

# Edutain@Grid: A Business Grid Infrastructure for Real-Time On-Line Interactive Applications

Justin Ferris<sup>1</sup>, Mike SurrIDGE<sup>1</sup>, E. Rowland Watkins<sup>1</sup>, Thomas Fahringer<sup>2</sup>,  
Radu Prodan<sup>2</sup>, Frank Glinka<sup>3</sup>, Sergei Gorlatch<sup>3</sup>, Christoph Anthes<sup>4</sup>,  
Alexis Arragon<sup>5</sup>, Chris Rawlings<sup>6</sup>, and Arton Lipaj<sup>7</sup>

<sup>1</sup> IT Innovation Centre, University of Southampton, UK  
{jff,ms,erw}@it-innovation.soton.ac.uk

<sup>2</sup> Institute for Computer Science, University of Innsbruck, Austria  
{tf,radu}@dps.uibk.ac.at

<sup>3</sup> Institute of Computer Science, University of Münster, Germany  
{glinkaf,gorlatch}@math.uni.muenster.de

<sup>4</sup> Institute of Graphics and Parallel Processing, University of Linz, Austria  
canthes@gup.jku.at

<sup>5</sup> Darkworks S.A., France  
a.arragon@darkworks.com

<sup>6</sup> BMT Cordah Ltd., UK  
chris.rawlings@bmtcordah.com

<sup>7</sup> Amis d.o.o., Slovenia  
arton.lipaj@amis.net

**Abstract.** Grid infrastructures are maturing to a point where they are attracting the interest of businesses in many application domains. While many large-scale on-line gaming platforms exist, they fail to take into consideration the potential business to business relationships when it comes to dynamic on-line game hosting. This work presents an initial implementation of the edutain@grid architecture to support business value chains identified for on-line gaming and e-learning application hosting. An analysis of business actors and value chains is presented briefly before a detailed description of the edutain@grid implementation. We also consider first results concerning how best to construct appropriate value chains using bipartite and bi-directional Service Level Agreements.

**Keywords:** Business models, Service Level Agreements, Grid, Trust, Security, Value chains.

## 1 Introduction

Emerging Grid technologies [1] have the capability to substantially enhance on-line games and similar applications. Just as the World Wide Web enables people to share content over standard, open protocols, the Grid enables people and organizations to share applications, data and computing power over the Internet in order to collaborate, tackle large problems and lower the cost of computing.

The edutain@grid project [2, 3] aims to develop a novel, sophisticated and service-oriented Grid infrastructure which provides a generic, scalable, reliable and secure

service infrastructure for a new class of ‘killer’ applications of the Grid: Real-Time, On-Line, Interactive Applications (ROIA). ROIA include a broad sub-class of commercially important applications based on virtual environments, including massively multiplayer on-line gaming applications (MMOG), and interactive training and other e-learning applications. The `edutain@grid` project is aiming to provide an infrastructure to make such applications easier to develop, more economic to deploy and operate, and more capable of meeting the Quality of Experience expected and demanded by end-users.

Grid middleware systems such as Globus [4], gLite [5] and UNICORE [6] enable high-throughput applications by sharing computational resources for processing and data storage to meet the needs of individual and institutional users. ROIA such as multiplayer on-line computer games are soft real-time systems with very high user interactivity between users. Large numbers of users may participate in a single ROIA instance, and are typically able to join or leave at any time. Thus ROIA typically have extremely dynamic distributed workloads, making it difficult to host them efficiently. Initiatives such as Butterfly Grid [7] and Bigworld [8] have applied Grid computing to on-line gaming with some success, enabling ‘scalable’ or ‘elastic’ terms for hosting such games. However, these ‘scalable’ hosting services are only as scalable as the hoster supporting them, and typically don’t guarantee how far this will be. The `edutain@grid` project addresses these challenges using ‘business Grid’ developments such as GRIA [9, 10], but extending them to support scalable, multi-hosted ROIA applications, allowing scaling beyond the limits of any one hoster.

The rest of this paper is organised as follows. In section 2 we present an overview of the business models and actors supported by `edutain@grid`, and highlight some of the business issues addressed by the project. Section 3 describes the implementation of the `edutain@grid` framework in more detail, and discusses the initial results and their implications. Section 4 provides a summary of the overall work on `edutain@grid` value chains to date, and discusses the direction of future work.

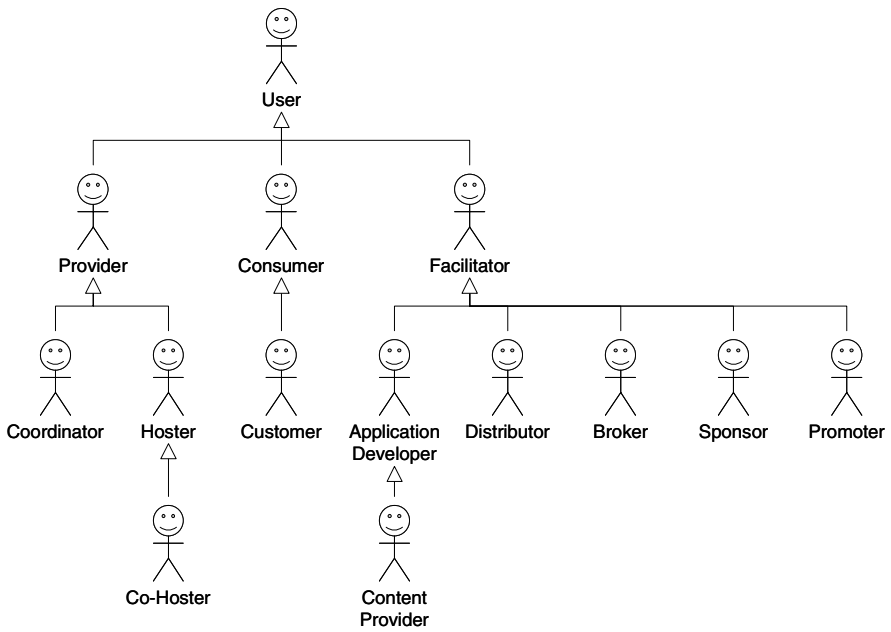
## 2 Business Actors and Value Chains in `Edutain@Grid`

To ensure business models for Grid-based ROIA will be economically viable, it is necessary to analyse the value chains (i.e. business actors and value flows) in which ROIA (specifically on-line games and e-learning applications) will be operated and used. The goal of `edutain@grid` is to support value chains corresponding to commercially viable scenarios, preferably in such a way that the same ROIA application software need not become locked into one particular business scenario.

The business actors (roles) supported by the `edutain@grid` infrastructure must be generic enough to meet the needs of both application sectors, and flexible enough to allow business models to be tuned to best fit the market conditions which may vary even within each sector. The analysis revealed an extensive hierarchy of business roles, as shown in Figure 1. These include ‘providers’ who host services through which the ROIA is delivered to users, ‘consumers’ who access the ROIA by connecting to these services, and ‘facilitators’ who play other business roles in the creation of ROIA application software, its distribution to providers and consumers, and the operation of ROIA instances.

Three important sub-classes of ROIA providers were also identified that have to be supported by the project:

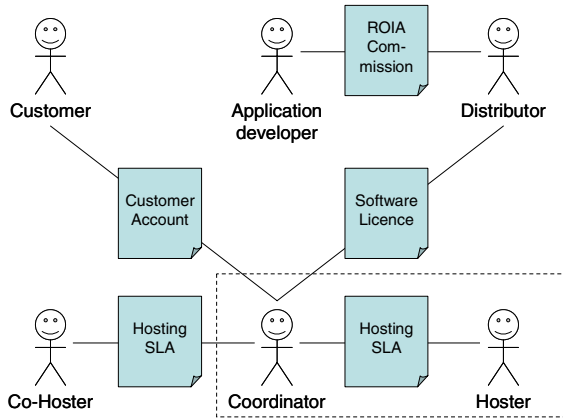
- **Hoster:** is an organisation that hosts core, usually computationally intensive processes that support a ROIA virtual environment including interactions of users with this environment and with each other.
- **Co-hosters:** are other hosters participating in the same ROIA instance – where more than one hoster is involved in a single ROIA instance, each hoster will regard the others as ‘co-hosters’ of the ROIA instance.
- **Coordinator:** is an organisation that makes a ROIA instance accessible to its consumers, and coordinates one or more hosters to deliver the required ROIA virtual interactive environment.



**Fig. 1.** Business Actors in edutain@grid

Today, on-line game hosters exist, but there are no ‘co-hosters’ or ‘coordinators’ because there is only one hoster per game instance. The edutain@grid infrastructure breaks away from this limitation, enabling new business models to manage risks of ROIA hosting and delivery, and provide genuine scalability for ROIA provision.

To achieve these benefits, edutain@grid allows actors to combine in a wide range of ‘value chains’ through which ROIA software and services are produced, deployed and delivered to the end users. The links in these value chains are defined through business agreements: either Service Level Agreements (SLA) or in some cases software licence agreements. These agreements are always bipartite, following the pattern used in the NextGRID project [11, 12] and used with GRIA in the SIMDAT project



**Fig. 2.** A typical edutain@grid value chain

[13]. Each agreement is between two parties only, and defines the service provided by one party and the obligations on both sides as to how this service will be delivered and used. The details of each agreement depends partly on the topology of the value chain through which funds flow from the customers (who ultimately pay for everything) to the other actors. The edutain@grid project allows for a wide range of topologies, of which a typical example is shown in Figure 2.

In this example, the ROIA application is commissioned by a distributor from an application developer. The distributor licenses coordinators to use and sub-licence the software. The coordinators pay hosters for the capacity to run ROIA services, and sell access to these services to customers, sub-licensing the required software to each as part of the deal with each. In the example shown, the coordinator also hosts some of the resources needed to run the ROIA, shown by the dotted line indicating that the coordinator and one of the hosters are actually the same organisation. Other co-hosters can be brought in to handle peaks in demand, or if the ROIA becomes so popular that one organisation cannot host it all any more.

### 3 Implementation and First Results

#### 3.1 Scope and Architecture

The edutain@grid project has produced a first implementation of the framework to support these business actors and value chains. The prototype focuses on the core edutain@grid actors: the coordinator, the hoster (or co-hoster), and the customer. The framework is based on a Service Oriented Architecture, organised in four layers, as shown in Figure 3. The real-time layer provides a framework [14] for ROIA developers to create scalable applications capable of running across multiple sites. The management layer deals with the allocation and management of resources (and ROIA processes) by hosters. The business layer handles setting up and enforcing the terms

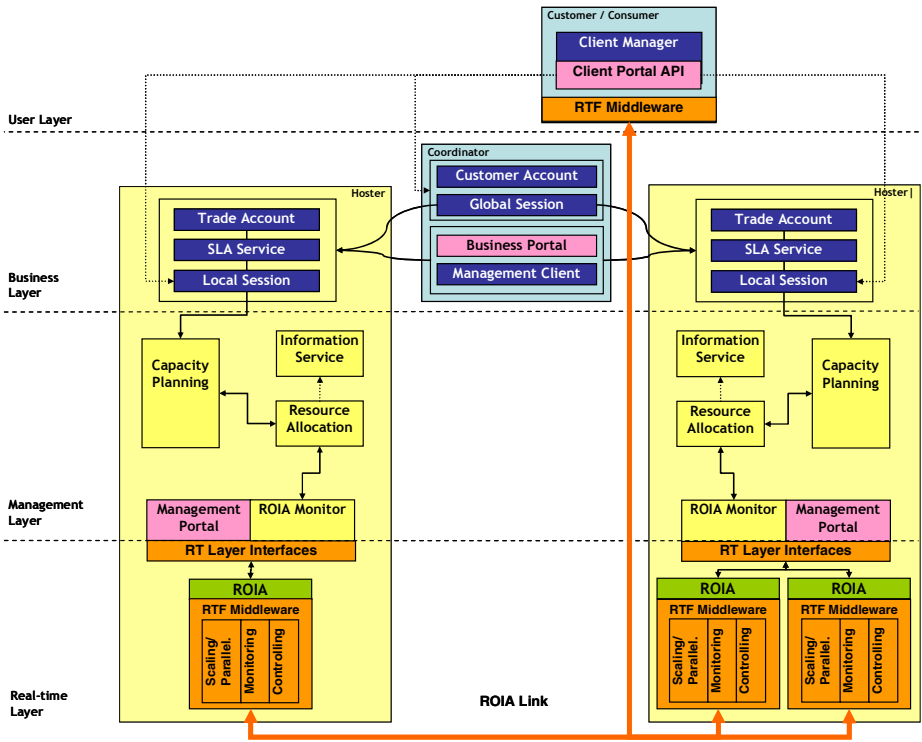


Fig. 3. Prototype edutain@grid architecture

of SLA (including hosting SLA and customer agreements). The business layer is also responsible for enforcing security policies, including the need to sign up to an appropriate SLA before accessing ROIA functionality. There is also a client layer which provides programming interfaces to use services from the other three layers.

### 3.2 Business Layer

The business layer is responsible for establishing and managing value chains through which ROIA services can be provided and consumed. This is based on management services from the GRIA 5.2 middleware [10] with some custom ROIA services developed by the edutain@grid project.

In this first edutain@grid prototype implementation, facilitator roles have not been considered, and it has been assumed that customers will only pay for themselves to access ROIA. This makes it possible to use a lightweight user authentication mechanism based on their ability to pay, because no consumers should have access unless they are paying. This authentication procedure is handled by the Customer Account service which is run by the coordinator. It acts as a WS-Trust token validator service allowing other business layer services to validate credentials supplied to them (which may include self-signed tokens), by comparing them with the credentials used by that Customer when setting up their account. The Trade Account service has a similar role

at the hoster site, authenticating coordinators based on their ability to pay for services, and allowing them to establish SLA with the SLA service if this is the case. The Global and Local Session services are then used to mediate access to ROIA instances. The Global Session service provides the entry point for customers and coordinates the involvement of hosters, including setting up security policies at each hoster to restrict access to genuine Customers. The Local Session is used to manage and provide access to ROIA processes contributed by its hoster. The Global Session service is also responsible for propagating information between the co-hosters where more than one hoster (and Local Session) is involved in a single ROIA instance.

Finally, note that even with a fixed set of possible actors, a large variety of value chains and business models is possible, and different arrangements may be needed to suit different applications or even the same application in different market conditions. The `edutain@grid` infrastructure therefore has to be flexible and able to support a range of topologies and business models, even if each application uses only a single agreement topology and business model.

### 3.3 Management Layer

The management layer is responsible for predicting the QoS that can be delivered by available resources, monitoring and managing these resources to enforce QoS terms, and translating monitoring events back into business layer terminology so it can be provided to business layer services that handle SLA negotiation and cross-hoster coordination. This allows the negotiation and enforcement of QoS parameters such as minimum latency, maximum user load, or minimum update rate in highly scalable ROIA that may involve thousands or someday millions of online user connections to hosts running in multiple hoster sites.

Even when using the Grid to provision resources, delivering the required real-time QoS needs remains a challenging task since MMOGs (especially FPS games) are highly dynamic and users may become concentrated in each other's proximity within a short period of time causing excessive server load and loss of QoS. To address this, it is possible to change parallelisation strategies using rezoning or zone replication and instancing, but this is an expensive operation that may take many seconds. The management layer must therefore anticipate potential QoS breaches so corrective action can be initiated before the breach has occurred.

To solve these challenges, the management layer will use a resource provisioning strategy based on four distinct services:

- A monitoring service that interacts with the real-time layer and logs ROIA execution information such as the number, position and interactions of entities within the virtualised environment.
- A load prediction service that estimates the future distribution of entities based on historical monitoring information, allowing server congestion hot-spots to be predicted ahead of time.
- A capacity management service that estimates ROIA session load based on entity distribution and interaction data, and plans the use of resources to fulfil QoS requirements while maintaining hoster-specified metrics (e.g. utilisation).
- A resource allocation service that provisions the required CPU, memory and network resources required by each ROIA session.

In future the management layer will also provide support for competitive as well as collaborative interactions within virtual organisations of users, including prevention of cheating within the ROIA.

### 3.4 Real-Time Layer

The real-time layer provides special services to facilitate the development of ROIA that are executable in a grid environment under soft real-time constraints. These services are combined in the C++ based *Real-Time-Framework* (RTF) [15] and include, among others, a grid-aware communication infrastructure, integrated business- and management-related monitoring and controlling facilities, as well as a sophisticated API for the systematic, high-level development of scalable, distributed ROIA. The RTF integrated services will enable a ROIA developer to create ROIA using high level abstractions to deal with the distributed and dynamic nature of the application, as well as the resource management and deployment aspects of the underlying infrastructure (Grid).

The real-time layer provides the communication infrastructure to connect the resources used by ROIA processes, based on a communication protocol which is highly optimized with respect to the low-latency and low-overhead requirements of typical ROIA. This infrastructure is Grid-aware, so communication endpoints can be transparently redirected to a new resource, if, e.g., parts of the ROIA are relocated to a new Grid resource for load-balancing reasons. Security aspects are taken care of by supporting authentication and encryption of real-time communications. This is mandatory as customers that connect through the Internet will be charged for using a ROIA, and must be traceable via their real- or pseudo-identities as established in the business layer.

The RTF offers an API which provides the development of ROIA using different parallelization approaches: zoning, instancing and replication for the scalable application distribution across multiple resources. The API provides an abstraction from the underlying resources: RTF (re)distributes the zones, instances and replication instances [16, 17] transparently for the customers during runtime, as advised by the business- or management layer. This redistribution functionality is realized in a non-disruptive way which ensures that a ROIA, which often requires the adherence of client-server response times of 100ms and less, is not interrupted during the redistribution process.

The high level of abstraction allows RTF to monitor certain real-time application metrics transparently for the developer. These metrics include, e.g., the number of transferred in-application events, the number of in-application objects, minimal response time for client requests, the number of connections, communication latency and bandwidth usage, and the virtual environment update frequency. These and other metrics are provided to the business and management layer through a dedicated interface at the border of the real-time layer, which decouples the real-time sensitive ROIA within the real-time layer from the slower, upper management and business services.

The integrated monitoring functionality of RTF provides the management and business layer with the information that is required for appropriate management and business decisions. The monitoring also is required to observe the QoS-related terms of an SLA and to check if a certain term was breached. The available distribution steering functionality of RTF provides decision opportunities to the hoster and coordinator: they can reallocate their resources as required and most commercially viable for them [18]. Dynamic resource allocation is very important as the customer demand for the access to a ROIA typically varies depending on daytime or holidays. This advantage also applies if the user base is not changing and allows a hoster to tremendously improve his resource utilization.

### 3.5 Initial Results

At this stage, the `edutain@grid` framework is being used to perform experiments to determine what types of hosting SLA terms will be most useful in on-line gaming applications, what kinds of resource management strategies can deliver QoS promised in these SLA, and what kinds of dynamic adaptation facilities from the real-time layer will allow applications to exploit the provisioned resources efficiently. The challenge is to define terms that coordinators find useful in managing the risks of over/under-estimating user demand, yet allow hosters to retain control over their own resources and implement efficient (ideally autonomic) management processes to control QoS by exploiting application-level adaptive behaviour. Since `edutain@grid` allows co-hosting scenarios, this involves going well beyond existing (even Grid-based) on-line game hosting environments, as it is necessary to take account of possible interactions and dependencies on co-hosters.

The GRIA middleware supports SLA terms based on arbitrary (capacity-oriented) quality of service metrics, but up to now these have been used only with traditional data storage and processing applications using traditional metrics such as disk storage and CPU time. Only a few users have defined metrics on less traditional items such as the number of floating software licences that can be used for commercial codes [19]. In `edutain@grid`, it is already clear that these types of metrics are not very useful (or valuable) in a multi-hosted ROIA scenario, as they make it too difficult for the coordinator to work out whether their customer expectations can be met, and force the hoster to cede too many internal resource management decisions to the coordinator. Experiments are now focusing on quite higher level QoS terms for hosting SLA which allow the coordinator to manage ROIA Quality of Experience, but without needing to control (or even understand) the resources and management strategy at each hoster.

This also means the project has to investigate how a hoster could manage resources to deliver the required ROIA performance, when the limits on usage are defined in terms of customer and application behaviour. It is clear that if very few customers are connected to the ROIA, the hoster can use the freedom inherent in such an SLA to reduce the resources allocated – e.g. by running multiple ROIA processes on a single host. However, a ROIA may become computationally expensive for purely internal reasons – e.g. through an increased level of interaction. If the SLA doesn't specify a limit on resources, the hoster would be obliged to allocate more resources to maintain the specified ROIA performance. It is also possible for the ROIA itself to induce SLA



breaches. For example, if two co-hosters agree to support up to (say) 600 user connections each, they will be able to support (say) 1000 customers between them. But if the ROIA causes all the customers' virtual avatars to gather in one zone of the ROIA virtual environment, then they may all end up having to connect to just one hoster, breaching the limits agreed by the coordinator with that hoster. The hoster would be within their rights to start refusing connections, destroying the Quality of Experience for many of the customers.

The edutain@grid project is now starting to investigate how advanced (including predictive) resource management mechanisms and application adaptation can be used to address both types of problems. For example, is it possible to predict a gathering of on-line gamers in one location, and inform the coordinator in time for hosting SLAs to be re-negotiated, possibly involving additional hosters? If the management layer can predict problems well enough to do this, how would the QoS terms need to change to reflect a predictive exception handling facility for an otherwise constrained level of service?

## 4 Summary and Future Work

The edutain@grid project aims to create a new class of 'killer application' for the Grid: Real-time On-line Interactive Applications (ROIA). This class spans several commercially important applications, including on-line gaming and simulator-based training, both of which are being used in validation case studies in the project.

The project has investigated the need for value chains between business actors to deliver ROIA in a Grid-based environment. The analysis has led to a separation between the roles of the hoster (who hosts ROIA services) and the coordinator (who sells ROIA access to customers and guarantees their Quality of Experience), which makes it possible to support co-hosted (and hence more scalable) ROIA, as well as conventional single-hosted ROIA (in which a business acts as both hoster and coordinator). The edutain@grid architecture has been designed to be flexible enough to support a wide range of value chain topologies among the roles identified, and to accommodate facilitators such as brokers where such roles are economically viable.

The initial implementation of the edutain@grid framework is now complete, and experiments are being conducted to investigate how business values can be expressed in SLA terms that allow service providers to retain flexibility and control costs, while being attractive to service consumers. Initial findings suggest that the hosting SLA between ROIA coordinators and hosters should be expressed in terms of the outcomes for the coordinator, as more conventional SLA terms based on resource committed by the hoster are of lower value to the coordinator and force the hoster to lose control over aspects of their resource management.

Future work will focus on the analysis of business models constructed using these value chains and SLA terms, operational management of ROIA and resources to address outstanding challenges such as dynamic ROIA-induced load, customer load imbalances, and their relationship to application adaptation and scalability features provided by a generic, abstract real-time application framework. These challenges are already faced in on-line gaming applications, but today the only solution is to restrict

customer interactions in the game environment. The `edutain@grid` approach offers the prospect of Grid-based ROIA with few restrictions, which should also stimulate much greater commercial investment in the Grid itself.

**Acknowledgments.** The work described in this paper is supported by the European Union through EC IST Project 034601 ‘`edutain@grid`’.

## References

1. Foster, I., Kesselman, C. (eds.): *The Grid2: Blueprint for a New Computing Infrastructure*, 2nd edn. Morgan Kaufmann Publishers Inc., Elsevier, Boston (2004)
2. Fahringer, T., Anthes, C., Arragon, A., Lipaj, A., Müller-Iden, J., Rawlings, C., Prodan, R.: *The Edutain@Grid Project*. In: Veit, D.J., Altmann, J. (eds.) *GECON 2007*. LNCS, vol. 4685, pp. 182–187. Springer, Heidelberg (2007)
3. The `edutain@grid` website, <http://www.edutaingrid.eu/index.php>
4. Foster, I., Kesselman, C.: *Globus: A Metacomputing Infrastructure Toolkit*. *International Journal Supercomputer Applications* 11(2), 115–128 (1997)
5. Czajkowski, K., Ferguson, D.F., Foster, I., Frey, J., Graham, S., Sedukhin, I., Snelling, D., Tuecke, S., Vambenepe, W.: *The WS-Resource Framework* (March 2004)
6. Breuer, D., Erwin, D., Mallmann, D., Menday, R., Romberg, M., Sander, V., Schuller, B., Wieder, P.: *Scientific Computing with UNICORE*. In: Wolf, D., Münster, G., Kremer, M. (eds.) *Proceedings of NIC Symposium 2004*. NIC Series, vol. 20, pp. 429–440. John von Neumann Institute for Computing, Jülich (2003)
7. IDC Case Study, *Butterfly.net: Powering Next-Generation Gaming with On-Demand Computing*, <http://www.ibm.com/grid/pdf/butterfly.pdf>
8. Big World Technology, [http://www.bigworldtech.com/index/index\\_en.php](http://www.bigworldtech.com/index/index_en.php)
9. Surridge, M., Taylor, S., De Roure, D., Zaluska, E.: *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, Los Alamitos (2005)
10. <http://www.gria.org>
11. Snelling, D., Fisher, M., Basermann, A.: *NextGRID Vision and Architecture White Paper*. This is updated periodically: at the time of writing the published version dates from 30 July 2006 (2006)
12. Mitchell, B., Mckee, P.: *SLAs A Key Commercial Tool*. In: Cunningham, P., Cunningham, M. (eds.) *Innovation and the Knowledge Economy: Issues, Applications, Case Studies*. IOS Press, Amsterdam (2005)
13. Phillips, S.C.: *GRIA SLA Service*. In: *Cracow Grid Workshop*, Cracow, Poland, pp. 15–18 (October 2006)
14. Müller, J., Gorlatch, S.: *Scaling Online Games on the Grid*. In: *Proceedings of the Fourth Annual International Conference in Computer Game Design and Technology*, 15-16 November (2006)
15. Glinka, F., Ploss, A., Gorlatch, S., Müller-Iden, J.: *High-Level Development of Multi-Server Online Games*. *International Journal of Computer Game Technology – Networking for Computer Games* (August 2008)
16. Müller-Iden, J., Gorlatch, S., Schröter, T.: *Parallelization and Scalability of Multiplayer Online Games via State Replication*. In: Grandinetti, L. (ed.) *High Performance Computing and Grids in Action*, March 2008, pp. 384–402. IOS Press, Amsterdam (2008)

17. Müller, J., Gorlatch, S.: Enhancing Online Computer Games for Grids. *Parallel Computing Technologies*. In: Malyshkin, V.E. (ed.) PaCT 2007. LNCS, vol. 4671, pp. 80–95. Springer, Heidelberg (2007)
18. Müller, J., Gorlatch, S.: Scaling Online Games on the Grid. In: *Proceedings of GDTW 2006 - Fourth International Game Design and Technology Workshop and Conference*, Liverpool, UK, pp. 6–10 (2006)
19. The Simdat project, [http://www.scai.fraunhofer.de/about\\_simdat.html](http://www.scai.fraunhofer.de/about_simdat.html)