Interactive Grid Computing - Adapting HLA–based Applications to the Grid

Katarzyna Rycerz[1,2], Marian Bubak[1,2], Maciej Malawski[1,2], Peter Sloot[3]

Institute of Computer Science, AGH, al.Mickiewicza 30,30-059 Kraków
Academic Computer Centre – CYFRONET, Nawojki 11,30-950 Kraków, Poland
Faculty of Sciences, Section Computational Science, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
kzajac|bubak|malawski@uci.agh.edu.pl
sloot@science.uva.nl

Abstract. In this paper we present the design of a Grid HLA Management System (GHMS) supporting execution of HLA–based interactive applications in a Grid environment, and, at the same time to preserve backward compatibility with the HLA standard allowing for running HLA legacy codes. In particular, we describe the following system components: the *HLA–Speaking Service* for multiple federates that interface the HLA application to the system, *Monitoring Service* integrated with the OCM-G monitoring system and HLA–based *Benchmark Services* informing the *Broker Service* of what can be expected from application behavior.

Keywords: interactive simulation, Grid Computing, HLA, OGSA

1 Introduction

Interactive Grid Computing is a very interesting aspect of modern computer science. The definition of interactivity is very broad and a possible Grid architecture for supporting such applications can be found in [3]. In this paper we focus on applications composed of distributed elements, where one or more elements are interaction components that interface with a human and thus their behavior cannot be easily predicted. A user uses interaction components to steer simulation components of the application in near-real time. We assume that communication between the components is performed using one of the most common standards for distributed interactive applications – High Level Architecture (HLA)[13]. The conceptual architecture of such an application is shown in the Fig.1. As an example we refer to the ISS-Conductor [27] system that is a piece of middleware built over HLA RTI for interactive applications such as the CrossGrid medical application [18].

This kind of applications usually requires extensive computing resources. The Grid [9, 10] gives the opportunity for better and more convenient usage of distributed resources that were previously inaccessible. However, existing distributed simulation community standards for large-scale simulations are not yet suitable for direct usage on the Grid.

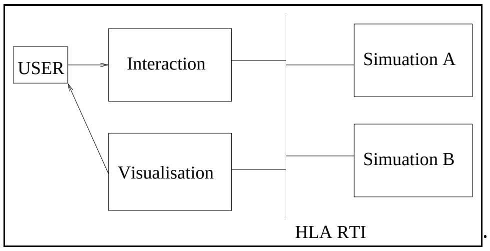

Fig. 1. The Conceptual Architecture of Distributed Interactive Applications

The HLA [13] supports development of simulations (called federations in HLA terminology) from distributed components (called federates). It provides the simulation developer with many useful features such as time and data management required by time– or event–driven distributed simulations. However, the HLA standard was developed assuming a certain quality of service in the underlying environment of simulation execution. It is based on the assumption that the environment it executes on is faultless. By definition, a Grid environment is a set of shared resources that cannot be strictly dedicated to near-real time applications and therefore the behavior of such applications can vary depending on circumstances. Also, the Grid is unreliable and requires fault tolerance mechanisms.

This paper is organized as follows in Section: 2 we provide a brief HLA overview, in Section 3 we outline related work, in Section 4 we present the overall architecture of the system. In the next three sections we describe its essential components. The service that interfaces HLA application to the system is presented in Section 5. The service that monitors execution of the HLA application supporting the *Broker Service* when the application is performing badly and needs to be migrated is shown in Section 6. Last but not least, the service that measures potential behavior of HLA applications on the Grid and supporting the broker in its decisions on where to migrate is described in Section 7. We conclude the paper in Section 8.

2 HLA as a support for distributed interactive simulations

HLA Runtime Infrastructure (RTI) federations [13] are distributed systems that consist of multiple processes (federates) communicating across computer nodes. HLA provides application developers with an RTI that acts as a tuple space. In order to send messages, the applications that are plugged into the RTI have to publish well-defined objects in that space. The applications that receive messages have to subscribe to those objects. Each time an object is to be sent, the application must call a method that updates the attribute values of this object; RTI then notifies the subscribed applications that the object has been updated. This approach allows for easy plugging in of late-arriving components into the running system. HLA also supports time synchronization (i.e. for time-driven or event-driven simulations) and data distribution management between distributed components of the interactive application. Federates are controlled by the RTI Executive (RTIexec) process.

In HLA there is no mechanism for migrating federates according to the dynamic changes of host loads or failures. This means that HLA does not guarantee fault tolerance in an unreliable Grid environment.

3 Related work

The modeling and simulation community realized that there is no widely accepted standard that can handle both DOD Modeling and Simulation standards like HLA [13] and Web/IT standards [23]. It was noticed that current solutions in HLA implementations such as DoD HLA-RTI do not perform efficiently in wide area networks and traditional approaches to high-performance RTI implementations assume relatively static configurations of federates. This was the motivation behind the XMSF project [25], aiming to develop a common standard for Web and DOD Modeling and Simulation. As a prototype, a web-enabled RTI has been developed, being basically a Web service wrapper over existing RTI functionality. It serves as an example to create more general XMSF profiles. These profiles are planned as documents describing a set of protocols, data and metadata standards used for the application domain, and a detailed description of applying protocols and data standards to implement the architecture. Unfortunately, the final specification of XMSF profiles has not yet been defined.

In parallel, research effort is going on within the Commercial Off-The-Shelf Simulation (COTS) Package Interoperability Forum (HLA-CSPIF) [6]. As there is no real standard use pattern for the High Level Architecture within the context of this area, the ultimate goal of the Forum is to create standards that will facilitate interoperation of COTS simulation packages.

Next, there is research related to supporting management of HLA–distributed simulations. Cai et al., [4] implemented a load management system for running large-scale HLA simulations in a Grid environment based on Globus Toolkit 2 and they also present a framework where the modeler can design parallel and distributed simulations with no knowledge of HLA/RTI [20]. In [21] they presented a protocol that supports efficient checkpointing and federate migration for dynamic load balancing.

Further effort towards a scalable and fault-tolerant framework for HLA–based applications is being undertaken by the Swedish Defence Research Agency (FOI), where the Decentralized Resource Management System (DRMS) is under development. DRMS is a JXTA-based peer-to-peer system for the execution of HLA federations in a decentralized environment.

In [16], the authors propose a fault tolerant resource sharing system (FT-RSS) for HLA–based simulations. The framework and its components allow the individual configuration of FT mechanisms to be included in federates and federations. In the design and implementation of a fault tolerant federation, the developers are supported by an FT configuration tool. During the execution of a distributed HLA simulation using the FT-RSS, the manager and client components of the FT-RSS are responsible for the enforcement of FT mechanisms. There is also work on another approach that uses the HLA concept to build a scalable peer-to-peer infrastructure [22], proposed by FederationGrid (FedGrid), which is built over the standard High Level Architecture and is based on the concept of hierarchical HLA federations. It implements a scalable Grid supporting real-time collaborative applications.

The works presented above are related to the issue considered in this paper, however, none of them present an environment designed for running interactive applications on the Grid based on the Grid Services approach. Only the approach presented in [4, 20] bears significant relation to our proposal, but it does introduce the concept of an additional layer between HLA and actual applications in order to facilitate execution management. Although it allows for developing efficient migration protocols [21], it is not sufficient for porting HLA legacy code to the Grid. In contrast, the approach taken in this project allows not only for building HLA–based applications that can be executed on the Grid, but also for easy adaptation of HLA legacy code.

4 Design of Grid HLA Management System

4.1 Requirements of interactive applications

In our research we focus on "human in the loop" simulations, which mean that a user can **steer** the simulation **while it is running** by providing some input data online. This approach is suitable for the kind of applications, where the **parameter space is not known before runtime** and therefore parameter study approaches cannot be applied. Three main requirements of such applications may be defined:

1. Distributed simulations require specific functionality, such as time management for **event–driven or time–driven simulations** and data distribution management for convenient data exchange between distributed components of the simulation. This issue can be solved by a system supporting interaction on the application level.
2. Distributed interactive simulations belong to the field of High Performance Computing (HPC) in the sense that they require a certain level of quality in the execution environment, that allows for **near-real time communication** between **distributed** components.
3. Usually, these applications consists of modules with different functionalities and require **access to distributed resources** which additionally turns them into High Throughput Computing (HTC) applications.

To fulfill requirements specific for interactive simulations, such as time and data management services, we have decided to use existing solutions for distributed and interactive simulation systems. We chose the High Level Architecture (HLA) [13] as it is an IEEE standard well recognized in the area of distributed simulations and it offers all necessary functionality for simulation developers. HLA fulfills the first requirement mentioned above, but it does not include any management support for the simulation execution in its underlying environment and assumes a certain quality of service. Using it is therefore only a partial solution to the problem.

One of the most important concepts that strongly influence the scientific distributed computing area is Grid computing, which has emerged as an important and rapidly expanding approach to distributed systems in the past few years [2]. Ian Foster [9] has presented a checkpoint list defining a Grid as a system that coordinates resources that are not subject to centralized control, but where coordination should be done by means of standard, open, general-purpose protocols and should deliver nontrivial quality of service. This approach gives a potential opportunity for better and more convenient usage of distributed resources that were previously inaccessible, and can fulfill the requirements of distributed simulations. However, current Grid environments lack the ability to fully support near-real-time applications. When no network reservation mechanisms are present, it becomes difficult to achieve a certain quality of service while assuring near-real-time responses to the human in a processing loop. Although various QoS mechanisms are technically possible, they are often not present in Grid environments and require support from network devices as well as additional effort to set them up, which can limit access to some of the resources for which it is difficult or impossible (for various reasons).

It is often very difficult to predict the performance of a task in a shared Grid environment. If no reservation or prioritization mechanisms are available, HPC applications require support for dynamic adaptation to the changes in the execution environment and thus call for a system that would control their execution. Unfortunately, the Globus Resource Allocation Manager (GRAM) [12], which is the Grid interface to local management systems, does not support all of the features of the underlying batch system. A variety of useful features, such as the ability to checkpoint, are missing. Condor-G [19], used by the DataGrid resource broker [7], has the same disadvantages as GRAM. Also, in the current OGSI-enabled Globus Toolkit 3 [10], which actually exposes GRAM functionality as a service, the same problems arise.

During previous research, the **Grid HLA Management System (GHMS)** has been presented as a continuation of a system proposed in [26] to **support efficient execution of HLA–based interactive simulations in the Grid** environment in spite of the issues presented above. **GHMS fills the gap between interactive simulations and Grid Infrastructure** as shown in Fig. 2. GHMS is placed between HLA and Grid layers. HLA supports distributed interactive simulations on the application level, supplying the developer with necessary services related to time and data management. Grid infrastructure is the middleware that provides access to available resources like computational power, storage systems etc. GHMS prepares the Grid environment to be efficiently used by interactive simulations.

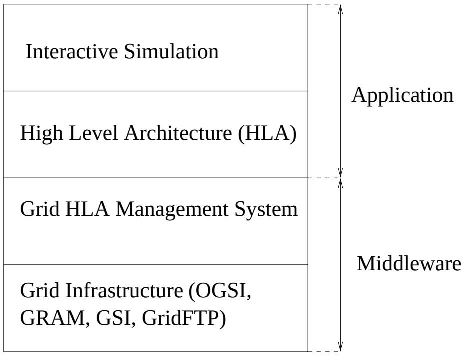

Fig. 2. The Grid HLA Management System in the context

4.2 System Overwiev

In previous papers [26, 17] we presented an overall concept of a system architecture that supports running distributed interactive applications based on the HLA standard in a Grid environment. In [26] we focus on a prototype of the *HLA–Speaking* service that interfaces the system with actual code of one federate. Management includes saving and restoring fedarate state from a checkpoint file.

In [17] we proposed HLA–based benchmarks as possible support for choosing resources for HLA–based applications on the Grid. In this paper we present an advanced architecture including a redesigned *HLA–speaking service* that can manage all federates from an application that are executed on its site. We also describe in more detail the system component responsible for monitoring federates in order to help the *Broker Service* decide when the applications starts to perform badly and needs to change its location.

Our current experimental architecture is based on the OGSA/OGSI standard, since it has been in the mainstream of the Grid effort. Although new concepts such as WSRF are approaching [24], we expect that the architecture described here, based on Grid and Web Services is invariant to such changes in standards. The main idea behind WSRF is to converge concepts of Grid and Web Services.

The components of the architecture can be categorized as follows:
Services global per application:
The Broker Service responsible for setting up and controlling all federates within application basing on information from the external *Registry Service*, as well as information from services described below, application monitoring components that include the *Main Service Manager* responsible for gathering information from *Service Managers* residing on each site and presenting it to the *Performance Decision Service* which triggers the *Broker Service* to perform the action of migration.
Services that should reside on each Grid site :

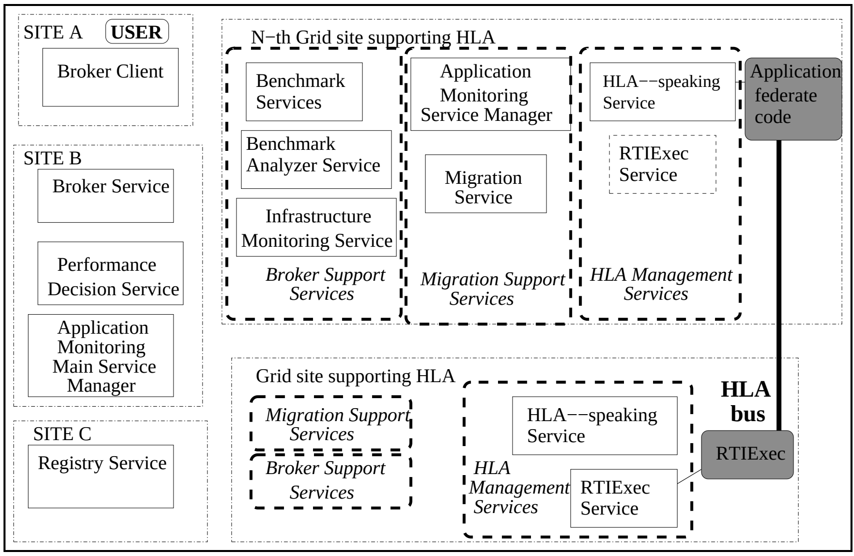

Fig. 3. The Grid HLA Management System

The *HLA Management Services* that interfaces and manages actual HLA installations on that site. This group includes the *HLA–Speaking Service* that represents HLA installations on a particular Grid site and the *RTIexec service* that interfaces with the RTIexec process. A more detailed description of the advanced *HLA–Speaking Service* that can manage multiple federates running on its site can be found in Section 5.

The *Broker Support Services* that aim at providing the *Broker Service* with information necessary for decisions about setup and migration of application components. The *Broker Service* should take into account information from *Infrastructure Monitoring Services* present in the Grid environment that provide it with host load information within a single Grid site. This research is part of the CrossGrid project [5], which also involves infrastructure monitoring tools [14]. In [17] we described *Benchmarks Services* as another possible source of information about HLA–based application behavior.

The *Migration Support Services* that provide direct support for fault tolerant and effective performance of HLA–based applications. This group includes *Local Monitors* and the *Migration Service*. Once the *Broker Service* decides where to migrate, the actual action is done by the *Migration Service* that asks the *HLA Speaking Service* to checkpoint states of federates that are running on that site. Next, the *Migration Service* starts the *HLA Speaking Service* on the remote site it wants to migrate to, transfers the code and checkpoint files and restarts the federates. A technical description of the HLA migration process can be found in the previous paper [26].

5 HLA-Speaking Service as an interface to federates

The *HLA–Speaking Service* is a service residing on a Grid site that interfaces HLA federates executing on that site. The communication scheme is depicted in Fig. 4. The *HLASpeaking Service* communicates with control federates via control federation that is created only for that purpose and it is invisible to the users. Apart from that, application federates communicate with each other by joining federations specific for their applications. *HLA–Speaking Services* communicate with other components of the system using the Grid Services interface. The *Broker Service* can invoke on the *HLA Speaking service* any operations that start application federates on its site, save the state of selected federates and restore them from a checkpoint file. The *HLA Speaking Service* is designed to cooperate with GRAM in order to start the application processes on its site. Save and restore requests are sent via the control HLA federation.

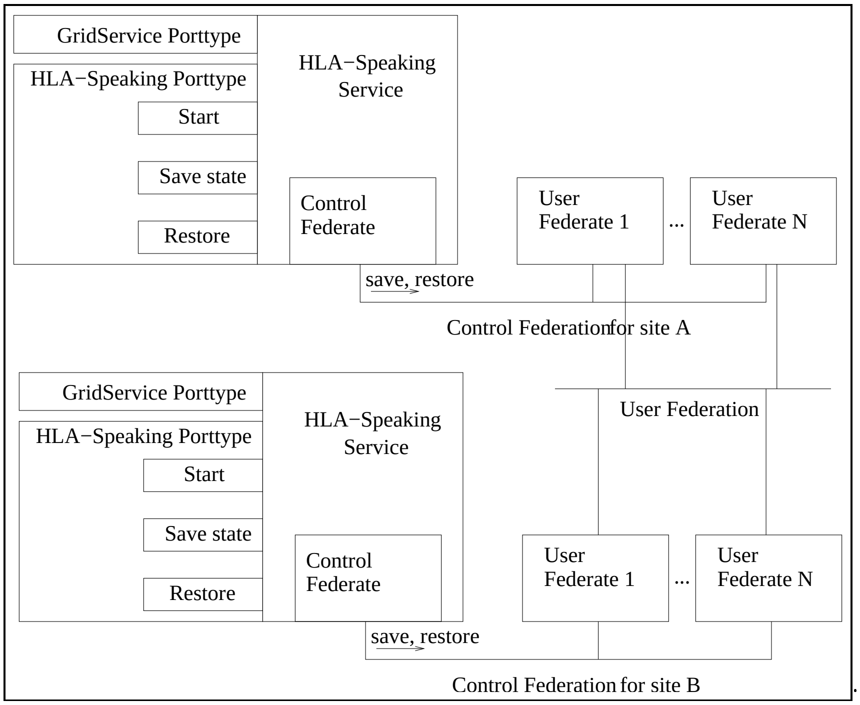

Fig. 4. HLA–Speaking Service for multiple federates

6 When to migrate ? – HLA Application Monitoring Services

For monitoring the application in order to extract performance information, we use the OCM-G monitoring system [1] developed in the CrossGrid project [5].The OCM-G is an autonomous, distributed, decentralized system which exposes monitoring services via a standardized OMIS interface [15].

The OCM-G is composed of two types of components: per-host Local Monitors and per-site Service Managers. Additionally, there is a Main Service Manager which distributes data to and collects it from site-SMs. The components are shown in Fig.3.

By using OCM-G we can gather all information required by *Decision Performance Service* that communicate with the *Main Service Manager* using the OMIS[15] protocol. The system can monitor the invocation of all HLA services: federation management, data management, data distribution management, time management and ownership management. The system allows for gathering data required for performance measurement such as the amount of data sent, time of sending data and process identifiers. This allows the *Decision Performance Measurement Service* to discover which federates participate in communication within the HLA tuple space and to decide if their location on the Grid is sufficient for achieving the performance needed by the user in a processing loop.

The architecture of our HLA monitoring system [**?**] is as follows. Between the HLA Applications and the RTIambassador we insert an additional class, which looks for applications similar to the original RTIambassador. This class has to communicate with the original RTI (it has to be invisible to the application), and to send data about it to the OCM-G Monitoring service.

The OCM-G architecture allows to control the amount and frequency of monitored data sent from the monitored application to the *Decision Performance Measurement Service.* It is scalable and suitable for running in a Grid environment.

7 Where to migrate? – Infrastructure Monitoring and Benchmark Services for HLA applications

Network performance is important for the operation of applications with a human in a processing loop. One of possible solutions to this problem is to use a prediction system such as the Network Weather Service (NWS) [11] or network monitoring systems like JIMS [14].

We have decided to develop our own benchmark suite to experimentally infer the type of behavior that can be expected from the Grid environment for HLA–based interactive applications. Details can be found in [17]. We focus on the scenario present in the CrossGrid biomedical application [5]. This application consists of three components: interaction, simulation and visualization. While simulation is performed, a user can change its parameters using the interaction module (sending small messages from interaction to simulation) and see the results through visualization (simulation sends large updates to visualization). According to this scenario, our benchmark consists of two components: an integrated Interaction-Visualization Component (IVC) and a Simulation Component (SC). The IVC sends small messages, which are human interactive requests to change selected SC parameters. The SC responses are large messages that present the altered state of the simulation to the user. Certainly, this scenario is suitable only for some classes of applications. In the future, we are going to extend our set of benchmarks.

The HLA benchmark federation consists of two federates, acting as IVC and SC respectively. IVC sends the SC requests in which it asks for certain amounts of data and measures the time between sending the request and receiving a response. Messages are sent through updates of HLA data objects [13]. The benchmark is managed within the Grid Services framework by the *Benchmark_IVC Service* and the *Benchmark_SC Service*, that start, control and measure performance of IVC and SC respectively.

Apart from the *Benchmark Services* our architecture also contains a *Benchmark Analyzer Service*. The goal of this service is to gather statistics based on *Benchmark Services* results.

8 Summary and Future Work

This paper presented an approach to Interactive Simulations on the Grid. As runtime support for interactive simulations we have chosen the HLA standard [13]. For efficient execution of HLA–based applications on the Grid we have proposed a Grid HLA Management System. We focus on important parts of the system, namely the *HLA–Speaking Service* for multiple federates that interface the HLA application to the system, the *Monitoring Service* integrated with the OCM-G monitoring system and HLA–based *Benchmark Services* informing the *Broker Service* what can be expected from application behavior. In the future, we plan to develop the following:

Broker Service Performing adequate brokering decisions in spite of the highly-changeable nature of the network and host environments is a nontrivial task, although benchmark results can provide valuable support. We plan to develop a *Broker Service* architecture and associated algorithms. For that purpose, it is also necessary to have a service that will analyze benchmark results according to specific needs of the broker.

Performance Decision Service. There is a clear need for a service that will analyze the performance of an application basing on data gathered by the monitoring system to supply the *Broker Service* with information on whether and when migration is needed.

Infrastructure Monitoring. Apart from benchmark services we also plan to analyze the use of existing infrastructure monitoring [14] and prediction services [11] for our purposes.

Acknowledgments The authors wish to thank Piotr Nowakowski for additional remarks. This research is partly funded by the European Commission IST-2001-32243 Project "CrossGrid".

References

1. Baliś, B., Bubak, M., Funika, W., Szczepieniec, T., Wismüller, R.: Monitoring and Performance Analysis of Grid Applications, in: Sloot, P. M. A., et al. (Eds.), Proceedings of Computational Science - ICCS 2003, International Conference Melbourne, Australia and St. Petersburg, Russia, June 2003, vol. I, no 2657, Lecture Notes in Computer Science, Springer, 2003, pp. 214-224
2. Berman, F., Fox, G., and Hey, T. (Eds): Grid Computing. Making the Global Infrastructure a Reality. Wiley 2003;
3. Bubak, M., Malawski, M., Zając, K.: Architecture of the Grid for Interactive Applications, in: Sloot, P. M. A., et al. (Eds.), Proceedings of Computational Science - ICCS 2003, International Conference Melbourne, Australia and St. Petersburg, Russia, June 2003, vol. I, no 2657, Lecture Notes in Computer Science, Springer, 2003, pp. 207-213
4. Cai,W., Turner, S.J., Zhao, H.: A Load Management System for Running HLA-based Distributed Simulations over the Grid in : Proceedings of the 6th IEEE International Workshop on Distributed Simulation and Real-Time Applications 2002 IEEE, IEEE Computer Society, Fort Worth, Texas, October 2002, pp.7-14
5. Crossgrid project web page. http://www.crossgrid.org/
6. HLA CSPIF home page: http://www.cspif.com
7. DataGrid Project Home Page
http://www.datagrid.com
8. Eklöf, M., Sparf, M., Moradi, F: HLA Federation Management in a Peer-to-Peer Environment in In Proceedings of the 03 European Simulation Interoperability Workshop, organized by the Simulation Interoperability Standardization Organization. Stockholm, 2003. http://www.sisostds.org/
9. Foster, I.: What is the Grid? A Three Point Checklist. GRIDToday, July 20, 2002 http://www.globus.org
10. Foster I, Kesselman C., Nick J., Tuecke S.: The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002
11. Gaidioz, B., Wolski, R., and Tourancheau, B.,: Synchronizing Network Probes to avoid Measurement Intrusiveness with the Network Weather Service, in : Proceedings of 9th IEEE High-performance Distributed Computing Conference, August, 2000, pp. 147-154
12. Globus Project Homepage http://www.globus.org
13. HLA specification, http://www.sisostds.org/stdsdev/hla/
14. Ławniczek, B., Majka, G., Słowikowski, P., Zieli'nski, K., Zieliński S.: Grid Infrastructure Monitoring Service Framework Jiro/JMX Based Implementation. Electronic Notes in Theoretical Computer Science. 82(6): (2003)
15. Ludwig, T. , Wismüller, R., Sunderam, V., and Bode, A.: OMIS — On-line Monitoring Interface Specification (Version 2.0) Technische Universität München july, 1997, Munich, Germany, TUM-I9733, SFB-Bericht Nr. 342/22/97 A,
http://wwwbode.in.tum.de/˜omis/

16. Lütchi, J., Grossman, S.: FT-RSS: A Flexible Framework for Fault Tolerant HLA Federations in: Bubak, M.; Albada, G.D.v.; Sloot, P.M.A.; Dongarra, J. (Eds.) : Proceedings of the 4th International Conference on Computational Science, Kraków, Poland, June 6-9, 2004, Lecture Notes in Computer Science, Vol. 3038, pp.865-872 Springer-Verlag, Heidelberg, 2004.
17. Rycerz, K., Bubak, M., Malawski, M., Sloot, P.: Support for Effective and Fault Tolerant Execution of HLA–based Applications in the OGSA Framework. accepted for Workshop on HLA-Based Distributed Simulation on the Grid International Conference on Computational Science 2004 (ICCS 2004) Krakow, Poland, June 7-9, 2004
18. Sloot, P.M.A., Tirado-Ramos, A. Hoekstra,A.G., and Bubak M.: An Interactive Grid Environment for Non-Invasive Vascular Reconstruction, in 2nd International Workshop on Biomedical Computations on the Grid (BioGrid'04), in conjunction with Fourth IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGrid2004), IEEE, Chicago, Illinois, USA, April 2004. ISBN 0-7803-8431-8.
19. Thain D., Tannenbaum T., and Livny M: Condor and the Grid, in Fran Berman, Anthony J.G. Hey, Geoffrey Fox, editors, Grid Computing: Making The Global Infrastructure a Reality, John Wiley, 2003. ISBN: 0-470-85319-0
20. Yuan, Z., Cai, W. and Low, M.Y.H.: A Framework for Executing Parallel Simulation using RTI published in: in: S.J. Turner and S.J.E. Taylor, editors, Proceedings Seventh IEEE International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2003), IEEE Computer Society, Delft, The Netherlands, October 2003, pp. 20-28
21. Yuan, Z., Cai, W., Low, M.Y.H., Turner, S.J.: Federate Migration in HLA-Based Simulation in: Bubak, M.; Albada, G.D.v.; Sloot, P.M.A.; Dongarra, J. (Eds.) : Proceedings of the 4th International Conference on Computational Science, Kraków, Poland, June 6-9, 2004, Lecture Notes in Computer Science, Vol. 3038 pp 856-864. Springer-Verlag, Heidelberg, 2004
22. Vuong, S., Cai, X., Li, J., Pramanik, S., Suttles, D., Chen, R.: FedGrid: An HLA Approach to Federating Grids in: Bubak, M.; Albada, G.D.v.; Sloot, P.M.A.; Dongarra, J. (Eds.) : Proceedings of the 4th International Conference on Computational Science, Kraków, Poland, June 6-9, 2004, Lecture Notes in Computer Science, Vol. 3038 pp. 889-896. Springer-Verlag, Heidelberg, 2004
23. Web services, http://www.w3.org/2002/ws/
24. WS-Resource Framework project home page http://www.globus.org/wsrf
25. XMSF project home site. http://www.movesinstitute.org/xmsf/xmsf.html
26. Zając, K., Bubak, M., Malawski, M., Sloot, P.: Towards a Grid Management System for HLA–based Interactive Simulations, in: S.J. Turner and S.J.E. Taylor, editors, Proceedings Seventh IEEE International Symposium on Distributed Simulation and Real Time Applications (DS-RT 2003), IEEE Computer Society, Delft, The Netherlands, October 2003, pp. 4-11.
27. Zhao, Z., van Albada, G. D., Tirado-Ramos, A., Zając, K., Sloot, P. M. A., ISS-Studio: A Prototype for a User-Friendly Tool for Designing Interactive Experiments in Problem Solving Environments, in: Sloot, P. M. A., et al. (Eds.), Proceedings of Computational Science - ICCS 2003, International Conference Melbourne, Australia and St. Petersburg, Russia, June 2003, vol. I, no. 2657, Lecture Notes in Computer Science, Springer, 2003, pp. 679-688