Advanced Techniques for Scheduling, Reservation, and Access Management for Remote Laboratories

Philipp Wieder
Research Centre Jülich
Central Institute for Applied Mathematics
52425 Jülich, Germany
ph.wieder@fz-juelich.de

Oliver Wäldrich, Wolfgang Ziegler
Fraunhofer Institute SCAI
Department of Bioinformatics
53754 Sankt Augustin, Germany
{oliver.waeldrich, wolfgang.ziegler}@scai.fraunhofer.de

Abstract

In Grid and Service Oriented Architecture-based environments the use of technologies supporting Service Level Agreements, advance reservation, and Virtual Organisations mark the start of a new way accessing distributed resources or services. Experience made so far indicate the potential for overcoming the limitations of the hitherto best effort approaches. In this paper we propose an approach that integrates remote laboratories as another resource into the Grid. We show how techniques developed for compute, storage, or network resources in Grids may be used for managing access to Collaborative Remote Laboratories. In particular we discuss how existing technologies for advance reservation, Service Level Agreements, and Virtual Organisations can help to set up, configure, and manage remote laboratories and provide support in the preparation of collaborative control of and mutual access to laboratory equipment.

Keywords: *Collaborative Remote Laboratories, Virtual Organisations, Service Level Agreements, Advance Reservation, UNICORE, Grid*

1 Introduction

Managing access to Collaborative Remote Laboratories (CRL) has several aspects in common with managing the access to and usage of remote compute resources. However, while for the latter Grid infrastructures have been developed and are deployed today in various testbeds and some production-oriented environments, like for example EGEE [10] or DEISA [7], only a limited number of experiments have been carried out to make use of the Grid infrastructure for accessing and using remote laboratories. The most obvious common aspects are

- advance reservation to allow usage at a defined point in time,
- access control based on roles,
- negotiation of the usage conditions (i.e. Quality of Service), and
- Service Level Agreements as reliable contracts between service providers and users.

In Grids and environments based on the Service Oriented Architecture (SOA) paradigm, the use of Service Level Agreements (SLAs), advance reservation (AR) techniques, and the Virtual Organisation (VO) concept have been the advent of a new way to access distributed resources and services [13]. Experience made so far indicate the potential of these concepts for overcoming the limitations of the hitherto best effort approaches. Similar to expensive high performance computing facilities, laboratories are shared for experiments by collaborative research groups who are working at different distant locations thus both reducing the costs for the individual members and reducing the amount of experimental data to be transferred and analysed afterwards. In this paper we present an approach that is taking up previous work aiming at integrating remote laboratories

Proceedings of the Second IEEE International
Conference on e-Science and Grid Computing (e-Science'06)
0-7695-2734-5/06 $20.00 © 2006 IEEE

as first class resources into the Grid. We show that techniques developed for the usage of compute, storage, or network resources in a Grid context may be used for managing access to CRLs. We discuss in particular how existing Grid technologies for advance reservation, SLAs, and Virtual Organisations can be used to set up, configure, and manage remote laboratories. Based on this discussion we present an architecture to support CRLs in the preparation of collaborative control of and mutual access to laboratory equipment. Integrating CRLs into the Grid infrastructure immediately allows to orchestrate laboratories together with other resources, e.g. compute, storage, and network resources thus allowing to set up a problem solving environment (PSE) of a new dimension stretching from real experiments over simulations to evaluation and verification. The additional software layer for computer supported collaborative work (CSCW) will support collaboration during the experiment, sharing the view on the data, the simulation, and the evaluation.

1.1 Remainder of the Paper

The remainder of the paper is organised as follows. In Section 1.2 we give an overview on related work in the area of Collaborative Remote Laboratories, Section 2 presents the relevant approaches for orchestration of resource through advance reservation and SLAs and the respective VO management technologies. How these technologies can be used to integrate remote laboratories will be described in Section 3. We briefly present the existing MetaScheduling Service (MSS) [18] for the orchestration of the different resources in Section 4. An overview about further developments towards service orchestration in Section 5 concludes the paper.

1.2 Related work

There is a small number of past and ongoing activities to integrate sensors and instruments as Grid resources. Astrophysicists aiming to establish a Grid-based network of radio-telescopes within Europe coined some of the early initiatives, MONET - MOnitoring NEtwork of Telescopes [14] - being one of them. Also, in the domain of life-sciences some efforts are made to make micro-array experiments accessible through the Grid to automate the evaluation and to couple these experiments with docking simulations. Another prominent example is EGEE, providing a Grid infrastructure to distribute and process the vast amount of data resulting from experiments in the large hadron collider at CERN. The UK e-Science Programme features a working group ("Instruments on the Grid") addressing e.g. integration of X-ray Crystallography or sensors for urban pollution into the Grid. During iGrid 2005 the transparent operation of a biology experiment on a test-bed of globally distributed visualization, storage, computational, and network resources was demonstrated [?]. The environment was based on the Distributed Virtual Computer (DVC)

However, these approaches have a number of different drawbacks: they tend to be either too focused with respect to the targeted instruments, or do not address interactive steering of instruments, or are pat of a dedicated demo environment where the environment set-up making the laboratory equipment available is mostly performed manually by the participating researchers.

A more generic approach, driven by an initiative supported by the US National Science Foundation, led to the definition of the Common Instrument Middleware Architecture (CIMA) [5]. CIMA is targeting on developing a Web-service based middleware stack allowing to treat arbitrary instruments as Grid resources (see Figure 1). Current related projects are X-ray Crystallography and the Automated Observatory. CIMA could become a generic interface to instruments and we will evaluate this stack once it becomes available.

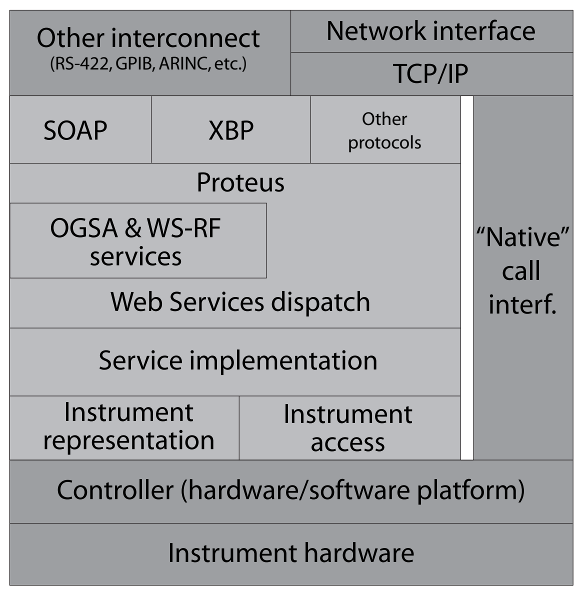

Figure 1. Common Instrument Middleware Architecture (CIMA) [6]

2 Grid technologies for resource orchestration

During the last years a number of technologies has become mature paving the way for orchestration of Grid resources and services. The most important ones being de-

Proceedings of the Second IEEE International
Conference on e-Science and Grid Computing (e-Science'06)
0-7695-2734-5/06 $20.00 © 2006 IEEE

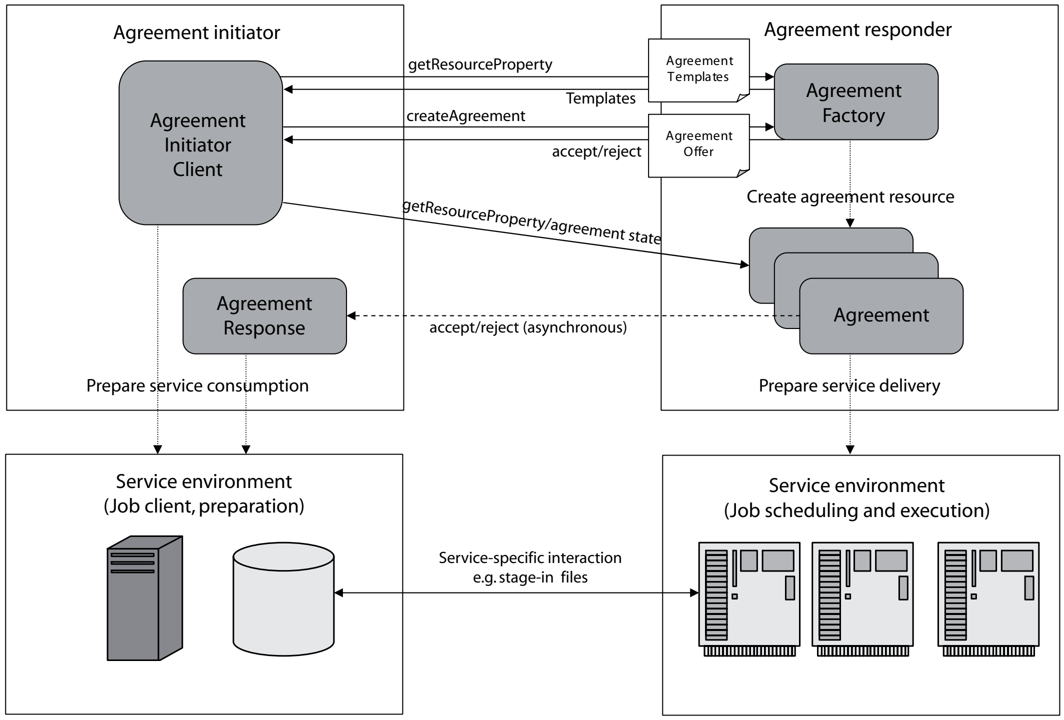

Figure 2. Concepts and interfaces of WS-Agreement

scribed in the following sections: negotiation of advance reservation (Section 2.1), management of virtual organisations (Section 2.2) and service level agreements (Section 2.3).

2.1 Negotiation of Advance Reservation

The need for multiple resources at the same time or at different points in time are two common patterns in Grid use-cases. While the former plays a major role if an application needs more than a single resource, e.g. a multi-physics simulation requiring a compute cluster for each of the simulation codes, the latter is of vital importance for workflows needing multiple resources at different points in time during their execution. Submitting a such a complex job without reservation usually wastes resources as the component started first either has to wait until the other resources become available as well. With advance reservation, however, all resources are available, when they are needed for the execution of a single application or during workflow execution.

Sharing an instrument among different researchers from different organisations is another use-case where a coordination in time is needed to allow all participants joining a collaborative session. Given that laboratories with expensive instruments are just as heavily used as high performance computational infrastructure basically the same possibilities exist: either wait until the instrument becomes available and try to coordinate the collaborators ad-hoc or submit a reservation request for the laboratory and schedule the CSCW tool for the time of the reservation confirmed.

The negotiation process for the common time-slots across all resources needed is briefly described in Section 4, the complete sequence is depicted in Figure 4.

2.2 Virtual Organisations

Virtual Organisation (VO) technology is commonly used today in Grid environments to enable collaboration and coordinated resource sharing in dynamic, multi-institutional virtual organisations. VOs usually are not persistent, but dynamically created for a specific joint (research) task and existing temporarily until the task has been performed. VOs can thus vary in their purpose, size, structure, etc. The primary aspect of VOs in the context of CRLs is management of roles and role-based authorisation for resource access and usage, i.e. who controls the experiment and who is allowed to participate. There are two major - currently incompatible - approaches for authorisation: VOMS [3] (based on X.509 certificates and central management of user attributes per VO) and Shibboleth [16] (based on authentication and management of attributes at the home-site of a user). VOMS is

Proceedings of the Second IEEE International
Conference on e-Science and Grid Computing (e-Science'06)
0-7695-2734-5/06 $20.00 © 2006 IEEE

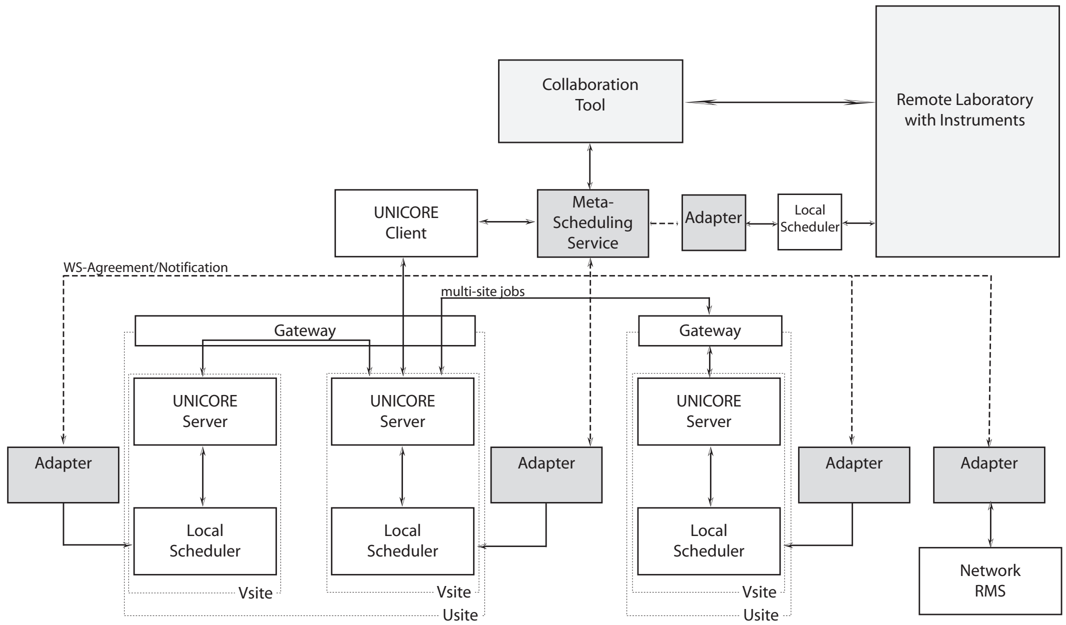

Figure 3. Integration of MSS, Collaboration Tool, Laboratory, UNICORE Compute Resources, and the Network

mainly used together with the gLite middleware in the context of the EGEE project, while the domain of Shibboleth are university campuses, libraries and publishers.

2.3 Service Level Agreements

SLAs are used to transfer the results of the MSS negotiation between all parties into reliable contracts thus bringing together individual service level objectives and service level guarantees of the service provider. This makes for a common understanding of the Quality of Service for both the service consumer and the service provider. In order to create a SLA a service provider usually publishes one or a set of templates describing the services and QoS parameters a provider is able to offer. A consumer may specify the parameters of the SLA according to his needs and send the new offer back to the provider. The provider can now choose whether he can deliver the specified service and accept the offer or not. If he decides to accept the offer a new SLA is created. In Compute-Grid environments SLAs usually specify the resource requirements for compute tasks, e.g. the number and type of compute resources, memory and storage needed, licenses that have to be available, requirements concerning the network connection, and the duration of the usage. For CRLs the SLA will specify additionally e.g. parameters necessary to set up the instrument for the specific operation and authorisation information. However, SLAs are not only used to describe a service that will be delivered, they also define costs that are associated with a service and the penalties in case a service is not delivered with the agreed QoS. Furthermore, SLAs may define the way how the QoS parameters are measured and monitored during resource usage. In our approach we use WS-Agreement [2] as a framework to generate SLAs. WS-Agreement defines a way to create SLA templates and documents, accept and monitor SLAs based on existing standards like XML, SOAP, WS-Addressing and Web Service Resource Framework (WSRF) [19]. The concepts and interfaces of WS-Agreement are presented in Figure 2.

3 Integrating Remote Laboratories

The components of the service environment described in this paper so far do not substitute or supersede an tool for CSCW. On the contrary, we rely on the fact, that there are already a number of useful CSCW tools. Thus the MSS as described in Section 4 is rather a sub-system within a collaborative environment, responsible for

- negotiation and reservation of e.g. instruments, compute clusters, storage, network connections and people,
- establishing the SLAs based on the negotiation, including additional requirements coming from the collaborative environment,

Proceedings of the Second IEEE International
Conference on e-Science and Grid Computing (e-Science'06)
0-7695-2734-5/06 $20.00 © 2006 IEEE

- providing roles and authorisation information for the initial setup of the collaborative environment,
- orchestration launching of additional tools, applications, and services.

The integration of the MSS as depicted in Figure 3 in a collaborative environment can be realised in three flavours: (i) using the command-line interface, (ii) through its simple SOAP [17] interface comprising five function calls:

- `couldRunAt()`,
- `submit()`,
- `state()`,
- `cancel()`,
- `bind()`

or (iii) through the WS-Agreement protocol. In the first and second case the collaboration tool totally ignores the SLAs using the MSS as a reservation front-end of the laboratory while the MSS acting as the service consumer is the agreement initiator. To set up the laboratory environment and to provide the initial authorisations the collaboration tool needs to further negotiate directly with the laboratory. In the third case the collaboration tool is the acting as as the service consumer and is the agreement initiator, while the MSS performs the negotiations as defined in the WS-Agreement template provided by the tool. Here set up of the laboratory environment and initial authorisation can be fixed beforehand in a single negotiation phase.

Once either the command-line interface, the SOAP calls, or the WS-Agreement protocol are integrated in the CW tool for, the familiar, but enhanced collaboration environment can be used to manage the access to and the reservation of the laboratory. Depending on the implementation only allowing reservation through MSS or set up the collaboration while negotiating the SLAs as described above at the same time.

4 The MetaScheduling Service MSS

The MSS is a Web-service developed in the German VIOLA project [1] and is the Grid-level Scheduler in the VIOLA optical testbed since end of 2005. The MSS is responsible for negotiation of the allocation of resources with their local scheduling systems. In case of laboratories without such a local scheduling system a simple one allowing advance reservation and publishing of the local schedule (or at least the next time-slot the laboratory may be used) will be provided. To interact with different types of local scheduling systems the adapter pattern approach is used. The role of these adapters is to provide a single interface to the MetaScheduling Service by encapsulating the specific interfaces of the different local scheduling systems. As mentioned in Section 5 depending on the local scheduling systems of the laboratories new adapters need to be in place to interface with them.

For the negotiation of a reservation the user specifies the duration of the experiment and additionally reservation characteristics, like parameters for setting up the instrument, roles of the participants, or the bandwidth of the connections between laboratory and storage system. In the UNICORE [9] middleware environment for computational Grids the UNICORE client is used to describe the request of the user, whereas in the environment of CRLs the respective collaboration tool will be used. Depending on the implementation, the collaboration tool then sends either single SOAP requests for the reservation or - using the WS-Agreement protocol - the complete description of the reservation to the MetaScheduling Service. Based on the information received the MetaScheduling Service starts the resource negotiation process by querying the local scheduling systems of the resources for their availability and then negotiates the reservations across all local scheduling systems involved. The complete sequence of the negotiation protocol is described in Figure 4.

The result of the negotiation is either returned to the CSCW tool as return value of the SOAP calls (while the accepted agreement is stored by the initiator, the MSS) or as accepted agreement defining the SLA for using the laboratory. In the latter case, the CSCW tool stores the WS-Agreement template(s) for monitoring and possible control of later accounting.

As already depicted in Figure 3 with the proposed architecture it is immediately possible to co-allocate compute resources together with the laboratory for accompanying online simulations or for analysis of the data produced by the experiment. The CSCW tool submits the description of the necessary resources to the MSS using either of the three interfaces described in Section 3. The MSS separates the reservation requests for the laboratory and the computing resources, performs the necessary negotiations and send the result to the UNICORE client for further processing as usual.

The MSS is independent from the Grid middleware used and a proof of concept for the integration into the Globus Toolkit 4 environment [11] has recently been completed. We are now working on the implementation of the full integration into GT4 in order to support two of the major Grid middleware platforms. An integration in the gLite environment [4] is under study.

Proceedings of the Second IEEE International
Conference on e-Science and Grid Computing (e-Science'06)
0-7695-2734-5/06 $20.00 © 2006 IEEE

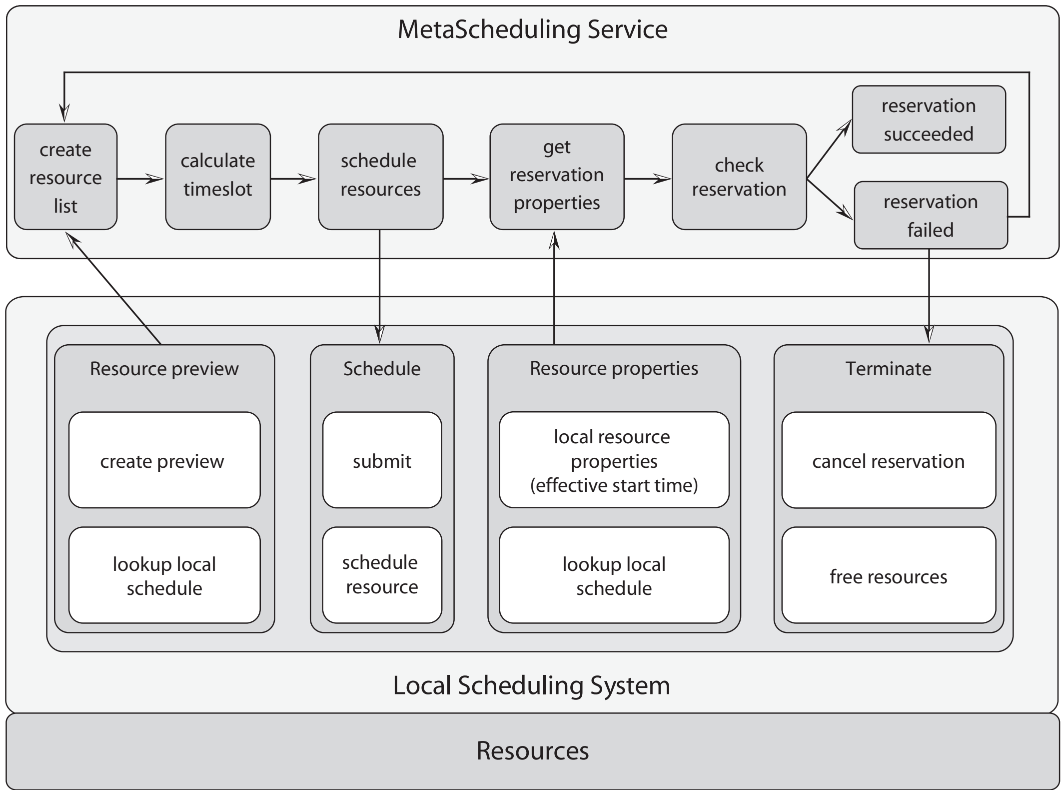

Figure 4. The negotiation process

5 Future Perspectives

The experience made with the MSS in the VIOLA testbed proof that it is an appropriate approach for co-allocation of arbitrary resources. Based on the MSS we plan a prototype implementation of the architecture for collaborative virtual laboratories together with an operator of a laboratory. The results will be analysed and compared to other existing approaches for collaborative virtual laboratories in order to proof the feasibility and usefulness of a service oriented approach for access and sharing of such laboratories . The work done so far and planned is contributing to an ongoing technological shift from providing and using resources to service provision and service consumption. Important milestones being the development of the Open Grid Service Architecture (OGSA) [12] and the WSRF standards in the Grid domain and the development of Service Oriented Architectures (SOA) in the service provisioning domain, e.g. on a European level through NESSI [15]. Wrapping access to and use of CRLs into services is a pre-requisite for full integration of laboratories into the emerging global service infrastructure. Similar to making advanced network features available to the end-user and his application, e.g. Bandwidth on Demand (BoD), CRLs will be made available to end-users and their applications, e.g. Measurements on Demand (MoD).

The proposed CIMA architecture [8] and the corresponding Web-service based middleware stack will be evaluated and the stack will be used as generic interface to instruments once it becomes available. As it is specified now it is expected to smoothly integrate in the current multi-layer scheduling architecture of the MSS.

In order to bring CRLs, MSS and people together two new adapters for the MSS need to be designed and implemented: one towards the laboratories or instruments (or CIMA), the other one to negotiate with the participating researchers to find a common time slot for performing an experiment. Finally, the currently used JSDL scheme for the resource description has to be extended to allow description of laboratories as well. The latter could probably be done on the base of schema developed for CIMA.

6 Acknowledgements

Some of the work reported in this paper is funded by the German Federal Ministry of Education and Research through the VIOLA project under grant #01AK605L. This paper also includes work carried out jointly within the

CoreGRID Network of Excellence funded by the European Commission's IST programme under grant #004265.

References

[1] VIOLA – Vertically Integrated Optical Testbed for Large Application in DFN, 2006. 29 March 2006 <http://www.viola-testbed.de/>.

[2] A. Andrieux and K. Czajkowski and A. Dan and K. Keahey and H. Ludwig and T. Nakata and J. Pruyne and J. Rofrano and S. Tuecke, and M. Xu. WS-Agreement - Web Services Agreement Specification, 2006. 9 July 2006 <https://forge.gridforum.org/projects/graap-wg/document/WS-AgreementSpecificationDraft.doc/en/31>.

[3] R. Alfieri, R. Cecchini, V. Ciaschini, F. Spataro, L. dellAgnello, and A. Frohner ; K. Lörentey. From Gridmap-File to VOMS: Managing Authorization in a Grid Environment, 2004.

[4] R. Berlich, M. Kunze, and K. Schwarz. Grid Computing in Europe: from Research to Deployment. In *Proceedings of the 2005 Australasian Workshop on Grid Computing and e-Research (CRPIT '44)*, Newcastle, New South Wales, Australia, 2005.

[5] The Instrument Middleware Project portal. Web site, 6 April 2006 <http://www.instrumentmiddleware.org/metadot/-index.pl>.

[6] Common Instrument Middleware Architecture. Web site, 20 September 2006 <http://www.instrumentmiddleware.org/metadot/-index.pl?id=2119&isa=Category&op=show>.

[7] Distributed European Infrastructure for Supercomputing Applications (DEISA). Web site, 11 July 2006 <http://www.deisa.org//>.

[8] T. Devadithya, K. Chiu, K. Huffman, and D.F. McMullen. The Common Instrument Middleware Architecture: Overview of Goals and Implementation. In *Proceedings of the First IEEE International Conference on e-Science and Grid Computing (e-Science 2005)*, Melbourne, Australia, December 5–8, 2005.

[9] D. Erwin (ed). UNICORE Plus Final Report. Technical report, Research Center Jülich, Germany, 2003. ISBN 3-00-011592-7.

[10] Enabling Grids for E-sciencE (EGEE). Web site, 21 June 2006 <http://www.eu-egee.org/>.

[11] I. Foster and L. Childers. Introduction to GT4. Tutorial APAC Conference and Exhibition on Advanced Computing, Grid Applications and eResearch (APAC05)., September 2005.

[12] S. Tuecke I. Foster, C. Kesselman. The Open Grid Services Architecture. In Carl Kesselman Ian Foster, editor, *The Grid: Blueprint for a new Computing Infrastructure*, pages 215–242. Elsevier, Morgan Kaufmann, 2004.

[13] H. Ludwig, T. Nakata, O. WŁldrich, Ph. Wieder, and W. Ziegler. Reliable Orchestration of Resources using WS-Agreement. In *2006 International Conference on High Performance Computing and Communications (HPCC-06)*, Lecture Notes in Computer Science, pages 782–791, München, Germany, September 13–15, 2006. Springer.

[14] Astronomie and Internet telescopes. Web site, 16 May 2006 <http://monet.uni-goettingen.de/cgi-bin/WebObjects/MonetPortal>.

[15] Networked European Software and Services Initiative (NESSI). Web site, 17 July 2006 <http://www.nessi-europe.com/index.htm>.

[16] Shibboleth Technical Introduction.. Web site, 2006 <http://shibboleth.internet2.edu/shib-tech-intro.html/>.

[17] Simple Object Access Protocol Specification, SOAP Specification version 1.2. Web site, 2005 <http://www.w3.org/TR/soap12/>.

[18] O. Wäldrich, Ph.Wieder, and W. Ziegler. A Meta-scheduling Service for Co-allocating Arbitrary Types of Resources. In *Sixth International Conference on Parallel Processing and Applied Mathematics (PPAM 2005)*, volume 3911 of *Lecture Notes in Computer Science*, pages 782–791, Poznan, Poland, September 11–14, 2006. Springer.

[19] Web Services Resource Framework (WSRF) Primer v1.2. Web site, 23 May 2006 <http://docs.oasis-open.org/wsrf/wsrf-primer-1.2-primer-cd-02.pdf/>.