

Cross-middleware Interoperability in Distributed Concurrent Engineering

E. Rowland Watkins, Mark McArdle, Thomas Leonard, Mike SurrIDGE
IT Innovation Centre
{erw, mm, tal, ms}@it-innovation.soton.ac.uk

Abstract

Secure, distributed collaboration between different organizations is a key challenge in Grid computing today. The GDCD project has produced a Grid-based demonstrator Virtual Collaborative Facility (VCF) for the European Space Agency. The purpose of this work is to show the potential of Grid technology to support fully distributed concurrent design, while addressing practical considerations including network security, interoperability, and integration of legacy applications. The VCF allows domain engineers to use the concurrent design methodology in a distributed fashion to perform studies for future space missions.

To demonstrate the interoperability and integration capabilities of Grid computing in concurrent design, we developed prototype VCF components based on ESA's current Excel-based Concurrent Design Facility (a non-distributed environment), using a STEP-compliant database that stores design parameters. The database was exposed as a secure GRIA 5.1 Grid service, whilst a .NET/WSE3.0-based library was developed to enable secure communication between the Excel client and STEP database.

1. Introduction

The ESA Concurrent Design Facility (CDF) is a state-of-the-art facility equipped with networking and teleconferencing tools that enable teams of experts from different disciplines to apply the concurrent engineering method to the design of future space missions [1].

Located at ESA ESTEC in Noordwijk, The Netherlands, the CDF is regularly used to conduct pre-phase 1 [2] assessment studies for the design of new space vehicles. During a study, domain engineers share design parameters from a common Microsoft Excel 2003 [3] spreadsheet. To ensure integrity of the spreadsheet and prevent concurrent access, the study manager orchestrates access. Apart from concurrency issues, the current CDF suffers from the travel costs

incurred since team experts must travel to ESA ESTEC from different parts of the European Union.

The aim of the Grid Distributed Concurrent Design (GDCD) [4, 5] project was to provide a demonstrator system showing the potential of Grid technology to support fully distributed concurrent design, while addressing practical considerations including network security, interoperability with other technologies required, and integration of legacy applications. As part of this work, we took the existing CDF client software and Integrated Design Model (IDM), and provided a secure, distributed implementation of its capabilities using Grid technology, to support fully distributed collaboration. To accomplish this has meant careful consideration in the choice of architecture and technology that address interoperability considerations, yet enable ESA to re-distribute proposed solutions to Member States. To show the feasibility of the prototype Virtual Collaboration Facility (VCF) components, a live demonstration of the VCF was performed with ESA in June 2007.

This paper is organized as follows. Section 2 describes our approach to enabling the legacy Excel CDF client to communicate with secure Web Services. Section 3 introduces the problems that arise from middleware interoperability that have placed restrictions on the GDCD approach. In section 4 we discuss the dynamic security and policy management required by the VCF to manage access to multiple domain engineers taking part in the concurrent design process. In section 5 we discuss our experiences during the project and current technical limitations. We consider related work in section 6 and summarize in section 7.

2. Legacy Application Integration

One of the main tasks of the GDCD project has been to investigate the construction of VCF components using existing client software, notably the ESA CDF Microsoft Excel 2003 Visual Basic for Applications (VBA) [6] macros. These macros provide domain engineers with the necessary infrastructure for

middleware systems in the GDCD project we have been able to demonstrate interoperability between GRIA [20], based on Java and Apache Axis [21], and Microsoft's .NET platform with WSE 3.0 extensions.

Since neither of these two middleware has a common language or implementation, achieving interoperability has primarily been through the use of industry-recognized standards. As we have already noted in the previous section, integration of legacy applications in VBA pose their own interoperability challenges. The use of .NET 2.0/COM interoperability to connect Excel with WSE 3.0 is achievable, albeit to a limited capability as we discussed in the introduction.

3.2. GDCD Architecture Stack

Figure 2 shows the architecture stack used in the GDCD VCF. As can be seen, the VCF stack is built primarily on Web Services with extensions for security (WS-Security) and contextualization (WS-Addressing). Above the messaging layer we have a set of GRIA-based services that support a core set of capabilities needed in the VCF. GRIA supports out-of-the-box management and security services that align well with the security and business trust issues of the VCF. Persistence is also important as we shall see in the next session as this is used to support the VCF IDM database.

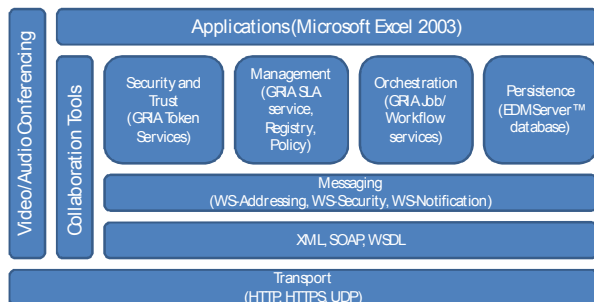


Figure 2. GDCD VCF Stack

Another interesting feature of this stack is the inclusion of collaborative and teleconferencing tools. Tools such as aveComm [22] and WebEx [23] have been tested in conjunction with the current VCF implementation.

3.1. Wire Interoperability

Although Web Services by definition should be interoperable, backwards compatibility with previous distributed computing paradigms like the Remote Procedure Call (RPC) has thrown up its own problems.

The SOAP specification [24] defines two different message styles, RPC and Document. In the RPC

(Remote Procedure Call) style the XML message corresponds to a function call in a programming language. The interface is defined as a series of arguments within an element representing the function call itself. In the Document style, the message can be any XML document, and the interface is specified using the XML schema language.

SOAP also offers the choice between encoded and literal formats. An encoded message can represent an arbitrary graph of objects, while the literal format restricts messages to a tree structure.

The WS-I organization aims to improve interoperability by providing clarifications and restrictions on existing specifications. The WS-I Basic Profile [25], for example, does not allow the encoded style (R1005-7), only the simpler literal style. Since .NET does not support the RPC/literal combination, this leaves Document/literal as the only viable format for SOAP messages. For GRIA services, we use a special Axis style called “wrapped”; this makes the service appear as a Document/literal style service to the outside world, while using the RPC style internally to simplify implementation of the service.

A service's interface is described using WSDL. The formats used by .NET and Axis 1.4 are compatible, provided that the document/literal style is used. It is important to use the same namespaces in SOAP messages and in the WSDL; Axis will accept elements that have the wrong namespace, but .NET will (correctly) ignore them. This issue can be fixed by adding serialization metadata to complex type class definitions.

Both .NET and Axis use the WS-Addressing specification for identifying service endpoints. This specification defines an Endpoint Reference (EPR) Type containing the URL to which messages should be sent, any additional SOAP headers which may be required, and a container for any extra metadata elements. Although WS-Addressing Core 1.0 was published in May 2006, GRIA uses a draft from August 2005, while .NET WSE 3.0 uses a draft from August 2004.

Different WS-Addressing version hampers interoperability when using contextualized services like those found in WSRF. Although .NET WSE 3.0 defines a WS-Addressing EPR type, it does not automatically deserialize XML EPRs of different versions into this type. Instead, when it reads the WSDL for a service it defines a new EPR type just for that service. It was then necessary to provide methods to convert between these two EPR representations within the .NET WSE 3.0 client code.

SOAP also allows for two different array styles. In the first, an array is represented by a single element, which contains a child element for each item of the

array. In the second, there is no explicit array element and items appear directly in the message. .NET only supports the second style, which must be enabled in Axis by using explicit metadata in the class definitions.

3.2. Security

Security at the message level can be handled in several ways depending on requirements for confidentiality and integrity. Message confidentiality can be supported using transport layer security (TLS) or WS-Security (WSS) encryption (XML Encryption). From a performance standpoint, TLS is more mature than WSS-based encryption. From a protocol standpoint, TLS is more accessible since it operates further down the protocol stack and is widely available.

Message level integrity can be supported using WS-Security (WSS) digital signatures. Both Axis (WSS4J) [26] and WSE 3.0 support WS-Security 1.1, which was released in 2004, so this caused no problems with version incompatibilities.

Apache's WSS4J requires elements in the WS-Security header to occur in a particular order. WSE 3.0 places the timestamp element before the signature, which causes WSS4J to reject the message. If WSS4J is configured to put the timestamp first then it fails to sign the timestamp. Therefore, we had to modify the WSS4J code to allow the timestamp to occur at any position in the WS-Security SOAP header, allowing both WSE 3.0 and Java clients to use the service. Although the WS-I Basic Security Profile 1.0 [27] constrains the order of some of the headers (R3212) it does not place requirements on the position of the timestamp.

WSE 3.0 also has to be configured to sign the WS-Addressing header giving the resource context of the request based on its use of policy assertions. A GRIA service requires this header to be signed to prevent an attacker from redirecting a valid request to a different resource. This is not an issue for service responses, as they do not contain any context headers.

4. Dynamic Security and Policy Management

We have so far described the integration approach taken in the GDCD project to connect ESA's Excel CDF client to a secure web service-based Grid infrastructure. What is now needed is a method of managing access to remote services of the VCF so that domain engineers can conduct concurrent engineering in an efficient manner.

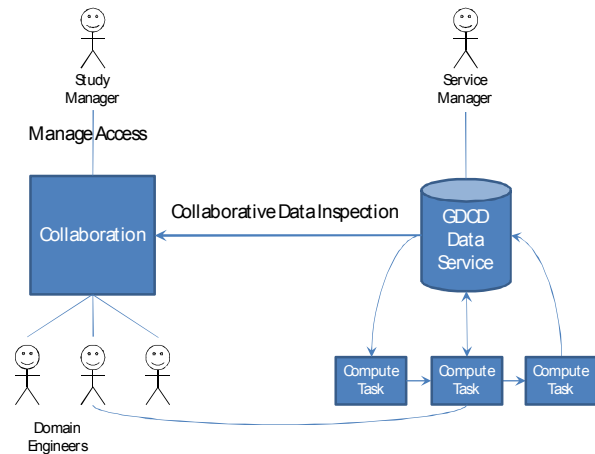


Figure 3. GDCD VCF Architecture

Figure 3 shows the GDCD VCF architecture with three types of actors: a study manager, several domain engineers and data service manager. Domain engineers collaborate together from their respective institution networks, orchestrated by the study manager who is also location independent. A typical CDF session is highly dynamic, and may even be somewhat informal, which presents no serious problems when everybody is in the same room at ESTEC. The challenge was to support this dynamicity while protecting against errors that would be easy to trap when everybody is local, but hard to detect when the users are distributed.

The solution was to use the dynamic security policy facilities provided by GRIA, which allows distributed management of security policies to control concurrent access. The study manager creates study sessions (Virtual Organization [28]) at the GDCD Data Service, and then permits domain engineers to retrieve and update parameters to the IDM as the study progresses. Domain engineers can also create and invoke distributed workflows that read or update the IDM. The service manager is not involved, but can override the access policies if he needs to (e.g. if he detects abuses of his service by one or more users). Access to the VCF IDM is thus controlled by the study manager, within constraints set by the data service manager at the hosting service provider site.

4.1. Concurrent Access

The main requirement for the VCF is that multiple domain engineers have concurrent access to a single volatile data source (the IDM) from remote sites, under the control of the study manager. This leads to two issues that must be addressed: synchronization and data redundancy.

Synchronized access to the IDM database is essential if domain engineers are to collaborate effectively. For example, parameter values produced by Power engineers affect System engineer calculations. If another engineer (Thermal) were able to overwrite the same parameter value, calculations will become inaccurate and devalue the concurrent engineering method.

Data redundancy is also an issue when local copies of parameters are not up-to-date with that of the IDM database. In the Excel CDF client most parameter values are linked using VBA dynamically. Out of date values will have an effect on a large amount of subsequent calculations, causing a cascading effect throughout domain engineer's Excel CDF client.

In the current CDF used by ESA, IDM parameters are synchronized by fetching updates from the data service when the study manager tells the collaboration that an update has been made. Domain engineers can't overwrite values out of turn, because the study manager can control the access policy at the IDM service.

4.2. GDCD Approach

Although domain engineers require "write" access to the IDM for parameter updates, they do not need this privilege throughout a study's lifecycle. The study manager will orchestrate write access similar to a version control system that supports *lock-update-unlock* primitives. Since the study is concurrent, this approach does not suffer from long-term locks inherent in distributed version control. As domain engineers complete their updates, the study manager will update the GDCD Data Service policies to permit write access for the next domain engineer. This means synchronization is enforced and that the access control mechanism used by the GDCD Data Service must support dynamic updates to security policies.

The approach taken in the GDCD project to satisfy dynamic security was to develop the GDCD Data Service as a GRIA 5.1 [29] service. This service served as a front end to EPM's EDMServer [30], which stored the IDM database. GRIA supports dynamic policy updates using its Process Based Access Control (PBAC) authorization mechanism [31].

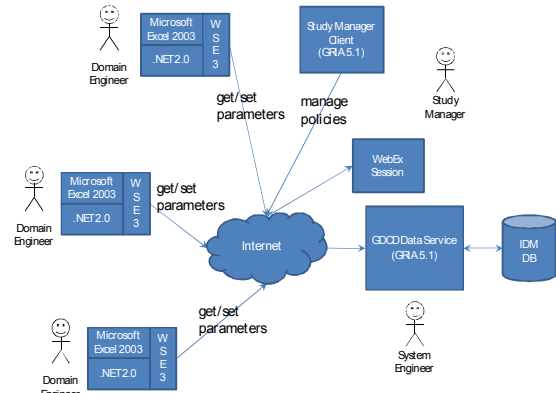


Figure 4. GDCD VCF Implementation

PBAC is capable of managing access to contextualized Web Services based on WS-Security and WS-Addressing. For each SOAP invocation, PBAC will check that a *subject-resource-operation* triple can be satisfied. The subject is defined by an authenticated attribute such as their Distinguished Name (DN) or a SAML assertion from the WS-Security header, the resource is a service-defined WS-Resource identifier found in a SOAP header. In the case of the VCF, the resource is a single study located at the GDCD Data Service, using PBAC subject DN authorization.

Figure 4 shows the current VCF implementation that demonstrates end-to-end connectivity between the Excel CDF client and the IDM database, and protected by the GDCD Data Service. The process for creating and managing a study session can be summarized as follows:

1. Study manager creates a VCF study session at GDCD Data Service and receives a WS-Addressing EPR containing the service address and the session's resource identifier;
2. Study manager collects credentials (e.g. X509 certificates) of the domain engineers he wants in the session, and adds access control rules to permit read access to the study session for these people;
3. Domain engineers discover study session EPR using GDCD Data Service `getResources()` method;
4. Domain engineers update Excel CDF clients with parameters from GDCD Data Service;
5. Study manager selects a "Systems" domain engineer to update parameter values, and enables write access at GDCD Data Service;
6. "Systems" domain engineer sets new parameter values;

7. Other domain engineers retrieve updates and incorporate them into their own Excel clients;
8. Study manager selects next domain engineer for write access, and so on.

It is clear from this description that there are three key access control roles being used to create study sessions and manage access to them. The roles used by the GDCD Data Service are:

- Manager – having this role allows granting and revocation of read and write roles;
- Read – having this role gives the domain engineer access to read parameter values from the VCF IDM;
- Write – having this role gives the domain engineer permissions to modify parameter values in the VCF IDM.

For convenience, we developed a simple graphical interface for the study manager to manage access control policies at the GDCD Data Service. This client extends the standard GRIA client GUI with a single-click interface that allows the study manager to set read and write roles.

4.3. VO Management

As we have already hinted, the creation of the study session is akin to creating a very dynamic and short-lived Virtual Organization (VO). The study manager sets up and controls the VO containing identities of domain engineers from different organizations that represent the VCF study team. Given the distributed nature of the VCF it is possible to have a large number of participating organizations whom may join and leave the study team based on availability.

5. Discussion

5.1. Experiences

Although we discovered inevitable issues with sending messages between WSE 3.0 and the GDCD Data Service (namespace consistency, order of elements in signed messages), we found the main interoperability challenge was not so much between WSE 3.0 and GRIA, but rather between the Excel CDF client and WSE 3.0. It became clear that while Microsoft had produced web service integration tools for VBA, they may have not anticipated the need for more advanced scenarios where WS-Security or WS-Addressing was needed. We showed that our approach to using COM interoperability with WSE 3.0 to be a

viable approach for connecting to contextualized Web Services despite the technical limitations placed on the approach.

5.2. Technical Limitations

Technical limitations with COM/WSE 3.0 integration meant that certificate management was a real challenge in a Windows environment. Turn-key WS-Security policy assertions provided by WSE 3.0 could not be flexibly configured in a COM environment. This placed restrictions on where certificates and CRLs could be located when verifying response messages.

Since the Excel CDF client and WSE 3.0 component had to be installed on every domain engineer's workstation, there were usability issues regarding the installation of certificates, CAs and CRLs. For example, CRLs for the GDCD Data Service had to be placed in a very specific location (Console/Local Machine/Intermediate Certification Authorities) otherwise WSE 3.0 would not be able to verify response messages.

6. Related Work

Several Grid interoperability initiatives have taken place in the past few years, looking at general message-level interoperation for job submission and data transfer. Notable examples include the Grid Interoperability Project (GRIP) between Globus and UNICORE [32], as well as, the EUChinaGrid project between gLite and CNGrid GOS [33]. The above initiatives, however, all concentrate on eScience, rather than the concurrent design environment used by ESA.

One notable parallel to the work done in GDCD can be found in the NextGRID project, which has had some experience in cross-middleware interoperability within its derivatives pricing experiment [34]. This experiment is also based on an Excel client talking to secure Web Services, although not yet in a concurrent, collaborative environment.

7. Summary

The aim of the GDCD project was to present to ESA ESTEC a new approach to distributed collaborative engineering using Grid technology. The result of our work, as described in this paper, was a prototype VCF that used an existing Excel CDF client and successfully enabled it to send secure SOAP messages to a remote Grid service controlling the IDM database. Access to this remote GDCD Data Service was controlled by a dynamic access control mechanism

provided by the GRIA grid middleware that enabled distributed concurrent engineering to take place in a highly dynamic way, with very low set up overheads.

While our approach has been successful, there are obvious improvements that can be made to the VCF beyond the proposed Open Concurrent Design Server (OCDS) [35]. These include tighter .NET integration, federated security models, and better end-user deployment.

Full integration using VSTA/VSTO would be desirable in an Excel environment, such that the VBA macros were ported into a CLR language such as VB.NET. Such integration would remove all the constraints placed on WSE 3.0 by COM interoperability with the additional option of using Windows Communication Foundation.

Our use of dynamic policy updates, while essential to support the concurrent nature of the VCF, could be improved by introducing federated role-based security using SAML tokens based WS-Trust and WS-Federation specifications. This is supported by GRIA 5.1, but was not used in the GDCD environment because it requires additional WS-Trust security token services at the domain engineer sites.

Lastly, there is a need to improve the usability of certificate management for end users in a way that can be used in legacy (VBA) applications. It would be preferable that keystores, certificates, CAs and CRLs were not store in MMC, rather in a known location on the file system. This would, however, require custom message validation on response messages in WSE 3.0.

8. Acknowledgements

This work was under the GDCD project, funded by European Space Agency under the General Studies Programme.

The GDCD project is carried out by a consortium led by DATAMAT S.p.A. and including Thales Alenia Space France, Thales Alenia Space Italia S.p.A., Det Norske Veritas AS, Jotne EPM Technology AS and University of Southampton IT Innovation Centre.

References

[1] ESA CDF, <http://www.esa.int/SPECIALS/CDF/>.

[2] M. Bandecchi, B. Melton, B. Gardini and F. Ongaro, "The ESA/ESTEC Concurrent Design Facility", In Proceedings of 2nd European Systems Engineering Conference (EuSEC 2000), München, <http://esamultimedia.esa.int/docs/cdf/cdf.pdf>, retrieved on 17.07.2007.

[3] Microsoft Excel 2003 Product Information, <http://www.microsoft.com/uk/office/excel/prodinfo/default.msp>, retrieved on 17.07.2007.

[4] Carlo Paccagnini, Andrea Martelli, Stefano Beco, Massimo Bandecchi, "GDCD: Grid-based Distributed Concurrent Design", DASIA 2006 - DATA Systems In Aerospace, Berlin, Germany, 22 - 25 May 2006.

[5] Stefano Beco, Andrea Parrini, Carlo Paccagnini, "Architecture of a Grid-based Virtual Collaborative Facility for Space Projects", 2nd Concurrent Engineering for Space Applications Workshop 2006, ESA ESTEC, Noordwijk, The Netherlands, 19 - 20 October 2006.

[6] Microsoft Visual Basic for Applications, <http://msdn2.microsoft.com/en-us/isv/Bb190538.aspx>, retrieved on 17.07.2007.

[7] M. Bandecchi, "The ESA Concurrent Design Facility", 2nd Nordic Systems Engineering Boat Seminar, September 2001, Helsinki, <http://www.finse.org/2001Autumn/presentations/Bandecchi.pdf>, retrieved on 17.07.2007.

[8] ISO 10303-1:1994 Industrial automation systems and integration Product data representation and exchange - Overview and Fundamental Principles, International Standard, ISO TC184/SC4, 1994.

[9] Web Services Interoperability Organization, <http://www.ws-i.org/>, retrieved on 19.07.2007.

[10] Tim Banks, "Web Services Resource Framework – Primer v1.2", <http://docs.oasis-open.org/wsr/wsr-primer-1.2-primer-cd-02.pdf>, retrieved on 19.07.2007.

[11] Microsoft Office 2003 Web Services Toolkit 2.01, <http://www.microsoft.com/downloads/details.aspx?familyid=fa36018a-e1cf-48a3-9b35-169d819ecf18&displaylang=en#Overview>, retrieved on 17.07.2007.

[12] Microsoft Visual Studio Tools for Applications, <http://msdn2.microsoft.com/isv/aa700828.aspx>, retrieved on 17.07.2007.

[13] Microsoft Visual Studio Tools for Office, <http://msdn2.microsoft.com/office/aa905543.aspx>, retrieved on 17.07.2007.

[14] .NET Framework Developer Center, <http://msdn2.microsoft.com/en-us/netframework/default.aspx>, retrieved on 17.07.2007.

[15] Microsoft .NET Framework Developer's Guide: Exposing .NET Components to COM, [http://msdn2.microsoft.com/en-us/library/zsfww439\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/zsfww439(VS.80).aspx), retrieved on 17.07.2007.

- [16] Web Service Enhancements for Microsoft .NET, <http://msdn2.microsoft.com/en-us/webservices/Aa740663.aspx>, retrieved on 18.07.2007.
- [17] Martin Gudgin, Marc Hadley, and Tony Rogers (Editors), “Web Services Addressing 1.0 – Core”, W3C Recommendation, 9 May 2006, <http://www.w3.org/TR/ws-addr-core/>, retrieved on 19.07.2007.
- [18] Anthony Nadalin, Chris Kaler, Ronald Monzillo, and Phillip Hallam-Baker (Editors), “Web Services Security: SOAP Message Security 1.1”, OASIS Standard, 1 February 2006, <http://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf>, retrieved on 19.07.2007.
- [19] Anthony Nadalin, Marc Goodner, Martin Gudgin, Abbie Barbir, and Hans Granqvist (Editors), “WS-SecureConversation 1.3”, OASIS Standard, 1 March 2007, <http://docs.oasis-open.org/ws-sx/ws-secureconversation/v1.3/ws-secureconversation.pdf>, retrieved on 19.07.2007.
- [20] M. Surridge, S. Taylor, D De Roure, and E. Zaluska, (2005), “Experiences with GRIA — Industrial Applications on a Web Services Grid”, in Proceedings of the First International Conference on e-Science and Grid Computing, pp. 98-105. IEEE Press.
- [21] Apache <Web Services/> Project, <http://ws.apache.org/axis/>, retrieved on 19.07.2007.
- [22] Atinav aveComm, <http://www.avecomm.com/>, retrieved on 19.07.2007.
- [23] WebEx, <http://www.webex.com/>, retrieved on 19.07.2007.
- [24] Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen, Anish Karmarkar, and Yves Lafon (Editors), “SOAP Version 1.2 Part 1: Messaging Framework (Second Edition)”, W3C Recommendation, 27 April 2007, <http://www.w3.org/TR/soap12-part1/>, retrieved on 19.07.2007.
- [25] Keith Ballinger, David Ehnebuske, Christopher Ferris, Martin Gudgin, Canyon Kevin Liu, Mark Nottingham, and Prasad Yendluri (Editors), WS-I Standard, “Basic Profile Version 1.1”, <http://www.ws-i.org/Profiles/BasicProfile-1.1.html>, retrieved on 18.07.2007.
- [26] Web Services Security for Java (WSS4J), <http://ws.apache.org/wss4j/>, retrieved on 19.07.2007.
- [27] Michael McIntosh, Martin Gudgin, K. Scott Morrison, and Abbie Barbir (Editors), “Basic Security Profile Version 1.0”, WS-I Standard, 30 March 2007, <http://www.ws-i.org/Profiles/BasicSecurityProfile-1.0.html>, retrieved on 19.07.2007.
- [28] Ian Foster, Carl Kesselman, Steven Tueke, “The Anatomy of the Grid: Enabling Scalable Virtual Organizations”, International Journal of Supercomputer Applications, 15(3), 2001. <http://www.globus.org/alliance/publications/papers/anatomy.pdf>, retrieved on 19.07.2007.
- [29] GRIA - Service Oriented Collaborations for Industry and Commerce, <http://www.gria.org>, retrieved on 19.07.2007.
- [30] Jotne EPM Technology, “EDMServer”, <http://www.epmtech.jotne.com/products/edmserver.html>, retrieved on 17.07.2007.
- [31] GRIA PBAC 2 Manual, <http://www.gria.org/documentation/manual/pbac-2-manual>, retrieved on 19.07.2007.
- [32] D. A. Nicole, “UNICORE and GRIP: Experiences of Grid Middleware Development”. In Proceedings of 2005 International Conference on Grid Computing and Applications, pp. 11-17, Las Vegas, Nevada, USA. Arabnia, H. R. and Ni, J., Eds.
- [33] Interconnection & Interoperability of Grids between Europe and China, <http://www.euchinagrid.org/docs/EUChinaGRID-Project-Presentation.pdf>, retrieved on 19.07.2007.
- [34] Mark Parsons and Guy Lonsdale, “Feedback from NextGRID Applications Experiments”, EGR-RG, OGF20, Manchester, UK, 09.09.2007, http://forge.ogf.org/sf/docman/do/downloadDocument/projects_egr-rg/docman.root.meeting_materials.meeting_materials_ogf20/doc14493;jsessionid=DC9AC762CA4FF004377BDCC41279CFAD, retrieved on 17.07.2007.
- [35] M. Bandecchi and A. Matthyssen, “ESA Open Concurrent Design Server”, 2nd Concurrent Engineering for Space Applications Workshop 2006, ESA ESTEC, Noordwijk, The Netherlands, 19 – 20 October 2006, http://esamultimedia.esa.int/docs/2006-10-24_AbstractsBook-WebsiteVersion.pdf, retrieved on 18.07.2007.