GES 2007 German e-Science

Available online at http://www.ges2007.de

Medical Image Processing in MediGRID

S. Kottha[1], K. Peter[2], T. Steinke[2], J. Bart[3], J. Falkner[3], A. Weisbecker[3], F. Viezens[4], Y. Mohammed[4], U. Sax[4], A. Hoheisel[5], T. Ernst[5], D. Sommerfeld[6], D. Krefting[7] and M. Vossberg[7]

[1] Center for Information Services and High Performance Computing (ZIH), TU Dresden, Noethnitzer Strasse 46, D-01187 Dresden, Germany
[2] Zuse Institute Berlin, Takustrasse 7, D-14195 Berlin-Dahlem, Germany
[3] Fraunhofer IAO, Nobelstr. 12, D-70569 Stuttgart, Germany
[4] Department of Medical Informatics, Georg-August-University Göttingen, Robert-Koch-Strasse 40, D-37075 Göttingen, Germany
[5] Fraunhofer FIRST, Kekuléstr. 7, D-12489 Berlin, Germany
[6] Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen, Am Fassberg, D-37077 Göttingen, Germany
[7] Institut für Medizinische Informatik, Charité - Universitätsmedizin Berlin, Hindenburgdamm 30, D-12200 Berlin, Germany

Abstract

In MediGRID a diverse spectrum of application scenarios from areas of bioinformatics, medical image processing, numerical simulations and clinical trials will be integrated into a Grid environment. In this paper we present the MediGRID infrastructure especially as required by medical image processing. Motivated by this selected application scenario the major MediGRID components i) enhanced security requirements ii) data management, iii) portal technology iv) workflow management and v) information service are discussed.

1 Introduction

Scenarios in medical image processing like e.g. blood vessel simulation, ultrasonic image processing, demand high computing power and storage capacity as well as secure treatment of data. Increasing usage of high resolution images and multidimensional data, like volume sequences or multi-modality data, amplify hardware requirements. Today the amount of data is roughly estimated about 5-7 terabytes per year in a 1000 bed hospital and will increase to about 5-7 petabytes per year in future. When results are required within a certain time compromises between accuracy and computing time are unavoidable on limited resources. Furthermore, new algorithms developed by research groups are often hardly available or adaptable for related research problems.

The aim of the MediGRID project, which is part of the German e-Science initiative D-Grid, is to develop the necessary technical and sociological infrastructure to solve challenging problems in medical and life sciences by enhancing the productivity and by enabling location-independent, interdisciplinary collaboration using Grid technologies.

Most of the users in the biomedical community are not IT specialists who have experiences in using high-end computing and storage resources as it is common in “traditional” high-performance computing domains. Hiding the complexity of the Grid infrastructure is therefore a basic requirement for acceptance. Thus, one objective of our work is to realize user friendly access to MediGRID applications from different domains for a large number of globally distributed users and to efficiently use the distributed and shared Grid resources of the community. Additionally to an easy to use interface MediGRID has to focus on security for critical data containing patient information. For this we look closer to the actually provided security functionalities of each middleware.

In the current phase of MediGRID, we establish and enhance a Grid middleware integration platform where a broad spectrum of prototype applications are integrated covering medical image processing, bioinformatics, numerical fluid simulation and clinical research. One advantage of running these applications in a Grid environment is a substantial reduction of overall processing times. Another benefit is easy-to-use access to applications together with a transparent access to various distributed information sources of academic and in the future clinical providers, too.

MediGRID aggregates resources of different resource providers distributed over germany. The grid middleware components as visible in figure 1 are used to provide access to all resources for using compute power on one side and storage capacity on the other side. In this paper each middleware component will be explained. At first we go into details of the medical image processing application which gives the motivation and defines requirements that MediGRID has to accomplish. Because of the medical applications operating with critical patient data MediGRID has a main focus on security. In section 3 the enhanced security is presented. In the following sections we explain the major MediGRID middleware components data management, portal technology, support for complex workflows and information service also highlighting their security capabilities.

2 Application Scenario: Medical Image Processing

As medical image computing benefits from many features of modern Grids, the image processing module is implementing exemplary applications in the MediGRID testbed. Beyond the account for the chosen research projects, the prototyping provides soft- and middleware solutions for a wide range of similar problems and lowers the threshold for implementation of further algorithms and workflows. Three application scenarios are chosen as representatives of typical medical image processing workflows from current research projects with clinical relevance. They encompass major medical image processing components and benefit strongly from implementation into a Grid infrastructure. Besides the specific algorithms and processing steps, all scenarios require Grid-wide image storage, transfer and metadata management. The three scenarios are: statistical

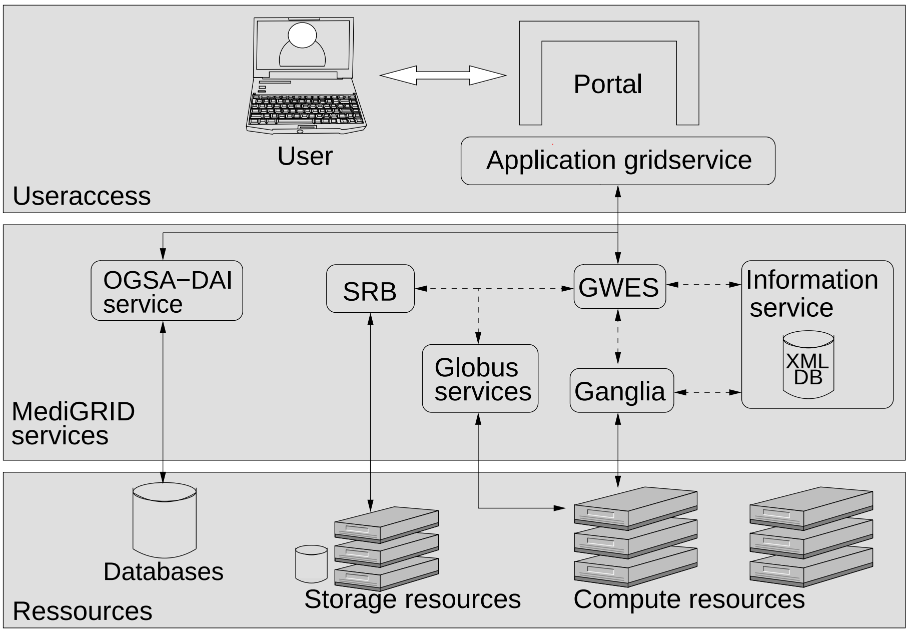


Figure 1: MediGRID middleware components

analysis of functional MRI, virtual vascular surgery and computer-aided diagnosis of prostate cancer. We will describe the latter in more detail and demonstrate the middleware solutions exemplary on this scenario.

Prostate cancer is the most common cancer in men. Current goldstandard for prostate cancer diagnosis is ultrasound guided prostate biopsy. Monitoring the prostate by transrectal ultrasound (TRUS), tissue probes are taken from different parts of the prostate. The present application determines and visualizes the position of the tissue probes within the prostate volume. The localization is done by automated segmentation of the biopsy needle in the guiding 2D ultrasound images. Each biopsy is documented by a sequence of 2D TRUS images of 10 seconds length, saved as a series of DICOM images within a PACS. The task is, besides some preprocessing steps, to determine the frame, when the tissue probe is taken. This is indicated by the moment, where the needle is extended to maximum length. Then the position of the needle within the 2D image is extracted. The location of the needle within the prostate volume is realized by subsequent registration of the 2D images into a previously taken 3D ultrasound volume. The complete image processing chain further encompasses data conversion, prostate segmentation, classification, insertion of the data into an image retrieval system and visualization of the result. The used algorithms are provided from different research institutes in order to combine the knowledge and expertises. They use *Matlab*,*C/C++*,*ITK/VTK/MITK* and *Java*. A SQL-based image processing database tracks the processing steps documenting all metadata beyond the basic image features, input parameters of the chosen algorithms and intermediate results. On one hand, we want to benefit from the computing power provided by the Grid infrastructure as most of the image processing algorithms, especially those processing the 3D volumes are CPU and memory demanding. They can easily be parallelized on a coarse-grained level. On the other hand, we want to combine different algorithms, running on different sites, to a complex workflow, linked together with metadata management. Such a scenario is strongly supported by a modern, service-oriented Grid architecture.

3 Enhanced Security Concept in MediGRID

The use of medical applications with person related data in a Grid environment is constrained to certain restrictions [8, 16]. The principles of confidentiality and privacy have to be respected at all times of a Grid workflow. Whereas medical applications within hospitals still take place under the umbrella of the doctor–patient confidentiality, research computing requires some more technical effort [12, 13]. Authentication via certificates and role based authorization should be standard for Grid security, while Grid map files are still in use. The patient – as owner of his data – has the right to be informed why, where and how long his data is processed and stored. Therefore medical Grid applications must be equipped with a comprehensible audit track in order to fulfill this requirement (*a-posteriori*). Furthermore we have to guarantee the patient, that his data will only be stored and processed in a trustworthy environment (*Tracking*, *a-priori*). This is a challenge in Grid computing, as every Grid node has to be assessed concerning the trustworthiness using trust metrics. The data itself is mostly represented as database content or as files. Current Grid security systems allow to control access on file level or on database entry level. Concerning structured medical documents this is not sufficient, as we need fine granular access control in order to grant access to certain parts of a document only. An enhanced security pilot environment was set up within the MediGRID project. By means of certificates users are authenticated in the MediGRID Portal. Using MyProxy lifetime-restricted credentials are generated for the first medical Grid applications. All actions in the portal are subjected to a strict usage policy, which reflects the paramount legal basis for medical Grid applications.

4 Data Management Systems in MediGRID

Medical data management in a distributed environment like a Grid is still a challenging research topic and it is well analyzed and worked upon in various studies [4, 9, 10, 11, 15]. In MediGRID we have to design a data management solution that fulfills the requirements of present and future MediGRID users.

Today, we run a testbed with two different Grid data management tools SRB and OGSA-DAI complementing one another. SRB and OGSA-DAI provide transparent access to different types of storage systems, i.e. file systems and databases. We decided to run two different independent middleware components because none of them can fulfill all data access requirements.

In the following paragraphs, some properties of OGSA-DAI and SRB are discussed to show how these middleware components meet the requirements of image processing applications.

Open Grid Service Architecture - Data Access and Integration (OGSA-DAI)'s [2] aim is to provide a unique interface to access and integrate data from separate sources. It is an extensive and extendable framework for accessing data in applications and supports relational databases, XML databases, and files. It hides the complexity of heterogeneous databases storage and location from the user. The image processing applications of MediGRID will access their metadata which is in the image processing database (Postgres) using OGSA-DAI.

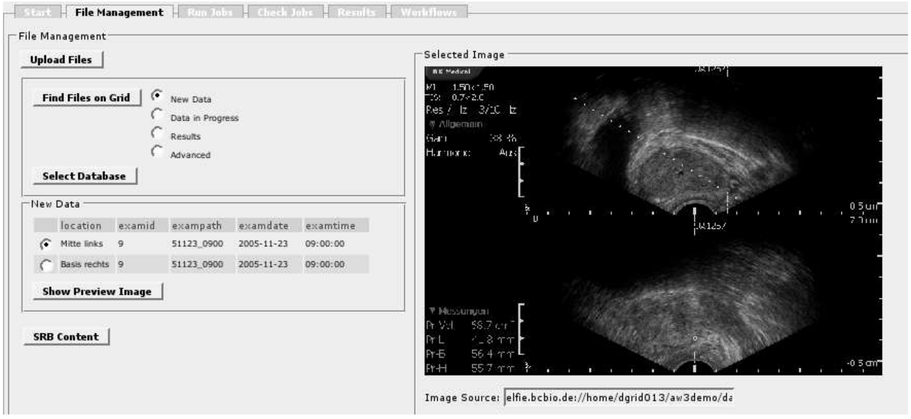
Usually, the database and file servers are kept behind firewalls. To make them accessible to an application, either the servers need to be placed outside the firewall or the ports need to be opened for all the machines in the Grid where the application might run. This poses a severe security threat and/or too much maintenance. Both options are not acceptable for medical applications. One possibility to circumvent this problem is to use OGSA-DAI to deploy the databases. In this case, medical databases are registered as webservices with globus container using OGSA-DAI. The database server port should be opened only for the machine where OGSA-DAI is installed and all the user queries are directed via OGSA-DAI server. The connection between database and OGSA-DAI server is secured using the secure shell tunneling and OGSA-DAI guarantees the security between OGSA-DAI server and the client with its message and transport level security features. GridFTP[1] or Reliable File Transfer (RFT)[2] are used to move files around in Grid.

SDSC Storage Resource Broker (SRB) [3, 14] is based on client-server architecture and provides a global view of multi-organizational heterogeneous storage resources by building a logical file system. For secure user-identification SRB is configured to use GSI authorization with X.509 certificates. Data could be shared or restricted within a community by defining groups of SRB users and granting access rights to relevant data objects. Using ticket mechanism available in SRB, certain data objects could be shared by different users during a defined time period. Image processing users can easily store image files in SRB space and the accompanying algorithm parameters in Metadata Catalog (MCAT) using any of the SRB clients. Besides the metadata created by SRB about the stored files and their replicas or versions, users can also define their own metadata in the form of attribute-value-unit triples and search their data objects using the user defined metadata. Furthermore, SRB provides tools to encrypt data objects at the client side before transferring them to SRB and storing them in SRB space. These tools can also be used by the users with proper key to retrieve the encrypted data objects from the SRB in a very secure

[1] http://www.globus.org/toolkit/docs/4.0/data/gridftp/

[2] http://www.globus.org/toolkit/docs/4.0/data/rft/

mode. The encryption keys are stored in MCAT server.



	location	examid	exampath	examdate	examtime
●	Mitte links	9	51123_0900	2005-11-23	09:00:00
○	Basis rechts	9	51123_0900	2005-11-23	09:00:00

Image Source: elfie.bcbio.de://home/dgrid013/aw3demo/da

Figure 2: Datamanagement within the applicationspecific portlet

Figure 4 is a snapshot of the datamanagement interface in the application portlet. The function ”find files on grid” calls OGSA-DAI for search on the metadata. The results of the query are displayed below. Image data can then be selected, displayed and its location is saved in the portal’s buffer for further processing. The ”SRB content” function connects to a SRB server and displayes all data available for the current user. The self-written webservice can be used to get or retrieve data from the SRB storage.

5 The MediGRID Portal

Grid computing allows reducing the complexity of using different services and resources, but nevertheless the barrier to bring the users to the provided applications in the Grid is still high, especially as they are not all IT specialists. One way to help the user is a Grid portal, where he can get access to and information about the applications and resources he wants to use.

The resources in MediGRID are accessible world-wide through the MediGRID portal. In the portal there are portlets prepared for each kind of application, which allow the users to access the applications without any previous knowledge about the architecture of the resources or the infrastructure of the underlying Grid.

In order to meet the strong security requirements of the biomedical community, the users log on to the portal with PKI X.509 certificates, which prove their identity reliably and more securely than a combination of username and password. As the use of the portal shall be limited to dedicated users in the initial project phase, access is provided only to those users put on a whitelist by a steering group of the MediGRID project. The users can create their accounts without any interaction of an administrator, as long as their name is on this white list. On the list the users are defined by the Distinguished Names (DN) of their certificates. For each user entry the portlets to which he shall have access to are defined. During the registration process, the user receives an email containing an activation link, which confirms the user's acceptance of the MediGRID security policies and verifies the user's email address.

With his account the user gets his own user space on the portal, where he can save the input data for the applications or their results. This allows the user to access his data from everywhere in the world via internet. To handle the large amounts of data of the users, the data is not kept locally on the portal server, but is managed by the Storage Resource Broker.

There has been a special focus on analyzing the needs of the users, in order to ensure that their requirements will be satisfied. Workshops and questionnaires have been used, not only to elicit the needs and wishes of the users, but also to meet the demands of the resource providers. The portal was designed and tested alongside the methods and processes of User Centered Design (UCD) and the Portal Analysis and Design Method (PADEM®) of the Fraunhofer IAO. UCD always keeps the requirements of the users in the centre of attention and is based on the design of interaction between users and the system referred to as interaction design. Its basic approach is to model a system from the user's perspective and to focus on the usability of the software system with which the user interacts.

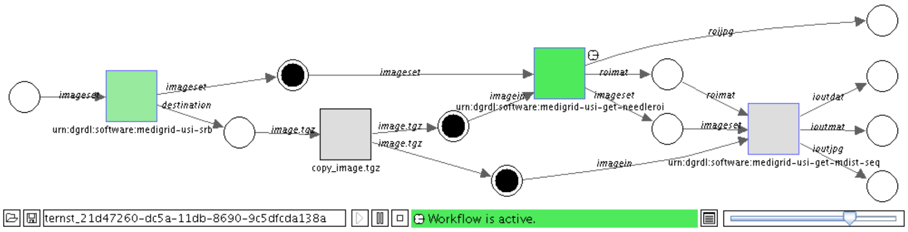
The portlets have been developed for the GridSphere Portal Framework, but are on principle portable to all portal frameworks using the JSR168 Standard. The communication between portal, applications and resources is based on a service-oriented architecture, elegantly solving the firewall problems. The MediGRID applications are therefore wrapped in web services, which are easy-to-use via the internet.

If the applications require access to information saved in databases distributed over the different MediGRID resources, these are provided via the Grid standard OGSA-DAI. Furthermore a portlet provides an easy-to-use graphical interface for designing application workflows based on resource information available in the Grid.

6 Workflow Management and Resource Virtualization

Many applications in MediGRID consist of multiple steps, some of which come with substantial computational resource demands but in many cases also lend themselves well to data-based partitioning, massive distribution and parallel execution over the hardware resources available in the Grid.

In MediGRID, the execution of such complex applications is supported by a flexible workflow orchestration infrastructure that allows to invoke both legacy software components (using WS-GRAM) and existing web services as atomic actions and thus contributes to the confluence of the areas of Grid computing and service-oriented architectures[7]. The control and data flow between the application components is modeled as a graph structure based on the Petri Net formalism[6]. Based on this workflow description (expressed in the XML-based Grid Workflow Description Language[1]) and resource descriptions as well as up-to-date resource monitoring data (both expressed in the D-GRDL[17] Resource Description Language and made available in a metadata repository) the Grid Workflow Execution Service (GWES)[5] automatically selects suitable resources for execution of all the *atomic jobs* of the application. GWES provides file staging capabilities i.e. it also can organize all necessary data transfers so the necessary data is available at each hardware resource on which an atomic job is scheduled and about to execute.



ternst_21d47260-dc5a-11db-8690-9c5dfcda138a

Workflow is active.

Figure 3: Monitoring an ultrasonic image processing application workflow

This workflow orchestration infrastructure enables full resource virtualization: the user no longer needs to care about on which hardware resource his jobs are executed but only interacts with an application-specific portlet which launches the application workflow. By offering a user interface fine-tuned to the task at hand and terminology of the application domain the user is familiar with, this portlet allows the user to fully concentrate on his work. For example in the ultra sonic image processing workflow the user just simply needs to click START WORKFLOW button to start the application. There is no need for the user to leave the application-specific portlet at all: as soon as the results are available he can proceed by e.g. invoking the portlet's data visualization features.

At the same time the workflow infrastructure also optimizes resource usage: for applications that are well-suited for data-based partitioning it is easy to define workflow graphs with an adequate number of branches which will be executed in parallel and on different hardware resources by the GWES as long as suitable resources are available (on which the current existing load is not prohibitive). And the Petri-net based workflow modeling approach enables yet another level of parallelism: data (such as files and XML fragments passed between web services calls) is modeled as tokens which are stored in *place* nodes (round nodes in the figure) and are produced or consumed by *transition* nodes (square nodes) which represent the atomic application components. As the formalism used is powerful enough to allow the presence of multiple tokens on one and the same place node, an arbitrary number of input data can simultaneously be fed to a single workflow instance, and the GWES will orchestrate their processing just as described above, i.e. not just sequentially but exploiting the resulting potential for distributed and parallel execution on the Grid. The GWES also offers a management portlet which amongst others contains a workflow monitoring Java applet called GWUI (Grid Workflow User Interface). This tool allows easy graphical inspection of running workflows and their current state (see figure).

7 Information Service and Metadata Management

The task of the MediGRID information service is to gather all resource information necessary for the operation of the Grid infrastructure. The core of the information service is the resource metadata repository which is realized by an XML database. The resource information in the database is collected by several information providers.

Software resources are program components and web services which can be invoked as atomic jobs in application workflows. Their locations, properties and dependencies on other resources are described using the D-Grid Resource Description Language (D-GRDL). The software resources metadata can be entered into the database with a portlet and must then be confirmed by an administrator.

Information describing available hardware resources is provided by the Grid Resource Database (GRDB) daemon. This metadata consists of static information about the available compute elements in the grid such as architecture, number of nodes and type of batch system as well as monitoring data informing about their current utilization. The GRDB daemon collects this data by querying the Globus Toolkit MDS4 which itself relies partly on the Ganglia cluster monitoring system. The MDS4 outputs resource information in GLUE schema, which is then converted by the GRDB daemon into D-GRDL format.

This way the metadata contained in the database can be used by the GWES for resource matching and covers all information necessary for distributing coarse-grained applications like the medical image processing on the Grid. Additionally the database is used by the GWES to store active and completed workflows in Grid Workflow Description Language. User-defined metadata for data is not kept inside the XML database, but handled by the Storage Resource Broker. This is sufficient as required data can be retrieved from anywhere in the Grid via SRB and OGSA-DAI.

Acknowledgements

This work is funded by the German Federal Ministry of Education and Research (BMBF) within the *MediGRID* Project, grant numbers 01AK803A–H. The workflow infrastructure discussed in section 6 is being developed based on results of the EU IST project *K-Wf Grid* as well as the BMBF-funded projects *Instant-Grid* and *D-Grid Integration project (DGI)*. The GRDB has been developed within the *Instant-Grid* project.

References

1. M. Alt, A. Hoheisel, H.-W. Pohl, and S. Gorlatch. A grid workflow language using high-level petri nets. In R. Wyrzykowski et al., editor, *PPAM2005*, volume 3911 of *LNCS*, pages 715–722. Springer-Verlag Berlin Heidelberg, 2006.
2. Antonioletti1 and et al. The design and implementation of grid database services in ogsa-dai. *Concurrency and Computation: Practice and Experience*, 17(24):357–376, February 2005.
3. C. Baru, R. Moore, A. Rajasekar, and M. Wan. The sdsc storage resource broker. In *CASCON '98: Proceedings of the 1998 conference of the Centre for Advanced Studies on Collaborative research*, page 5. IBM Press, 1998.
4. C. Germain, V. Breton, P. Clarysse, Y. Gaudeau, T. Glatard, E. Jeannot, Y. Legrè, C. Loomis, I. Magnin, J. Montagnat, J.-. Moureaux, A. Osorio, X. Pennec, and R. Texier. Grid-enabling medical image analysis. *The Journal of Clinical Monitoring and Computing*, 19(4-5):339–349, October 2005.
5. A. Hoheisel. Grid Workflow Execution Service - User Manual, Technical Report, Fraunhofer FIRST / K-Wf Grid Project, 2005.
6. A. Hoheisel and M. Alt. Petri Nets. In I.J.Taylor, E.Deeelman, D.B.Gannon, M.Shields, editor, *Workflows for e-Science*. 2007.
7. A. Hoheisel, M. John, and T. Ernst. Fraunhofer first connects SOAs and grid computing: Two concepts get acquainted. In *SAP INFO online*, August 2006.
8. Y. Mohammed, F. Viezens, U. Sax, and O. Rienhoff. Rechtliche Aspekte bei Grid-Computing in der Medizin. *Health Academy*, 2:235–245, 2006.
9. J. Montagnat, F. Bellet, H. Benoit-Cattin, V. Breton, L. Brunie, H. Duque, Y. Legré, I. Magnin, L. Maigne, S. Miguet, J.-M. Pierson, L. Seitz, and T. Tweed. Medical images simulation, storage, and processing on the european datagrid testbed. *Journal of Grid Computing (JGC)*, 2(4):387–400, December 2004.
10. J. Montagnat, V. Breton, and I. Magnin. Using grid technologies to face medical image analysis challenges. In *Biogrid'03, proceedings of the IEEE CCGrid03 (Biogrid'03)*, pages 588–593, Tokyo, Japan, May 2003.
11. J. Montagnat, T. Glatard, D. Lingrand, and R. Texier. Exploiting production grid infrastructures for medical images analysis. In *First Singaporean-French Biomedical Imaging Workshop (SFBI'06)*, Singapore, October 2006.
12. K. Pommerening and et al. Pseudonymization in medical research - the generic data protection concept of the TMF. *GMS Medizinische Informatik, Biometrie und Epidemiologie*, 3:1, 2005.
13. K. Pommerening and M. Reng. Secondary use of the ehr via pseudonymisation. *Stud Health Technol Inform.*, 103:441–6, 2004.
14. A. Rajasekar, M. Wan, R. Moore, W. Schroeder, G. Kremenek, A. Jagatheesan, C. Cowart, B. Zhu, S.-Y. Chen, and R. Olschanowsky. Storage resource broker - managing distributed data in a grid. *J. Comput. Soc, India*, 33(4):41–53, December 2003.
15. A. L. Rowland, T. Hartkens, M. Burns, J. V. Hajnal, D. Rueckert, and D. L. G. Hill. A grid enabled medical image database. In S. J. Cox, editor, *Proceedings of the UK e-Science All Hands Meeting 2004*, pages 1051–1054. EPSRC, September 2004.
16. U. Sax and et al. Medigrid - medical grid computing. In *EGEE'06 - Capitalising on e-infrastructures*, 2006.
17. A. Wolf. Spezifikation der D-Grid-Ressourcenbeschreibungssprache D-GRDL, DGI FG 2-4 Technical Report, Fraunhofer FIRST, 2007.