The gCube Interoperability Framework

Leonardo Candela[1], George Kakaletris[2], Pasquale Pagano[1],
Giorgos Papanikos[2], and Fabio Simeoni[3]

[1] Istituto di Scienza e Tecnologie dell'Informazione "Alessandro Faedo" – CNR, Pisa - Italy
{leonardo.candela | pasquale.pagano}@isti.cnr.it
[2] Computer Science Department, University of Athens – Athens, Greece
{g.kakaletris | g.papanikos}@di.uoa.gr
[3] Department of Computer and Information Sciences, University of Strathclyde, Glasgow, UK
fabio.simeoni@cis.strath.ac.uk

Abstract. Interoperability is one of the most challenging issues in the design and operation of large-scale computing infrastructures for multidisciplinary research. It requires solutions that can "embrace" heterogeneity, i.e. accommodate multiple incarnations of similar resources, as well as "hide" it, i.e. provide consumers with a homogeneous view of diverse resources. This paper overviews the measures put in place in the D4Science infrastructure to address a range of interoperability problems.

1 Introduction

eScience scenarios encompass research investigations that span multiple institutions and disciplines. This calls for innovative environments where scientists can seamlessly access data, software, and processing resources managed by diverse systems in separate administration domains. Dependently on context, these environments are commonly referred to as *Virtual Research Environments* [18,10], *collaboratories* [35,15], *digital libraries* [17,8], *e-Infrastructures* [5] and *cyberinfrastructure* [4]. A variety of systems fall within the scope of these definitions, from ad-hoc portals with minimal access services to content resources held in external repositories (low integration) to general-purpose management systems with advanced services defined over a wide range of resources (high integration). In some cases, motivations and design align with the principles of *grid computing* and its ecology of *virtual organisations* [16]. In all cases, the systems aggregate autonomous components and their ability to interoperate emerges as a core design requirement.

This paper describes the interoperability solutions that have been developed in the context of the D4Science-II EU Project [12], a continuation of the work carried out in the GÉANT, EGEE, DILIGENT and D4Science projects towards the deployment of networking, grid-based and data-centric e-Infrastructures. The distinguishing feature of D4Science-II is the aim to provide *on demand* Virtual Research Environments (VREs) for the creation and dissemination of scientific and technical knowledge. To this end, the project bridges a number of well-established e-Infrastructures from various domains, including high-energy physics, biodiversity, fishery and aquaculture resources management. The result is an infrastructure that exemplifies the concept of *e-Infrastructures ecosystem*.

The remainder of the paper is organised as follows. Section 2 outlines the scope of the project and overviews *gCube*, i.e. the software system that supports the operation of the D4Science-II infrastructure. Section 3 illustrates the project's perspective on interoperability and the solutions that have emerged from it. The goal of these solutions is to seamlessly manage, and consume services and resources which are “external” to the infrastructure. Section 4 presents the current status of the work, its success stories, and its main achievements to date. Finally, Section 5 concludes the paper and reports on future activities.

2 D4Science-II and its Enabling Technology in a Nutshell

D4Science-II [12] is a project co-funded by the European Commission and started in October 2009. Its goal is to build a knowledge ecosystem as a set of interoperable data e-Infrastructures, repositories, and scientific communities (cf. Figure 1). The project builds on the results of other projects and initiatives, including: DILIGENT and D4Science, which have built an e-Infrastructure for *on demand* VREs [9,3]; GÉANT, EGEE, DRIVER, GENESI-DR, INSPIRE, AquaMaps and other, which have built e-Infrastructures for storing and processing collections of documents and data, including statistical, satellite and species distribution data. Figure 1 illustrates the pivotal role of the D4Science e-Infrastructure in the target ecosystem: as a *virtual aggregator* of resources that originate *from* other e-Infrastructures, and as a *provider* of aggregated resources *to* other e-Infrastructures. Resource aggregation and provision occur in the context of complex VREs for multidisciplinary scientific communities.

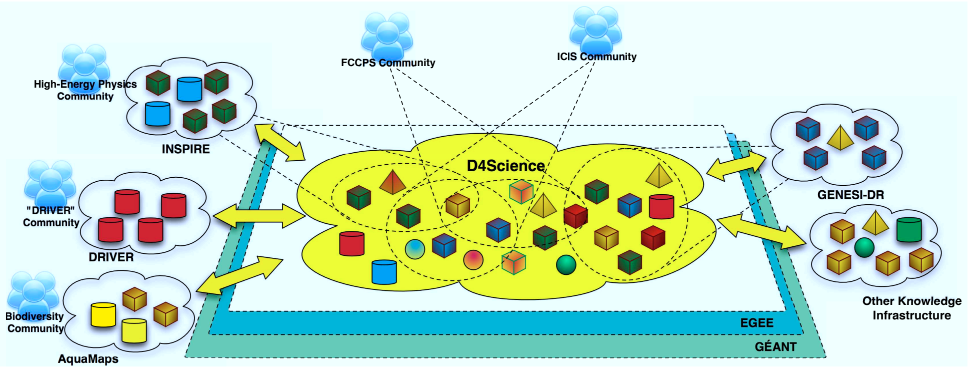

Fig. 1. The D4Science-II knowledge ecosystem

The ecosystem is enabled by an innovative software system, *gCube*[4]. gCube has been designed from the ground up to support the full lifecycle of modern scientific enquiry, with particular emphasis on application-level requirements of information and knowledge management. To this end, it interfaces pan-European Grid middleware for shared access to high-end computational and storage resources [2], but it complements this with a rich array of services that collate, describe, annotate, merge, transform, index, search, and present information for a variety of multidisciplinary and international communities. Services, data collections, and machines are infrastructural resources that communities select, share, and consume in the scope of collaborative VREs [10].

[4] `www.gcube-system.org`

gCube services are grouped in functional subsystems and are arranged in a Service Oriented Architecture. These include subsystems guaranteeing the operation of the infrastructure and its VREs; subsystems supporting the management (storage, organisation, description and annotation) of information represented via a rich and flexible model; subsystems guaranteeing information discovery; subsystems enabling the processing and manipulation of information via dedicated services (e.g. species distribution maps estimation) to produce new artifacts; subsystems providing a seamless access to the services and resources forming the infrastructure for human as well as programmatic consumption.

The architecture of the overall system adopts the following principles:

- the services are WSRF-compliant Web Services;
- the services discover each other dynamically, by mediation of the Information System;
- the services can be orchestrated by the system so as to execute workflows.

An application framework for developing gCube services and their clients has also been developed, the *gCore Framework* [25]. gCore allows gCube developers to abstract over functionalities that are offered at lower layers of the Web Services stack (WSRF, WS Notification, WS Addressing, etc.), and to make use of advanced features for the management of state, scope, events, security, configuration, fault, service lifetime, and publication and discovery.

In D4Science-II, this technology is consolidated and enhanced to address the interoperability requirements that emerge from the role of the D4Science infrastructure within the target ecosystem. The resulting services are described in the rest of the paper.

3 Interoperability Perception and Approaches in gCube

Despite the critical role given to interoperability in modern Information Systems, there is little theoretical guidance on how to address interoperability issues. There is in fact no definition of interoperability which is universally accepted [11]. The IEEE Glossary defines it as "*the ability of two or more systems or components to exchange information and to use the information that has been exchanged*" [19]. Two conditions must thus be met for a producer and a consumer to interoperate: (*i*) they must be able to exchange information; (*ii*) the consumer must be able to make effective use of the exchanged information, i.e. it perform the envisaged tasks by relying only on the exchanged information.

In the context of gCube, we define interoperability from a more general perspective, as *the ability of an arbitrary number of information systems to collaborate into achieving a common cause*. Typically, the common cause is the provision of functionality that is outside the scope of individual systems.

Here, we do not insist on consumer and producer roles and allow for different degrees of "autonomicity", from cases where interoperability is achieved in full automation to cases where human mediation is required. Most existing interoperability solutions fall within these extremes.

Analysing the problem in a top-down manner, the producer-consumer scheme is useful. Complex interoperability issues can be decomposed in fundamental concepts so as to allow for more fine-grained control and evaluation of interoperability solutions, whether these deal with message exchange protocols (manifestations, transports, etc.), message semantics, policies and so on.

As recognized by Paepcke et al. [24], over the years systems designers have developed different approaches and solutions to achieve interoperability. They have followed pragmatic approaches and started to implement solutions that blend into each other by combining various ways of dealing with the issues, including standards and mediators. Too often these solutions remain confined to the systems for which they have been designed. This leads in turn to "from-scratch" development scenarios and much duplication of effort when similar interoperability scenarios recur in different contexts.

In D4Science-II, we have adopted a two-step approach to effectively meet the challenges of developing interoperability solutions, while maintaining the maximum sustainability and visibility of the results. First, we have performed a per-case analysis in all the domains in which multiple infrastructures must interoperate, starting from the functional objective to then identify a set of individual actors that exchange messages. In the second step, we analyzed individual actors from the perspective of specifications, so as to conform to widely adopted ones, i.e. standards, rather than cope with message exchanges in an ad-hoc manner.

This approach has led to a number of interesting, widely visible and well-accepted results, which are enumerated later as success stories and adopted standards.

3.1 Resource Discovery

Resource discovery is a critical service in any distributed system. Its primary goal is to support the coordination of decentralised resources, using general-purpose, standard protocols as well as interfaces that fully characterise the available resources (*resource profile*) and that allow their identification against given requirements. In service-oriented infrastructures such as those enabled by gCube, the problem of resource discovery is complicated further by the plethora of available resource types, and by the fact that resources do change very frequently. Resource discovery raises also management issues, including workload balancing, performance monitoring, and problem diagnosis.

To support the role of the D4Science infrastructure in the knowledge ecosystem, the resource discovery facilities in gCube have been enhanced in two directions. From the modelling perspective, the set of supported resource types has been enlarged and the profiling capabilities have been improved. In addition to gCube resources, the new resource model supports other types of resources such as "external" data sources and "external" services. From the discovery perspective, the architecture of the information services has been revised so as to cope with the increased workload, i.e. the number of resource profiles and discovery requests. New strategies for dynamic data partitioning and replication have been conceived along with new facilities for asynchronous production and paged consumption of discovery results.

3.2 Data Access

Data access services deal with the provision of content available in a given scope. These services complement data discovery services (cf. Sec. 3.3), as in standard interaction patterns discovery is preliminary to access.

To cope with the scenarios envisaged in Section 2, a new content management system has been conceived. The *Open Content Management Architecture* (*OCMA*) acknowledges that gCube is concerned with content that may: (*i*) be hosted inside or outside a gCube infrastructure; (*ii*) be described with a variety of models, for different media, and with different degrees of structure; and (*iii*) be accessed with a variety of protocols. This service provides its clients with seamless access to content that resides in multiple repositories and is designed to be open and easily adaptable to a variety of repositories. In terms of data model an OCMA service assumes that: (*i*) content is created, accessed, and distributed in units called *documents*; (*ii*) documents are grouped in *collections*; and (*iii*) collections are hosted in local management systems called *repositories*. As far as openness is concerned, OCMA services rely on plug-in based adaptivity facilities offered by gCube [28], i.e. specific plug-ins are developed to manage the interaction with external data providers.

3.3 Data Discovery

The gCube platform offers a rich set of Information Retrieval (IR) capabilities built on top of a stack of services that operate within the platform itself, taking advantage of the constructs, tools and methodologies that comprise it. While extending the interoperability of the gCube platform, however, Information Retrieval components have been enriched to accommodate data providers that live outside the gCube boundaries.

Traditionally, gCube Search has exploited external data providers via a mediator-based approach, i.e. the inclusion of custom operators in its search plans [27]. These operators are invoked to reply to an information discovery enquiry on such data sources. In essence, they can forward whichever query excerpt is needed to an external provider, retrieve the results, and integrate them with data hosted in the platform. These operators should (*i*) identify instances of the external engine, (*ii*) profile the external engine in the same way as internal gCube search providers are profiled and (*iii*) possibly wrap the retrieved content with the same constructs the internal services utilize for data representation and transport.

Although this approach can handle all cases, it involves the generation of custom operators for the adaptation of each new external engine type, thus forcing the system into an endless extension of the code base. The OpenSearch [23] specification is of great assistance to interoperability in the Information Retrieval domain. It finds its way in gCube as a new search operator implemented as a brokered OpenSearch client. The new facilities added include source discovery, template parsing, information retrieval and result paging and transformation in one servicing entity. The full range of OpenSearch criteria is exploited, including the geo-spatial / temporal criteria that are of major interest to the communities targeted by major data infrastructures.

Furthermore, the gCube Application Support Layer – the area where external access is granted to gCube facilities over common HTTP API, i.e. REST web services – has been extended to offer an OpenSearch provider interface that enables gCube's Information Retrieval capacities to be integrated into external search engines and OpenSearch consumers (e.g. web browsers).

3.4 Process Execution

One common need of data-oriented infrastructures is large data processing over owned or shared content. The gCube infrastructure goes beyond the limits of such a paradigm, however. Not only does it need and consume computational resources, it also manages and exposes resources offered by neighbouring Grid and Cloud infrastructures. Exploiting these resources and exposing them to other infrastructures is challenging from a functional point of view, and it requires addressing interoperability issues beyond a strictly systematic point of view.

By definition, a gCube infrastructure is comprised of a distributed set of nodes that can service computation under a wide range of technologies. These technologies can be Web

Services of several paradigms [14,32,6], technology-specific binaries, script executables, and they all raise a variety of requirements in terms of storage, deployed dependencies, communication and control, each representing permutations in terms of characteristics and behaviour. Coping with such a diverse environment calls for a component of largely unbounded expressiveness, which enforces even greater demands on the upper layers.

In order to facilitate the exploitation of these capacities whilst hiding the complexity of the landscape, at the core of the gCube platform lies a mechanism that is able to orchestrate flows of invocations on this large and diverse set of targets. This mechanism, named *Process Execution Engine* (*PE2ng*), builds on modern principles of data flow processing [31] appropriately expanded in the direction of interoperability. These include the plan (flow of execution), the operators (i.e. executable logic), the transport and control abstraction (implemented by the gCube Result Set – gRS), the containers (areas of execution), the *state* holders (e.g. storage), the *resource profiles* (definitions of resources characteristics for exploitation in a plan).

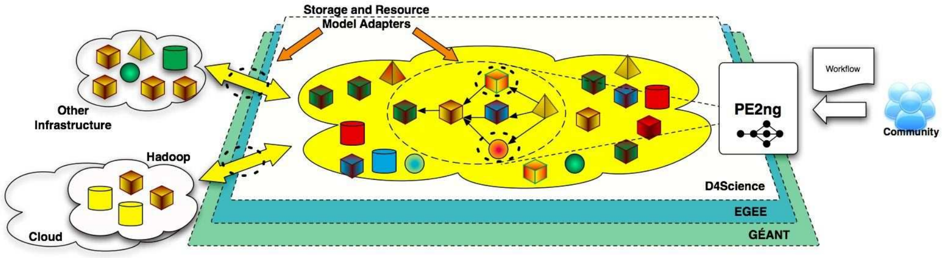

Fig. 2. PE2ng Architecture

PE2ng is suitable for modern Cloud computing infrastructures and it is capable of near-optimal exploitation of standalone and distributed computing resources, as it can employ selection and communication strategies transparently to its operators.

The PE2ng is not only able to execute flows consisting of executables of several technologies but it can operate as a gateway to other computational and storage infrastructures. The overall goal is to bridge gCube with heterogeneous platforms such as the underlying EGEE gLite grid [13], the Condor [29], the Hadoop [33] and more. By employing an architecture (cf. Fig. 2) that involves a number of abstractions (infrastructure adapters and storage providers) and a number of standards such as JSDL [1] for the specifications of the operations, PE2ng can serve effectively the aforementioned cause.

The *gCube Data Transformation Services* (gDTS) offer functionality to transform content and metadata into different formats and specifications. This functionality is essential to enable interoperability among Digital Libraries, as a number of facilities offered by Digital Libraries actually rely on the manifestations of the data they contain.

gDTS operates on top of the PE2ng, constructing on demand flows that are capable to transform content among two manifestations. Based on MIME-Type specifications that include “micro-format” refinements (e.g. qualities, sizes, encodings), gDTS is capable to locate paths that manage to migrate one content type to another and consequently employs PE2ng for carrying out the operation over the infrastructure.

A common use case of gDTS is its involvement in the preparation of content to be imported or exported to the Data Access and Retrieval mechanisms of gCube. The engine can also be exploited to apply transformations to objects of external infrastructures. In that case, the input data will not be imported from gCube services. Rather, they are supplied by external sources and made accessible over a set of common standards (e.g. ftp, http, grid ftp, a file system mount point). Depending on the use case the transformed content (and its metadata if such is the use) can be stored within the gCube infrastructure or sent back to the clients.

4 Evaluation

The new features of the gCube system, which are currently under development, are producing quite encouraging results, and success stories are growing by the day.

The new architecture for content management proved to fit the scope. An OAI-PMH [20] plug in has been developed and various OAI-PMH based data providers, including DRIVER, have been easily managed. In addition, the same protocol is used to further expose content previously aggregated in the D4Science infrastructure.

Information Retrieval interoperability via OpenSearch protocol is exploited in a number of cases. In the DRIVER / OpenAIRE [22] interoperability case, the external infrastructure IR services are exploited via basic OpenSearch compliant invocations. However, the GENESI-DR infrastructure[5] is more complex and a brokered approach is taken, leading initially to the discovery of data providers and subsequently to the results themselves. Finally, the gCube Information Retrieval is in itself exploited via OpenSearch and the reverse interoperability cases are also supported.

Data Transformation has allowed gCube to interoperate with numerous external sources provided by the communities of the aforementioned infrastructures. After being “imported”, the content is consumed by the gDTS, which takes care of transforming it into homogeneous formats that populate additional collections of the infrastructure (e.g. indexing). A more complex case is the provision on-demand of alternative representations of content that resides in external e-Infrastructures (e.g. thumbnails generation). In this case, large external datasets are temporarily “fed” to the D4Science e-Infrastructure and consumed by the gDTS service. The transformed results are then returned to the original e-Infrastructure.

Process Execution capacities are exploited in several scenarios, the most interesting of which from the interoperability is the INSPIRE - gCube scenario. Here, processing *Optical Character Recognition* (*OCR*) over large data volumes can be outsourced to the D4Science infrastructure, where it is made efficient by the underlying grid infrastructure. On the other side, daily reconstruction of inverted indices over an extended database requires the resource allocation strategies of the Cloud and constructs of the kind offered by a Map-Reduce Hadoop infrastructure. Somewhere in the middle, one could position the *Author Identification* case, which is less demanding in terms of data but requires parallel computation. This kind of process can be fruitfully executed on gCube worker nodes. In the AquaMaps scenario, model-based, large-scale predictions of marine species occurrences have to be generated. Predictions are generated by matching habitat usage of species, termed environmental envelopes, against local environmental conditions to determine the relative suitability of specific geographic areas for a given species. Data and algorithms are scattered along a

[5] `www.genesi-dr.eu`

number of gCube worker nodes to build up information-rich data sets. These are then further processed to construct biodiversity maps using the underlying Grid resources.

The major success of the Interoperability approach of gCube is the adoption of standards rather than proprietary protocols. As a result of this work a number of specifications are today exploited in a number of services and access points to the gCube platform, including OAI-PMH for data access, OpenSearch for data discovery, JDL for job submission, FTP / GridFTP / HTTP for storage access.

5 Conclusions and Future Work

Interoperability is one of the most critical issues that arise when building systems as "collections" of independently developed systems that should cooperate and rely on each other to accomplish larger tasks. An e-Infrastructure Ecosystem falls in this category of challenging systems and its realisation demands for the development of a rich array of interoperability approaches. In this paper, we have described the approaches put in place in the context of the D4Science-II. In particular, we have presented our interpretation of interoperability and the solutions put in place for resource discovery, data access, data discovery and process execution.

Current interoperability solutions in gCube are driven by the requirements and opportunities raised by the infrastructures addressed by the D4Science project, once these are seen under the perspective of generalization and standards adoption. We expect that this approach will maximise the usefulness of gCube in similar e-Infrastructure ecosystems. This assumption will be tested in other domains, for other e-Infrastructures and data providers have been already selected to enrich the D4Science knowledge ecosystem (e.g. 4D4Life[6], BioFresh[7]). The current solutions will be exploited to address the requirements that arise in these additional interoperability cases. The mechanisms for exposing gCube resources will be reinforced by supporting other standards and protocols. Mechanisms like the Linked Data [7] approach, the OAI-ORE [21] protocol and the SRU discovery protocol [30] are under investigation.

Acknowledgments The work reported has been partially supported by the D4Science project (FP7 of the European Commission, INFRA-2007-1.2.2, Contract No. 212488), the D4Science-II project (FP7 of the European Commission, INFRA-2008-1.2.2, Contract No. 239019) and the DL.org Coordination and Support Action (FP7 of the European Commission, ICT-2007.4.3, Contract No. 231551).

References

1. A. Anjomshoaa, F. Brisard, M. Drescher, D. Fellows, A. Ly, S. McGough, D. Pulsipher, and A. Savva. Job Submission Description Language (JSDL) Specification, Version 1.0. Technical Report GFD-R.056, Open Grid Forum, 2005.
2. O. Appleton, B. Jones, D. Kranzmuller, and E. Laure. The EGEE-II project: Evolution towards a permanent european grid inititative. 16:424–435, Mar. 2008.

[6] www.4d4life.eu

[7] http://www.freshwaterbiodiversity.eu/

3. M. Assante, L. Candela, D. Castelli, L. Frosini, L. Lelii, P. Manghi, A. Manzi, P. Pagano, and M. Simi. An Extensible Virtual Digital Libraries Generator. In B. Christensen-Dalsgaard, D. Castelli, B. A. Jurik, and J. Lippincott, editors, *12th European Conference on Research and Advanced Technology for Digital Libraries, ECDL 2008, Aarhus, Denmark, September 14-19*, volume 5173 of *Lecture Notes in Computer Science*, pages 122–134. Springer, 2008.
4. D. E. Atkins, K. K. Droegemeier, S. I. Feldman, H. García-Molina, M. L. Klein, D. G. Messerschmitt, P. Messina, J. P. Ostriker, and M. H. Wright. Revolutionizing science and engineering through cyberinfrastructure. Report of the National Science Foundation Blue-Ribbon Advisory Panel on Cyberinfrastructure, January 2003.
5. M. Atkinson, J. Crowcroft, C. Goble, J. Gurd, T. Rodden, N. Shadbolt, M. Sloman, I. Sommerville, and T. Storey. Computer Challenges to emerge from eScience. e-Science vision document., August 2002.
6. T. Banks. Web Services Resource Framework (WSRF) - Primer. Committee draft 01, OASIS, December 2005. `http://docs.oasis-open.org/wsrf/wsrf-primer-1.2-primer-cd-01.pdf`.
7. C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.
8. L. Candela. *Virtual Digital Libraries*. PhD thesis, Information Engineering Department, University of Pisa, 2006.
9. L. Candela, F. Akal, H. Avancini, D. Castelli, L. Fusco, V. Guidetti, C. Langguth, A. Manzi, P. Pagano, H. Schuldt, M. Simi, M. Springmann, and L. Voicu. DILIGENT: integrating Digital Library and Grid Technologies for a new Earth Observation Research Infrastructure. *International Journal on Digital Libraries*, 7(1-2):59–80, October 2007.
10. L. Candela, D. Castelli, and P. Pagano. On-demand Virtual Research Environments and the Changing Roles of Librarians. *Library Hi Tech*, 27(2):239–251, 2009.
11. L. Candela, D. Castelli, and C. Thanos. Making Digital Library Content Interoperable. In *Sixth Italian Research Conference on Digital Libraries (IRCDL 2010)*, 2010.
12. D. Castelli. D4Science-II - An e-Infrastructure Ecosystem for Science. *ERCIM News*, 79:9, October 2009.
13. EGEE. gLite: Lightweight Middleware for Grid Computing. `http://glite.web.cern.ch/glite/`.
14. R. Fielding. *Architectural Styles and the Design of Network-based Software Architectures*. Phd dissertation, University of California, Irvine, 2000.
15. T. A. Finholt. Collaboratories. In *Annual Review of Information Science and Technology*, volume 36, pages 73–107, 2005.
16. I. Foster and C. Kesselman. *The Grid: Blueprint for a Future Computing Infrastructure*. Morgan-Kaufmann, 2004.
17. E. A. Fox, R. M. Akscyn, R. Furuta, and J. J. Leggett. Digital Libraries. *Communications of the ACM*, 38(4):23–28, April 1995.
18. M. Fraser. Virtual Research Environments: Overview and Activity. *Ariadne*, 44, 2005.
19. A. Geraci. *IEEE Standard Computer Dictionary: Compilation of IEEE Standard Computer Glossaries*. IEEE Press, 1991.
20. C. Lagoze and H. Van de Sompel. The open archives initiative: building a low-barrier interoperability framework. In *Proceedings of the first ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 54–62. ACM Press, 2001.
21. C. Lagoze, H. Van de Sompel, P. Johnston, M. Nelson, R. Sanderson, and S. Warner. Open Archives Initiative Object Reuse and Exchange (OAI-ORE) User Guide - Primer. Technical report, Open Archives Initiative, 2008.
22. P. Manghi, M. Mikulicic, L. Candela, D. Castelli, and P. Pagano. Realizing and Maintaining Aggregative Digital Library Systems: D-NET Software Toolkit and OAIster System. *D-Lib Magazine*, 16(3/4), March/April 2010.
23. OpenSearch Community. OpenSearch. Project website.
24. A. Paepcke, C.-C. K. Chang, T. Winograd, and H. García-Molina. Interoperability for Digital Libraries Worldwide. *Communications of the ACM*, 41(4):33–42, 1998.

25. P. Pagano, F. Simeoni, M. Simi, and L. Candela. Taming development complexity in service-oriented e-Infrastructures: the gCore application framework and distribution for gCube. *Zero-In e-Infrastructure News Magazine*, 1(1):19 – 21, 2009.
26. E. Rahm and P. A. Bernstein. A survey of approaches to automatic schema matching. *The VLDB Journal*, 10(4):334–350, 2001.
27. F. Simeoni, L. Candela, G. Kakaletris, M. Sibeko, P. Pagano, G. Papanikos, P. Polydoras, Y. E. Ioannidis, D. Aarvaag, and F. Crestani. A Grid-Based Infrastructure for Distributed Retrieval. In L. Kovács, N. Fuhr, and C. Meghini, editors, *Research and Advanced Technology for Digital Libraries, 11th European Conference, ECDL 2007, Budapest, Hungary, September 16-21, 2007, Proceedings*, volume 4675 of *Lecture Notes in Computer Science*, pages 161–173. Springer-Verlag, 2007.
28. F. Simeoni, L. Candela, D. Lievens, P. Pagano, and M. Simi. Functional adaptivity for Digital Library Services in e-Infrastructures: the gCube Approach. In M. Agosti, J. Borbinha, S. Kapidakis, C. Papatheodorou, and G. Tsakonas, editors, *13th European Conference on Research and Advanced Technology for Digital Libraries, ECDL 2009*, 2009.
29. D. Thain, T. Tannenbaum, and M. Livny. Distributed computing in practice: the condor experience. *Concurrency - Practice and Experience*, 17(2-4):323–356, 2005.
30. The Library of Congress. Sru: Search/retrieval via url. Website.
31. M. Tsangaris, G. Kakaletris, H. Kllapi, G. Papanikos, F. Pentaris, P. Polydoras, E. Sitaridi, V. Stoumpos, and Y. Ioannidis. Dataflow Processing and Optimization on Grid and Cloud Infrastructures. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 32(1):67–74, March 2009.
32. W3C. Simple Object Access Protocol (SOAP).
33. T. White. *Hadoop: The Definitive Guide*. O'Reilly Media, 2009.
34. G. Wiederhold. Mediators in the Architecture of Future Information Systems. *Computer*, 25(3):38–49, 1992.
35. W. A. Wulf. The collaboratory opportunity. *Science*, 261:854–855, August 1993.