

# Grid Enabling your Data Resources with OGSA-DAI

Mario Antonioletti<sup>1</sup>, Malcolm Atkinson<sup>2</sup>, Neil P. Chue Hong<sup>1</sup>, Bartosz Dobrzelecki<sup>1</sup>, Alastair C. Hume<sup>1</sup>, Mike Jackson<sup>1</sup>, Kostas Karasavvas<sup>2</sup>, Amy Krause<sup>1</sup>, Jennifer M. Schopf<sup>2,3</sup>, Tom Sugden<sup>1</sup>, and Elias Theocharopoulos<sup>2</sup>

<sup>1</sup> EPCC, University of Edinburgh, JCMB, The King's Buildings, Mayfield Road, Edinburgh EH9 3JZ, UK.

<sup>2</sup> National e-Science Centre, University of Edinburgh & Glasgow, Edinburgh EH8 9AA, UK.

<sup>3</sup> Distributed System Laboratory, Argonne National Laboratory, Argonne, IL, 60439 USA.

**Abstract.** OGSA-DAI (Open Grid Services Architecture - Data Access and Integration) provides an extensible software framework allowing data resources, such as files, relational and XML databases, to be exposed through Web services acting within collaborative Grid environments or, more modestly, in stand-alone mode. OGSA-DAI may be deployed to WSRF-based platforms, such as the Globus Toolkit 4, as well as non-WSRF based ones, such as the UK OMII Server or standard versions of Tomcat and axis. Regardless of the platform, the core functionality provided remains the same. OGSA-DAI allows data resources to be accessed and integrated into the main infrastructures presently being used to construct Grids. OGSA-DAI provides a number of optimisations that reduce unnecessary data movement by shifting work to the Web service and encapsulating multiple client-Web service interactions into a single one, and allows for functionality to be added or customised based on the application. OGSA-DAI is widely used and is available from [www.ogsadai.org.uk](http://www.ogsadai.org.uk). It is also bundled with the OMII-UK and Globus Toolkit distributions. This paper gives an overview of what OGSA-DAI is, how it works, presents some usage scenarios, and outlines future enhancements.

**Key words:**Data, Databases, Grid, OGSA-DAI

## 1 Introduction

With current advances in technology and the decreasing cost of storage, increasingly large amounts of data are being produced, maintained, kept on-line, and shared within communities. For instance, astronomers are collecting data together, such as surveys of the sky made at different wavelengths and resolutions, and making it collectively available through *Virtual Observatories* [1]; biologist are gathering DNA and genomic data from different species and making this

data available to biologists through data stores, providing a rich source of data to pursue insights into biological systems [2] and in the health sector, digital medical data are being collected and maintained by hospitals allowing experts to collaborate in patient diagnosis and providing case histories that can be used to inform a prognosis for patients suffering from similar maladies [3, 4].

The need to access disparate data sources, often spanning multiple institutions, can lead to new insights and discoveries to be made. By combining different wavelength data for the same patch of sky, astronomers have been able to make new discoveries that would not have otherwise been possible from a single survey [5, 6]. Biologists now have the capability of performing cross-species comparisons to determine new genes and their function [7]. Doctors can improve the diagnosis of breast cancer by comparing current mammographs with old mammographs in combination with the associated patient histories [8]. The advantage being able to share data and resources in a controlled manner within a collaborative environment is clear. The provision of generic middleware to facilitate this process is the ethos that is currently driving the evolution of the Grid and, in the data area, OGSA-DAI (Open Grid Services Architecture - Data Access and Integration) provides software which makes it easy to publish and share data across organisational boundaries, and develop applications which use both public and personal data resources, through a secure, extensible framework based on web service standards.

OGSA-DAI is not the only solution currently available for data in the Grid space. *Storage Resource Broker* (SRB) [9], developed by the San Diego Super-computer Center, provides access to collections of data primarily using attributes or logical names rather than using the data's physical names or locations. SRB is primarily file oriented, although it can also work with various other data object types. OGSA-DAI on the other hand takes a database oriented approach to its access mechanisms. *WebSphere Information Integrator* (WSII), a commercial product from IBM, provides data searching capabilities spanning organisational boundaries, provides a means for federating and replicating data, as well as allowing for data transformations and data event publishing to take place [12]. A more detailed comparison between OGSA-DAI and WSII can be found in [10]. *Mobius* [11], developed at Ohio State University, provides a set of tools and services to facilitate the management and sharing of data and metadata in a Grid environment. To expose XML data in Mobius, the data must be described using an XML Schema, which is then shared via their Global Model Exchange. Data can then be accessed by querying the Schema using, for example, XPath. OGSA-DAI, in contrast, does not require an XML Schema to be created for each piece of data; rather, it directly exposes that information (data and metadata/schema) and relies on the resource's intrinsic querying mechanisms to query its data. These three products all provide mechanisms to share data across organisational boundaries, however they complement the functionality provided by OGSA-DAI.

In the remainder of this paper OGSA-DAI will be examined in more detail. Section 2 gives an overview of the current release of OGSA-DAI explaining the

underlying components and how they operate. Section 3 describes some common patterns of use for OGSA-DAI, and Section 4 describes some of the future work planned for the next release. Finally conclusions are provided in Section 5.

## 2 An Overview of OGSA-DAI

The first thing to note about OGSA-DAI is that it is not targeted directly at the end user, but rather it gives service providers the base functionality which they can use to create their own services and clients to expose data tailored to their own communities. OGSA-DAI has been made extensible by design so that any missing functionality can be developed and grafted to work within the same framework. In addition, different security models may be employed, static metadata can be exposed via configuration files, and dynamic metadata can be created and exposed at the service via the use of call back functions. OGSA-DAI may also be extended to support new types of data resources that are not already supported by the OGSA-DAI distribution. For example, the WebDB project has extended OGSA-DAI to cater for RDF based data [13]. Use of OGSA-DAI allows service providers to develop and deploy their own Grid solutions much more quickly and effectively than might otherwise be the case.

OGSA-DAI is tested and operates well with two current Grid fabric providers, the Globus Toolkit<sup>1</sup> and the OMII-UK<sup>2</sup> and there are plans to port OGSA-DAI to work with UNICORE<sup>3</sup> and gLite<sup>4</sup> under the OMII-Europe project [14]. This ensures that, if any of the above toolkits is to be used, that the OGSA-DAI services will meet user and developer needs in a wide variety of environments.

In OGSA-DAI data resource and service capabilities are exposed through the use of *activities*, the basic unit of work within OGSA-DAI. At the server, an activity is described by a piece of XML Schema specifying the syntax of an XML fragment that is used to activate an associated Java implementation class that performs the desired task at the server. Different XML activity fragments may be composed together in a *perform document* which contains one or more activities linked together through a named set of inputs and outputs describing the data flow between them. For example, an XPath query activity can wrap an XPath expression which then acts on an XML database, the results of this can then be transformed using XSLT activity and finally the transformed results may be delivered to a specified third party using a delivery activity. It is this ability to encapsulate multiple interactions in a single Web service interaction, through the use of perform documents, which otherwise would require multiple distinct client-service interactions, coupled with the fact that activities provide a framework for moving computation close to the data that is seen as one of the advantages of using OGSA-DAI. More complex behaviour may be obtained by composing OGSA-DAI services together and using these to provide more

---

<sup>1</sup> [www.globus.org/toolkit](http://www.globus.org/toolkit)

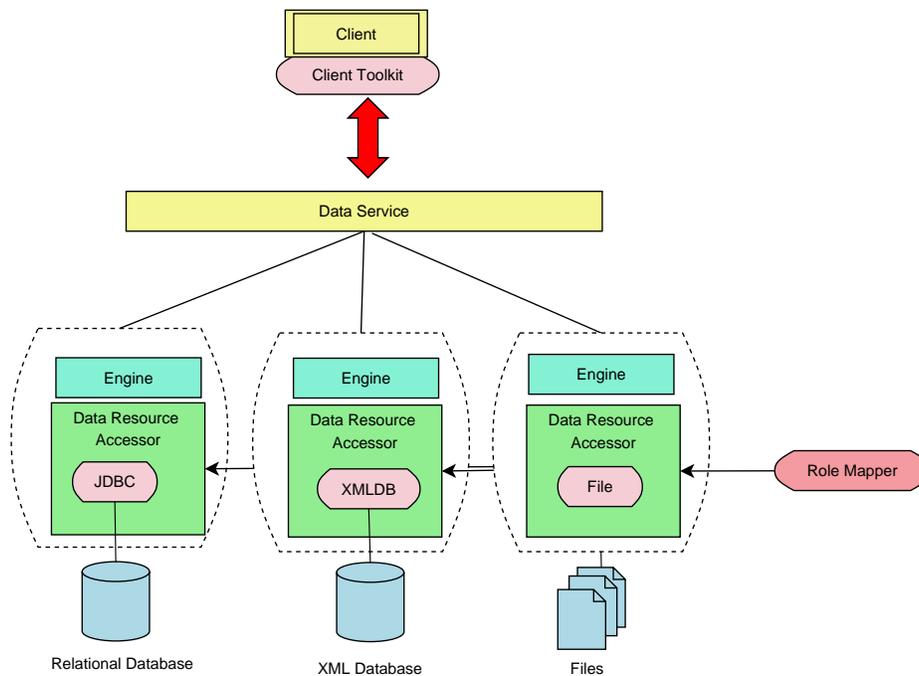
<sup>2</sup> [www.omii.ac.uk](http://www.omii.ac.uk)

<sup>3</sup> [www.unicore.org](http://www.unicore.org)

<sup>4</sup> [www.glite.org](http://www.glite.org)

sophisticated capabilities such as *Distributed Query Processing* as provided by the OGSA-DQP project [16].

The OGSA-DAI *Client Toolkit* (CTk) provides a programmatic interface that facilitates programming interactions with OGSA-DAI services. The CTk has an activity representation for each of the server side activities – these representations are essentially used to produce the XML fragment, within the context of a perform document, to trigger the corresponding server side activity. The CTk also provides a programmatic means for composing the client-side activities together to construct the desired perform document – in this way the user does not need to have to deal with any of the underlying XML. In addition, the CTk also handles the interactions with the service and provides methods to add (or extract) data from the request (or result) messages, respectively. Moreover, the CTk is agnostic as to whether a WSRF or non-WSRF service is being accessed providing an additional abstraction layer hiding the particular flavour of OGSA-DAI service that is being contacted. The overall aim of the CTk is to facilitate the provision of clients to interact with OGSA-DAI services.



**Fig. 1.** A schematic representation of an OGSA-DAI service.

Putting the above into context a schematic representation of an OGSA-DAI service is shown in Figure 1. A client, built using the CTk, sends a perform

document to an OGSA-DAI data service, which in the instance shown has three types of data resource associated with it. In the WSRF version of OGSA-DAI WS-Addressing *end point references* are used to specify the data resource being targeted by the client [17]. For the non-WSRF version the data resource name, specified at deployment time, is appended to the service URL, for example

*http://myhost:8080/MyService*

would become:

*http://myhost:8080/MyService/MyDataResource.*

Once a message is accepted by the service interface, the functionality for both flavours of OGSA-DAI is the same. A perform document and any Grid credentials are passed through the service layer to the *Engine* of the targeted data resource. The Engine coordinates the running of the activities in the perform document. A *Data Resource Accessor* (DRA) wraps the underlying *data resource*: this abstraction facilitates the addition of new types of data resources to OGSA-DAI. The Engine passes any Grid credentials to the DRA and, if these are valid, the DRA returns an open connection to the data resource that can then be used by any activity that interacts with the data resource. The DRA consults a *Role Mapper* that maps Grid credentials, essentially the *distinguished name*, to a database role that can be used to access the database. OGSA-DAI comes with a basic role mapper that attempts to match a database role (represented as simple username and password pairs) within an XML file for a given set of Grid credentials, thus allowing database systems which do not use Grid credentials to be accessed, albeit not in a scalable fashion. This is another extensibility point where service providers would wish to develop their own role mapper and substitute the existing one: two groups have developed different solutions to this.

The OGSA-DAI Engine ensures that all activities run correctly and coordinates the passing of data from one to the other. Failure in one activity signifies failure in the execution of the whole perform document. As yet there is no transactional behaviour, including rollback mechanisms, although this is planned for a future release. Data may be piped in from a third party using a *delivery from* activity or sent to a third party using a *delivery to* activity, both of these can use other transport protocols to pipe data into or out of a service obviating the requirement for SOAP and using, for example, GridFTP, FTP or HTTP to fetch data or send it to a third party. If the processing completes successfully the data or status of the processing is sent back to the client in a response document.

From this brief overview we can see that OGSA-DAI is a sophisticated piece of middleware that provides a uniform access interface to various types of data. It partially virtualises data: intrinsic connection mechanisms to the underlying data resource are no longer a concern but a client still needs to know the underlying type of data model that is being used – for instance SQL queries need to be targeted at a relational data resource and will make no sense when targeted at an XML database. Moreover, query expressions targeted at a particular data

resource are not inspected so any vendor specific language extensions must also be appropriate for the underlying data resource used. A client is able to determine the type of the underlying data resource via metadata available through the service interface that then allows it to direct the appropriate type of queries for that type of data resource. However, OGSA-DAI does provide the basis for providing data model integration, through the use of transformation activities which e.g. translate the results of queries to XML and relational data resources into WebRowSet before aggregating them. This then outlines the basics of the OGSA-DAI framework. The next section briefly outlines a couple of usage scenarios.

### 3 Deployment Scenarios

OGSA-DAI provides a versatile framework which can be used to provide data access capabilities within Grid infrastructures. Many projects already use OGSA-DAI, primarily in research areas such as GIS and bioinformatics. An up to date list can be found on the OGSA-DAI website<sup>5</sup>.

Five basic common usage patterns are illustrated in Figure 2.

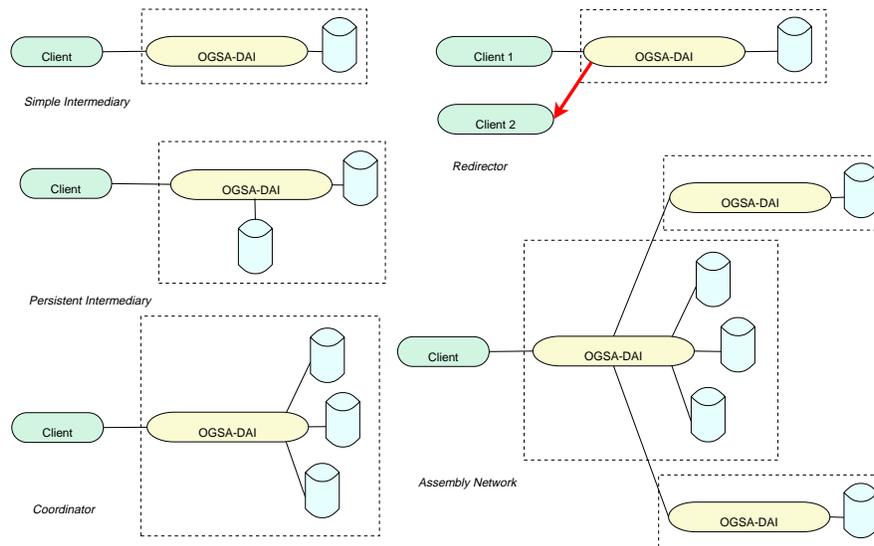


Fig. 2. OGSA-DAI scenarios

The *simple intermediary* is the simplest archetypal usage scenario supported by OGSA-DAI, and is the basis for many of the higher-level scenarios. This

<sup>5</sup> See <http://www.ogsadai.org.uk/about/projects.php>

scenario consists of an OGSA-DAI service interposed between client applications and a data resource providing a consistent interface for different kinds of data and supporting a rich, extensible set of operations that can be performed on that data. Using this base scenario one can envisage many discoverable OGSA-DAI services listed in third party registries and used by clients to retrieve data for their specific ends, all different types of data shared and made available through a common interface. In addition, examination of this basic usage pattern has also led to various optimisations being made for the 2.2 release of OGSA-DAI, see [18] for more details.

The *persistent intermediary* scenario illustrates the use of mechanisms for storing intermediate results which can then be used by subsequent requests. These intermediate results could be stored transparently in memory, a local database, the local file systems or some other suitable means on behalf of the OGSA-DAI service clients. This scenario currently is partially supported by using the OGSA-DAI *dataStore activity* which currently holds results in memory, although this can be extended to hold results in more permanent storage. It can also be implemented using OGSA-DAI by storing data temporarily in a scratch database accessible by the service. This functionality allows a coordinating service to hold temporary data to perform data joins from multiple data resources.

The *redirector* scenario allows data to be sent to a third party, including the originator, as opposed to embedding it in the response. Moreover the third party delivery protocol does not have to be SOAP based – data can be delivered using GridFTP, ftp, or some other delivery means. In this instance SOAP is effectively being used as the control channel while the data channel is done via a more efficient transport protocol. OGSA-DAI supports this scenario by allowing a number of alternative data transport mechanisms that can also be used to transfer data into OGSA-DAI services.

In the *coordinator* scenario, an OGSA-DAI service interfaces to an arbitrary number of data resources and presents them as a composite resource to its clients, producers, and consumers. This means that data can be routed between data resources or combined from those resources within a single request or session without routing data via the client. There is already some support for this type of scenario in OGSA-DAI as multiple data resources can be configured per data service and used with specialised query activities to provide resilient querying of a set of data resources sharing a common schema. This presents the set of data resources as a single virtualised data resource.

In the *network assembly* scenario an OGSA-DAI service uses an arbitrary number of other OGSA-DAI services as well as data resources already curated by the service in order to collect together data. This type of service coordination successively adds facilities that may be used in combination towards achieving a data-oriented workflow. The invoked services in this workflow do not have to be OGSA-DAI services. The multiple services may form a pipeline in order to draw on additional computation facilities or a tree in order to place parts of a total query close to the data sources. As this permits arbitrary fan out and arbitrary recursive composition, many architectures are possible: a simple exam-

ple is shown above. OGSA-DQP provides an instance of the assembly network pattern using OGSA-DAI services as well as some of their own service types.

In general the documentation of scenarios like those described above is beneficial as a means of providing best practice and guidelines for using the features and components of OGSA-DAI. There is insufficient space here to go into more depth but best practice and guidelines are being documented in the OGSA-DAI Web pages<sup>6</sup>. These scenarios, as well as other inputs such as performance studies [19], are being used to motivate the future directions being taken by OGSA-DAI which are briefly outlined in the next section.

## 4 Future Directions

A number of architectural changes are about to be introduced into the next OGSA-DAI release and following releases. Some of the highlights are:

- *Improved scalability* by providing load balancing capabilities to dispatch incoming request to different JVMs, potentially running on different machines, to execute perform documents. Initial policies will be simple, eg. round-robin, but additional more complex policies will be enabled as well.
- *Improved robustness* by allowing requests to run on different JVMs so if that a request has aberrant behaviour it does not bring down the whole container and compromise other jobs.
- *Improved activity model* to make it easier to develop and maintain activities while at the same time providing more powerful mechanisms to dynamically configure activities.
- *Improved sessions handling* will allow activities to store and retrieve data from an existing session, which could span multiple requests.
- *A new resource model* will allow perform documents to contain activities that can access more than one resource exposed by an OGSA-DAI service (currently a perform document can only target a single data resource). This new model will thus allow more powerful user-driven data integration scenarios to be enacted by an OGSA-DAI service between multiple resources. It will also allow other OGSA-DAI components to be treated as WSRF-resources. For example, sessions, requests, and data sinks/sources (input/output streams) can be modelled as resources which then allows these to be endowed with mechanisms for lifetime management and authorisation as available in other resources.
- *New data integration activities* taking advantage of the new resource model a new set of data integration activities are being designed that should facilitate the enactment of data integration scenarios.
- *Distributed Query Processing* capabilities are being introduced through the absorption of the OGSA-DQP project, currently distributed separately from OGSA-DAI, into the OGSA-DAI product itself.

---

<sup>6</sup> See [www.ogsadai.org.uk/documentation/scenarios](http://www.ogsadai.org.uk/documentation/scenarios) for details.

– A *new tuple intermediate data format* for relational data, called an *ODTuple* (for OGSA-DAI Tuple), will provide a common way for connected activities to exchange data. This will minimise the amount of data conversion that is required take place between activities. This format is:

- light-weight,
- able to stream well within and between processes,
- efficient for single types, elements, and tuples,
- able to support base types plus String, File, BLOB, and NULL,
- able to supports warnings, errors and exceptions, and
- easily extensible.

This relational structure can be used to represent the majority of the current data formats used within OGSA-DAI, such as WebRowSet and CSV (Comma Separated Values).

These additions to the next release will make OGSA-DAI a more powerful framework and increase the support for data integration as well as making the scenarios described in the previous section easier to implement and extend.

## 5 Conclusions

This paper has provided motivation for the production of middleware to facilitate the sharing of data within established communities to enable new insights and discoveries to be produced. The provision of middleware that facilitates this process is the underlying motivation for OGSA-DAI. OGSA-DAI is not targeted directly at the end-user but rather it provides a framework that has to be customised for a given user-community by its own developers. Through the use of OGSA-DAI the amount of effort required to produce these targeted data services and applications should be greatly reduced. A snapshot overview of OGSA-DAI has been given and some indicators of the future directions that are being taken to enhance the product and provide additional capabilities for those that rely on OGSA-DAI for their data access and integration base requirements. More information about OGSA-DAI and the software may be downloaded from the project Web site at [www.ogsadai.org.uk](http://www.ogsadai.org.uk).

## Acknowledgements

This work is supported by the UK e-Science Grid Core Programme, through the Open Middleware Infrastructure Institute UK, and by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, Office of Science, U.S. Department of Energy, under Contract W-31-109-ENG-38.

## References

- [1] S.G. Djorgovski. Virtual astronomy, information technology, and the new scientific methodology. Proceedings of the Seventh International Workshop on Computer Architecture for Machine Perception, 2005. CAMP 2005. pp.125-132, 4-6 July 2005.
- [2] R. Edgar, M. Domrachev and A. E. Lash. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Research*, 2002, Vol. 30, No. 1 pp. 207-210.
- [3] J.M. Brady, D.J. Gavaghan, A.C. Simpson, M. Mulet-Parada and R.P. Highnam, eDiaMoND: a Grid-enabled federated database of annotated mammograms. In: F. Berman, G.C. Fox and A.J.G. Hey, Editors, *Grid Computing: Making the Global Infrastructure a Reality*, Wiley Series (2003), pp.923-943.
- [4] K. H. Buetow. Cyberinfrastructure: Empowering a "Third Way" in Biomedical Research. *Science* 6 May 2005: Vol. 308. no. 5723, pp. 821-824.
- [5] Virtual observatory finds black holes in previous data. *News in brief. Nature* 429, 494-495, June 2004.
- [6] Astronomers Detect New Category of Elusive 'Brown Dwarf'. *The New York Times*, Tuesday, June 1 1999.
- [7] A. Wipat, Y. Sun, M. Pocock, P. Lee, P. Watson and K. Flanagan. Developing Grid-based Systems for Microbial Genome Comparisons: The Microbase Project. Proceedings of the UK e-Science All Hands Meeting 2004.
- [8] A. Solomonides, R. McClatchey, M. Odeh, M. Brady, M. Mulet-Parada, D. Schottlander and S.R. Amendolia. MammoGrid and eDiamond: Grids Applications in Mammogram Analysis. Proceedings of the IADIS International Conference: e-Society 2003. Lisbon, Portugal. June 2003. A Palma dos Reis and P Isaias, Editors pp 1032-1033.
- [9] Storage Resource Broker (SRB), [www.sdsc.edu/srb](http://www.sdsc.edu/srb).
- [10] R. O. Sinnott and D. Houghton, Comparison of Data Access and Integration Technologies in the Life Science Domain, Proceedings of the UK e-Science All Hands Meeting 2005, September 2005.
- [11] Mobius, [projectmobius.osu.edu](http://projectmobius.osu.edu).
- [12] Web Sphere Information Integrator (WSII), [www.ibm.com/software/data/integration](http://www.ibm.com/software/data/integration).
- [13] OGSA-DAI-RDF project, [www.dbgrid.org](http://www.dbgrid.org).
- [14] OMII-Europe project, [www.omii-europe.org](http://www.omii-europe.org).
- [15] InteliGrid project, [www.inteligrid.com](http://www.inteligrid.com).
- [16] N. Alpdemir, A. Mukherjee, A. Gounaris, N.W. Paton, P. Watson, and A.A.A. Fernandes. OGSA-DQP: A Grid service for distributed querying on the Grid. *LNCS Volume 2992*, p 858-861, 2004.
- [17] M. Gudgin, M. Hadley, T. Rogers. Web Services Addressing 1.0 - Core (WS-Addressing). W3C Recommendation, 9 May 2006.
- [18] B. Dobrzelecki, M. Antonioletti, J. M. Schopf, A.C. Hume, M. Atkinson, N.P. Chue Hong, M. Jackson, K. Karasavvas, A. Krause, M. Parsons, T. Sugden, and E. Theocharopoulos. Profiling OGSA-DAI Performance for Common Use Patterns. Proceedings of the UK e-Science All Hands Meeting 2006.
- [19] S. Kottha, K. Abhinav, R. Muller-Pfefferkorn, and H. Mix. Accessing Bio-Databases with OGSA-DAI - A Performance Analysis. To appear in *International Workshop on Distributed, High Performance and Grid Computing in Computational Biology (GCCB2006)*.