research activities

To achieve the objectives and to contribute to solve the research challenges described above, the activities of the STE Institute are coordinated around four main directions, which are presented in more details in the following sections.

2.6.2.1 Generic Platform

This research task aims at developing the common aspects of a generic, component-based platform. This platform will specify the basic “glue” from which higher-level components and services can be constructed. The task will focus on the challenges arising from the need to support heterogeneous software libraries, toolkits, operating systems and existing Grid frameworks in its approach. In particular, architecture, interoperability, security, and design methodology will be developed. The methodology work investigates different component models, enabling and existing technologies, features to be implemented or to be included, and case studies illustrating how the platform would handle real-world scenarios. This platform will be based on the Grid Component Model as defined by the PM Institute. Our platform is generic and lightweight, which could scale automatically upwards and downwards based on the context of use. The Grid component model supports flexible component adaptation statically and dynamically. The component framework facilitates designing the platform architecture, support extensibility of different aspects of the platform and allow easy to use and consistency-preserving reconfigurations.

2.6.2.2 Mediator Components

In the context of this second research direction, a suite of components will be developed that mediate between applications and system components, providing core services according to the component framework. These include such capabilities as application steering, meta-data retrieval and service discovery. The components delivering these capabilities will be integrated and exposed via a runtime environment that acts as a gateway to system components. With respect to their functionality, their implementation, and their integration into an overall Grid component platform, the proposed mediator components can be categorized into three categories. Firstly, the application-level components are intended to be integrated directly with the application code, and the corresponding process environment at runtime. Secondly, the service-level components are closely related and provide the functionality of Grid services. Thirdly, the meta-data components provide access to meta data, of both application and of other mediator components.
2.6.2.3 Integrated Toolkit

The third research direction aims at developing a toolkit to simplify the deployment of Grid-unaware applications while optimizing the performance of the application. The objective is the specification and design of an integrated toolkit for the generic Grid platform. This integrated toolkit will enable the development of Grid-unaware applications. However, by means of the integrated toolkit these applications can be run on the Grid and exploit its resources. The proposed goal for the integrated toolkit activities is to follow a components-based approach. An important aspect that will also be studied is the support for efficient communication between components. With our integrated toolkit, applications are programmed using a sequential control flow language that provides a simple and clean definition of the basic objects dealt with by the programmer. An automatic runtime system identifies the tasks that compose the application, detects task precedence and decides which tasks can be/should be submitted to the Grid. The runtime system checks whether remote execution on another Grid site is worthwhile, exploits the tasks concurrency and controls the distributed execution.

2.6.2.4 Advanced Tools and Environments for Problem Solving

The goal of this research task is to specify the design of a framework that allows component-based PSEs and portals to compose, expose, and monitor legacy and non-legacy applications. The spectrum of activities that could potentially fall under this research direction is quite broad. To narrow this area of research and thereby produce meaningful results, a set of use-cases will be developed to identify the following three general categories as most relevant to the Institute's research. Firstly, P2P enabling technologies to build hybrid systems that can integrate mobile, P2P and Grid architectures with commodity desktop Grids, bringing new levels of resource flexibility to traditional distributed systems. Secondly, workflow-centric programming environments, which allow us to combine potentially complex application logic into a visual representation, enabling users to easily construct workflows in either a heavy- or light-weight editor and then manage and launch them on the Grid. Thirdly, the legacy code wrapping as a process by which a legacy code is Grid-enabled by “wrapping” the code in another program (or script) to provide additional functionality, allowing us to use an unmodified application in a Grid environment.

2.6.3 Partners contribution to the research Roadmap

The STE Institute involves 13 CoreGRID partners including 80 researchers and PhD students. The roadmap will be implemented thanks to 9 research groups as listed in the following table.

<table>
<thead>
<tr>
<th>Grid Systems, Tools and Environments</th>
<th>IPP BAS</th>
<th>ICSFORTH</th>
<th>ICS</th>
<th>INRIA</th>
<th>SZTAKI</th>
<th>UWC</th>
<th>UNILE</th>
<th>UPI</th>
<th>UWC</th>
<th>CYFRONET</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design Methodology for a Generic Component Grid Platform</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Security in Grid platforms</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integration of Information Sources into an Application-Level Information Cache Mediator Component</td>
<td></td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application Deployment Component</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Integration Programming and Runtime Environments</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Efficient Component Composition</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Legacy Code Wrapping and Deployment Methodologies</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grid Applications Use Cases and Scenarios</td>
<td>✔</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Grid Portals</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3 High-level scientific positioning

The following sections describe the positioning of the CoreGRID research activities against other similar activities conducted within the context of the Grid research landscape.

3.1 Knowledge and Data Management

Research activities in the area of data Grids and knowledge Grids are being pursued in Europe, in the USA and in Asia by several research teams and, at the same time, big companies such as IBM, HP, ORACLE, and SUN Microsystems, are very active in the area. This demonstrates the key role of data management in Grids and the importance of developing knowledge-based applications that exploit the Grid features to achieve high performance and high availability.

The research tasks undertaken in the context of the KDM Institute compose a unified scenario of the data and knowledge management in Grids through a layered approach that starts from efficient data storage techniques up to information management and knowledge representation and discovery. At the same time, joint research activities pursued by the Institute partners are providing single solutions for data and knowledge management that will bring benefits to research and industry in Grid technology.

In the area of distributed storage management various projects in Europe, the USA and Asia currently strive to address issues such as building scalable storage systems that can hold petabytes of storage in a cost-effective manner and making the storage infrastructure location-independent and client-agnostic in an efficient manner. However, there are no common methods for describing storage services realized at various levels of abstraction by different elements of the storage infrastructure, and there is no common approach to the problem of virtualizing the storage resources without losing their specific, interesting features.

Research activities in the area of information and knowledge management are actively pursued all over the world; however European researchers and projects are playing a leading role. Projects like OGSA-DAI and OGSA-DQP produce Grid middleware and services that are largely used in Grid database management. Both research teams that worked on those systems are actively involved in the KDM Institute. Other important European projects such as DataGRID, DataTAG, OntoGRID and the UK e-Science project produced high-quality research in this area that is fueling the Institute's research work.

Finally, the number of high-level instruments to support the knowledge discovery and management in distributed environments is very low. This is particularly true in Grid-based knowledge discovery, although some research and development projects and activities in this area are going to be activated mainly in the USA and Europe. In particular, in Europe systems have been designed in the context of several projects such as the KNOWLEDGE Grid, the Discovery Net, the AdAM middleware and the DataMiningGrid. The KDM Institute is contributing also in this area and is cooperating with other teams and projects such as KDUbiqu.

3.2 Programming Model

The Programming Model Institute research activities are aimed at investigating suitable features of component based programming models. The research activities of the Institute are positioned in an international research and development framework that includes several “component based” or “component oriented” activities.

The two major initiatives in the framework of component based architectures are CCA (Common Component Architecture) and CCM (CORBA component model). Besides these initiatives that are both related somehow to the Grid environment, several other initiatives can be associated to component based programming models, including Microsoft .COM, Sun JavaBeans and, to a certain amount, the recent Microsoft .NET.

The CCA initiative, mainly contributed from US researchers, is aimed at “defining a standard component architecture for high performance computing”. (http://www.cca-forum.org/overview/index.html). Although the initial specification did not involve directly Grids, a consistent part of the recent work in the CCA community has been dedicated to target Grid architectures. The CCA architecture relies on the definition of several characterizing items such as Scientific IDL (SIDL) to define component interfaces, CCA component repositories...
implemented following a CCA Repository API, a CCA Framework Service interface stating the rule to access framework services and a CCA Configuration API basically ensuring that components can collaborate with different frameworks. A CCA component is “a software object, meant to interact with other components, encapsulating certain functionality or set of functionalities. A component has a clearly defined interface and conforms to a prescribed behaviour common to all components within an architecture. Multiple components may be composed to build other components”. The Programming Model Institute Grid Component Model (GCM) completely inherits and exploits the compositionality concept from CCA.

The CCM initiative is a fork of OMG (the Object Management Group consortium) that has members from industry and academia from US as well as from European Countries. It is a “server-side component model for building and deploying CORBA applications” (http://www-128.ibm.com/developerworks/webservices/library/co-cjct6/index.html). Open source implementations of the standard are maintained by European entities (OpenCCM http://openccm.objectweb.org), CCM provides its own Component Implementation Definition Language (CIDL) as well as a Component Implementation Framework (CIF), which defines the programming model for constructing component implementations and a container programming model. It also adds to components event interfaces, that is sort of ports that react according to an event driven interaction model rather than according to the more usual RPC/RMI model. Again the GCM inherits concepts from the CCM framework. Apart from many useful concepts related to the implementation of the GCM component framework, the concept of multiple interactions mechanisms (e.g. facets/ports and events) developed in CCM is further exploited in GCM where both ports, events and data (streaming, one way, one to one as well as many to many) ports are included.

Other kinds of initiatives, such as the Microsoft ones or the Sun JavaBeans are more loosely related to the activities of the Programming Model Institute, mainly due to the very specific contexts they address. Overall, the activities of the Programming Model Institute can be stated to be following the state-of-the-art technology improvements available in the component based distributed/parallel/Grid programming environments while trying to improve the general concept of component by adding specific features (autonomic component controllers, composite components as first class components, several peculiar component ports supporting specific, high performance communication patterns and, last but not least, full range of interoperability options including the possibility to invoke services from components and to wrap components as services) that make the component model highly suitable for Grid programming.

3.3 System Architecture

In the following we present a snapshot of the current state of technology in the Grid architecture research with specific focus on the three main key aspects, namely: scalability, adaptability, and dependability.

Grid and P2P are both resource-sharing systems having as their ultimate goal the harnessing resources across multiple administrative domains. They have many common characteristics such as dynamic behaviour and heterogeneity of the involved components. Apart from their similarities, Grid and P2P systems exhibit essential differences reflected mostly by the behaviour of the involved users, the dynamic nature of Grid resources (i.e., CPU load, available memory, network bandwidth, software versions) as opposed to pure file sharing which is by far the most common service in P2P systems. Another essential difference results from the demanding nature of sensitive Grid applications that are time and data critical and have strict fault tolerance and security requirements as opposed to P2P applications, which use commodity hardware and exhibit best effort behaviour. However, Grid systems can benefit from P2P technologies in order to have more scalable services capable of managing a large amount of resources. Recently several Grid systems have been proposed that incorporate techniques from the P2P paradigm in their resource discovery techniques. Many of these systems are based on the unstructured P2P approach while others borrow techniques from existing structured P2P systems (MAAN, NodeWiz, SWOR, Mercury, etc.).

Contributions to adaptability and self-management of resources within Grids, services and systems have been made under several industrial research initiatives and within traditional university research. The best-known example is the Autonomic Computing initiative, which comprised efforts to refine already known management techniques, as well as efforts to develop new methods. The initiative focuses mainly on the self-management of systems on hardware- and operating system level. Examples of research in this field include management of heterogeneous networks, prediction of resource demand and automatic adaptation, and automatic recognition of resource models. A special field of adaptability/self-management is devoted to modelling and prediction of system behaviour, as exposed by metrics like workload, performance change, resource usage and others. These
approaches play an important part in scheduling (long-term: capacity planning, short-term: resource sharing) and in fault-resilience (anomaly detection, proactive software rejuvenation). Currently, no major Grid middleware utilizes tools for modelling and prediction of system characteristics. This situation is partially due to the lack of implementations of suitable algorithms. While frameworks for short-term prediction and scheduling support for individual servers do exist, they are not appropriate for long-term prediction, anomaly detection, or exploiting the correlations between the applications in a cluster. A further impediment comes from the fact that in current Grid architectures, tools for demand modelling and prediction have not been foreseen as a part of the middleware, and so other components such as schedulers etc. cannot take advantage of them. Finally, there is lack of a suitable standard for model description and exchange.

With the advent of Grid computing there is a clear need to adapt the fault-tolerance schemes to scalable, dynamic and wide-area environments that may comprise heterogeneous modules and different Grid middleware. Existing Grid middleware systems are not reliable. For instance, the Globus Grid service container does not provide any means to achieve that reliability. This has to be dealt with by each particular service. The middleware that runs inside a cluster of the Grid may also lack support for fault-tolerance. As an example, MPI-based implementations are usually unreliable. In order to end successfully an MPI application all the involved units must run smoothly and any unexpected individual failure results in an application breakdown. Some fault-tolerance schemes for MPI have been proposed in the literature and there is still some work to do in this field. Similar problems can also be considered for job-batching middleware: if there is a breakdown, some mechanisms should exist to keep the partial results either to restart the job from a checkpoint or to allow the analysis that eventually would lead to identification of failure causes. All these modules of middleware should provide different mechanisms for fault-tolerance. The most difficult goal is to make these mechanisms more tightly integrated in order to provide full-dependability at the application level. The two most frequent causes of failures in Grids are due to configuration problems and middleware failures, followed by application errors and hardware outages. Another curious fact observed was that solutions for failure-handling are mostly application-dependent, which have been requiring a large effort from application programmers to diagnose and provide error-recovery code able to resume the application after the occurrence of a failure.

3.4 Grid Information, Resource and Workflow Monitoring Services

A large-scale heterogeneous Grid is subject to frequent changes and disruptions in the service of a huge number of components it is built of. To manage such a dynamic system and its resources, online monitoring of the resources to determine their availability is required. Without this information, the challenging goals described in the Next Generation Grid(s) report cannot be achieved. Despite the recent progress in this area, the level of architectures and concepts (like the Grid Monitoring Architecture) is still either too high or there are too specific implementations, like MDS, R-GMA, Mercury, or iGrid, which are not yet able to provide a reliable information and monitoring system for a general heterogeneous large scale Grid. The challenge of network monitoring is that it must face a task which, in principle, grows with the square of the dimension of the system. In addition, Network Monitoring is characterized by peculiar security problems. There are ongoing activities in a number of next generation network projects that intend to provide interfaces between the network layer and the Grid middleware, e.g. EU projects GEANT2 and MUPBED.

Due to historical reasons the checkpointing mechanism is not widely available. Only few operating systems are supporting this feature (IRIX, UNICOS). Additionally there are a few projects that provide checkpointing packages for other platforms. Almost all available checkpointing mechanisms impose some limitations on programs that have to be restored. Another approach is to deliver an interface to resource management systems able to call a C/R function, e.g. Sun Grid Engine (SGE) has an interface that allows using the checkpointing facility on nodes which provide this functionality. Further, the Condor project is shipped with a proprietary user-level checkpointing library and allows submitting jobs that are to be checkpointable. Currently the Grid virtually lacks in well-developed checkpointing-related services and interfaces. As this problem has been noticed by the Global Grid Forum, a Grid Checkpoint Recovery (GridCPR) Working Group has been established in which several CoreGRID researchers participate (VUA, INRIA). The group aims to define the user-level API and associated layer of services.

Using workflow has gained interest in the Grid community because it provides a high-level abstraction for application composition. The application of workflows for the composition of Web and Grid services has resulted in several frameworks. Due to the dynamic nature of the Grid, workflow composition is a challenging task because the system has to deal with resource unreliability and unpredictability, which are closely related to

CoreGRID – Network of Excellence
fault tolerance and scheduling. Unfortunately, these issues have not been addressed to by most of the related work on Grid-based workflows. Most of this work assumes no faults on the Grid and poor or no information provided to the system to deal with Grid dynamics. Purely graph-based workflow descriptions have been proposed (e.g. for Symphony, Condor’s DAGman tool, and Grid Workflow Execution Service – GWES) which are mostly based on Directed Acyclic Graphs (DAGs) or Petri Nets. Examples of script-based workflow descriptions are GRIDAnt. Another commonly used script-based approach to describe workflows is the Business Process Execution Language (BPEL) and its recent version for Web Services (BPEL4WS) that builds on IBM’s WSFL (Web Services Flow Language) and Microsoft’s XLANG (Web Services for Business Process Design). BPEL is intended mainly for modelling and implementing business processes. Further comparisons of workflow description formalisms are available at the Grid Workflow Forum (http://www.gridworkflow.org/), which has been set up by CoreGRID partners.

The problem of managing user accounts becomes a non-trivial one in a distributed environment (especially in a Grid environment) that includes many independent sites and virtual organizations with hundreds or even thousands of user accounts. The complexity arises from the point of view of time required for administration tasks and automatization of these tasks. The accounting issue becomes impossible in the distributed environment, but still possible on a single computer system. The next important problem is accounting in distributed systems. The existing solutions on the market allow the accounting of resources used for only one system or at most for local and homogeneous clusters. The problem arises in Grids when the environment is more complex by adding virtual organizations and dynamically assigned accounts. The GGF created the Accounting Models Research Group, whose goal is to work out the rules of data exchange and a general communication interface between sites, allowing information to be collected about resources used in Grids. The problem becomes more complex the greater the environment is, i.e. the number of VOs and the number of users accessing resources. Obviously, the accounting mechanism should provide us with some information independently of the service scale and range (a single system, local cluster or Grid environment). The possibility of accounting the resources used in Grids forms the basis for introducing the Grid economy concept.

### 3.5 Resource Management and Scheduling

While Grid scheduling is generally accepted as a crucial feature for Next Generation Grids, there are currently not many accepted solutions in this area. Instead there are several implementations for specific projects. The lack of solutions is caused by the complexity of resource management and coordination and its reliance on the availability of many other Grid functionalities and services (information, monitoring, resource/job description, execution management etc.) Most prominent as the foundation of many Grids is the Globus Toolkit, which does not provide any planning and brokerage functionality. In that context, it is generally regarded that planning is an application specific requirement that is not part of the Globus Toolkit. The Community Scheduling Framework (CSF) has been developed from Platform Computing and was contributed to the Globus Toolkit. It supports the implementation of Grid schedulers in conjunction with the Globus Toolkit and Platform LSF. However, not many applications are known to facilitate this framework. Within the Japanese Naregi project, several Grid scheduling solutions have been developed; most prominent is the Naregi superscheduler. The solutions provide several effective functionalities; however, the application is quite specific to the Naregi infrastructure and is not in wide use beyond the project. Similarly, within the Korean K-Grid a Grid resource management and scheduling solution has been conceived, the K-Grid scheduler. There are also several approaches in Europe, like the GRMS solution from the GridLab project, the WMS from the LCG effort. In general, all the available solutions support a subset of the requirements of Next Generation Grids. For example, the lack in the support of different scheduling strategies, workflow planning, orchestration of different resource types, data management, or economic models.

To this end, the Global Grid Forum (GGF) provides several activities in the area of resource management and planning to address the interoperability issues. This includes the OGSA model with its specific working groups on Execution Management Services, like OGSA-RSS, OGSA-BES, OGSA HPC Basic Profile. There are also relevant activities in the GGF beyond OGSA, in GRAAP (providing WS-Agreement), in the Grid Scheduling Architecture research group (GSA-RG).

Therefore, the Institute in Resource Management and Scheduling actively collaborates in these interoperability and design efforts in the GGF. Moreover, it moves forward the joint work in providing the foundation for efficient resource management solutions that integrate the different requirements.
3.6 Grid Systems, Tools and Environments

The specification of our generic Grid platform will advance significantly the state-of-the-art enabling the design of component-based Grid systems for both applications and tools/systems/environments to have a single, seamless “invisible” Grid software infrastructure. The results from this research are expected to be equally valid for both the client-server and the Peer-to-Peer paradigms. Our adopted approach enables the design of re-configurable Grid systems with dynamic properties guaranteed by our component-based framework. Thus, our flexible lightweight Grid platform is expected to be suitable for a wide range of devices from portable PDAs to parallel supercomputers. Beyond the contributions of the partners of this Institute, there exist a number of efforts and approaches for building Grid systems, tools and environments by researchers from all around the world. The STE Institute maintains working contacts and keeps updating its links with relevant research activities in Europe and worldwide.

The Model Driven Architecture (MDA) is an approach to application design and implementation (http://www.omg.org/mda/). It is based on the use of system models in the software development process. The MDA provides a way to organize and manage enterprise architectures supported by automated tools and services for both defining the models and facilitating transformation between different model types. In MDA, systems and platforms are specified independently and when a particular platform is chosen for a system the system specification is transformed for that particular platform. However, it is important to note that MDA (as UML) provides only a model, but not an implementation.

The GridSphere portal framework (http://www.gridsphere.org) has been developed in the European GridLab project, which allows projects to easily create customized portals as needed. GridSphere provides a flexible and standards-compliant (JSR 168) portal framework for implementing modular, reusable software components (portlet), which can interact with other compliant containers, such as BEA Weblogic, Sun One portal server, and IBM Websphere. Gridsphere has visual components providing extensive support for Grid operations, such as handling resources and certificates, launching jobs and transferring files, and provides a user-friendly interface to hide and reduce the complexity of typical Grid systems today and empowers people unfamiliar with the workings of the Grid.

AliCE (http://www.comp.nus.edu.sg/~teoym/atsuma.htm) is a lightweight Grid middleware facilitating aggregation and virtualization of resources. The modularized, object-oriented nature of its implementation supports possible extensions and varying levels of QoS, monitoring and security. The ALiCE architecture consists of multiple layers with the lowest, core-layer, providing resource discovery and system management while the second layer supports application development and deployment. However, the adopted approach is not generic and dynamic extensibility is a challenge as the project relies on an object-oriented approach as opposed to a component-oriented approach.

The Common Component Architecture (http://www.cca-forum.org/overview/index.html) specifies the means for interaction among components. In CCA, components interact using ports, which are interfaces pointing to method invocations. The framework provides services such as connection pooling and reference allocation. Dynamic construction and destruction of component instances is also supported along with local and non-local binding. Though CCA enables seamless runtime interoperability between components, one of its main weaknesses is the lack of support for hierarchical component composition and for control mechanisms thereof.

The reference model of the Enterprise Grid Alliance (EGA) is an effort to derive a common specification adopting Grid technologies for enhancing enterprise and business applications (http://www.gridalliance.org/en/WorkGroups/ReferenceModel.asp). The model, which is aligned with industry-strength requirements, relies on component technology adopting a top-down approach in building Grid systems whereas the STE Institute follows a bottom-up approach. One of the key features that the reference model suggests is the life-cycle management of components.

The CORBA Component Model (CCM) is a language-independent, server-side component model, which defines features and services to enable application developers to build, deploy and manage components to integrate with other CORBA services. The CCM supports grouping interconnected components into assemblies. However, in contrast to the CoreGRID GCM proposal, these CCM assemblies are not components and cannot be further composed – thus CCM does not permit hierarchical composition.

A Service-Oriented Architecture (SOA) is defined as a collection of decoupled services, where these services communicate with each other through well-defined interfaces (http://www.service-architecture.com). In a SOA...
framework the applications are no longer monolithic ones that run on a single server. Instead, they rely on one or more platform and language independent services deployed somewhere on a server, that can be invoked, reused, and shared in a dynamic fashion. SOA is a conceptual framework: CORBA can be considered a SOA system, and Web Services, OGSA or WSRF are other examples. The work of the STE Institute also conforms to the SOA framework.

The recently announced Networked European Software and Services Initiative (NESSI) is an industry-driven effort aiming at providing a unified view for European research in service architectures and software infrastructures that will define technologies, strategies and deployment policies fostering new, open, industrial solutions and societal applications (http://www.nessi-europe.com/). The proposed NESSI technological platform in particular is relevant to the research agenda of the STE Institute. Some initial contacts with NESSI members have been created with the view of possible collaboration and adoption of our components-based approach into the development of the NESSI technological platform.

4 Conclusion

We have presented in this document the annual scientific roadmap for the six research areas. It is the result of a collective effort by the Scientific Coordinator and the Institute leaders aiming at describing the research activities pursued by CoreGRID researchers in a self-contained and structured way. However, more detailed information of the Scientific Roadmap for each of the six institutes may be found in deliverables D.KDM.03, D.PM.03, D.SA.03, D.IRWM.03, D.RMS.03 and D.STE.04.