

Integrating Mobile Devices into the Grid: Design Considerations and Evaluation

Stavros Isaiadis and Vladimir Getov

Harrow School of Computer Science
University of Westminster, London, U.K.
{S.Isaiadis, V.S.Getov}@westminster.ac.uk

Abstract

Mobile devices increasingly offer functionality beyond the one provided by traditional resources – processor, memory and applications. This includes, for example, integrated multimedia equipment, intelligent positioning systems, and different kinds of integrated or accessible sensors. For future generation Grids to be truly ubiquitous we must find ways to compensate for the limitations inherent in these devices and integrate them into the Grid, in order to leverage available resources and broaden the range of supplied services. The unreliability and limitations surrounding the mobile resources and services could significantly degrade the overall Grid availability and performance. In this paper, we propose the utilization of various mobile devices in the form of a single virtual wireless “cluster” that will hide the heterogeneity and dynamicity, mask the failures and quietly recover from them, provide centralized management and monitoring and allow for the federation of similar services or resources towards advanced functionality, quality of service, and enhanced performance. Finally, we present and analyze the results from the simulative performance evaluation of this design.

1. Introduction

In the last two decades we have experienced a transition towards mobility in the form of mobile devices, like cell phones initially, and hand-held computers and smartphones later. These trends are going to increase in the following years, resulting in a huge mobile computing community. An analogous transition in the field of communications and networking from the wired networks and fixed points of attachment to the flexible wireless networks has been developing rapidly. Furthermore, in addition to traditional computer resources like processors, memory, storage, and applications, new generation mobile devices provide support for integrated multimedia equipment,

This research work is carried out partly under the FP6 Network of Excellence CoreGRID funded by the European Commission (Contract IST-2002-004265).

intelligent positioning systems, and a diverse range of sensors. A possible integration of these mobile devices into the Grid could benefit both the mobile and the Grid communities. There is no doubt about the former, since mobile users will gain access to a huge number of resources, high performance facilities, specialized hardware and software and enormous resource pools. For the latter, it is an opportunity to utilize available resources in the mobile community and increase its performance and capacity and broaden the range of services and supplied functionality. And for future generation Grids to be truly ubiquitous we must have the option of integrating mobile devices into Grid systems (in the rest of this paper we'll be using the term "mobile devices" to refer to resource limited mobile devices with wireless connectivity like laptops, personal digital assistants, smart phones etc.).

Integrating mobile devices into the Grid could lead to instability and reduced overall performance, due to the inherent limitations of mobile devices and wireless communication links. Whenever a failure occurred we would have an avalanche of reactions including job rescheduling, data migration, resource re-allocation and activation of failure recovery mechanisms, and considering the increased rates of failures in such an environment, this is not acceptable. In an attempt to overcome these obstacles, we propose in this paper the clustering of all mobile devices that fall into the same subnet in order to present them as a single virtual system to the Grid. The main advantages of our approach are that we manage to hide the heterogeneity, dynamicity and complexity of mobile devices from the Grid, present a centralized point for managements and monitoring and federate similar resources or services to provide advanced operations and enhanced performance. Acting on behalf of the wireless cluster and exposing all available services to the Grid, there will be a set of dedicated proxies. We use this architecture as our vehicle towards our ultimate goal of developing and evaluating high service availability and failure resilience techniques in such hybrid Grid environments.

The rest of this paper is organized as follows. In the next section, we provide a brief overview of relevant research in this field. Section 3 gives details on the challenges we face in trying to integrate mobile devices to the Grid and the architecture we are adopting in an effort to tackle these problems. Section 4 describes the implementation details, while section 5 presents the simulative evaluation results and diagrams. Finally, section 6 concludes the paper and lists our future plans.

2. Relevant Work

A number of recent and current research projects and papers are dealing with mobility in Grids. In [2] the authors propose a clustered approach for integrating mobile devices into the Grid, without however, providing any implementation considerations or evaluation of their approach. In [10] the authors also propose a virtual cluster approach and a middleware to provide peer-to-peer operations but do not address resource virtualization or federation of similar resources or even larger scale collaboration. Also, none of the aforementioned papers address the issues of mobility and failure resilience and they do not take into consideration the widely used

OGSA/OGSI (Open Grid Services Architecture / Infrastructure) or the recently adopted OGSA/WSRF (Web Services Resource Framework) specification.

Mobile OGSI.NET [11] is an implementation of an OGSI based Grid container on the .NET hosting environment on mobile devices based on Microsoft's PocketPC. Mobile OGSI.NET allows for Grid service state saving and restoring and distribution of workload among devices with the same types of services, but with the cost of having to change existing services to adhere to the specific Mobile OGSI.NET programming model. Furthermore, Mobile OGSI.NET can only be realized on PocketPCs with the .NET framework installed.

AKOGRIMO [9] is a European funded project that has recently started and is dealing with mobility issues in the Grid. The purpose of the project is to evaluate the mobile Grid introducing the notion of mobile dynamic virtual organizations through applications that highlight the challenges present in such mobile environments, like e-health, e-learning and crisis management. AKOGRIMO favors research on ad-hoc mobile Grids and the integration of mobile IPv6 to support mobility in a Grid environment.

Finally, in [1] an agents approach is adopted to tackle device mobility, but the proposed architecture only allows mobile devices to be the consumers of services and not the providers.

The potential benefit of the integration of mobile devices into the Grid, but also the challenges that this raises, has been the theme in many other papers and research projects lately, but none of these provide any implementation methodology or propose an architecture to support this integration [3, 4, 5, 6, 7].

3. Hybrid Wired – Wireless Grid Environments

Initially the integration of mobile devices into the Grid seems highly unlikely due to inherent limitations: limited resources (CPU, secondary storage, available memory etc.); increased power consumption sensitivity; increased heterogeneity; unpredictable long periods of complete disconnectivity; unreliable, low-bandwidth and high latency communication links; and very dynamic network layout because of devices entering and leaving in a very unpredictable manner.

While directly connecting mobile devices to the Grid is very straightforward and requires no modifications to the existing infrastructure, it could significantly degrade the overall Grid performance because of the high rate of failures. Whenever a failure occurred, the Grid components would have to reschedule and reallocate resources for the active application, possibly migrating data around the Grid thus reducing the response time. Considering that in the mobile edge of the Grid, the failure rate is increased, this is not something we would like in busy, heavily loaded and complex Grid environments. In order to avoid potential performance penalties, we have decided to follow another route: group all mobile devices that fall into the same subnet and present them as one single virtual resource to the Grid. For this virtualization to be realized we need a set of dedicated proxies residing between the existing Grid infrastructure and the wireless cluster (it has to be stated here that by saying "cluster"

we do not mean a cluster in its traditional form – it is more like a grouping of devices that *may*, however, borrow some ideas from the cluster domain). This method will allow us to delegate some control to the proxies, getting some load off the higher level Grid components. The actual number of proxies will be determined by early experiments, and it may be possible to distribute the proxy functionality across a number of workstations conveniently located behind the distribution network of the “cluster”. Both the direct and the clustered approach are depicted in Figure 1.

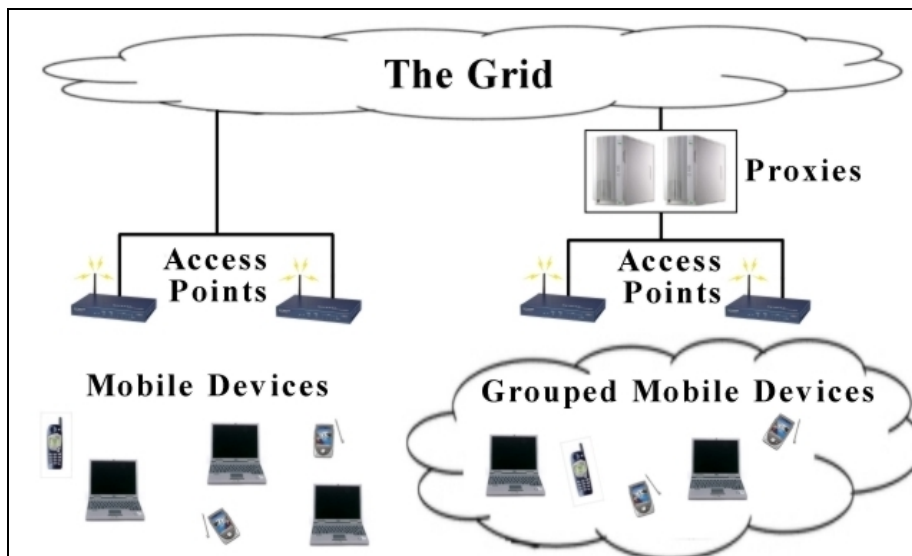


Fig. 1. Direct vs. Clustered connection to the Grid

By following this approach, we can encapsulate the heterogeneity and the dynamicity present in the cluster – thus imposing minimal disruption to the regular Grid operations, and leverage the resource repositories in the wireless cluster. Furthermore, we can mask all internal failures and recover from them locally without any notification of the higher-level Grid components. We deal with failures internally, re-allocating resources from within the cluster itself, locally rescheduling and migrating data if necessary.

4. Implementation Prototypes

In this architecture we are adopting a service-oriented approach currently conforming to the OGSA/OGSI specification and using the Globus Toolkit 3 (GT3) [15] implementation. In the near future we plan to migrate to the newest WSRF specification [8] and take advantage of the latest Web Services Description Language [14] (WSDL v1.2 or v2) that will provide a lot of enhancements for Grid service development.

All resources and specific functionality in the cluster are accessed through Grid service interfaces, which can fall into one of these two categories (depending on the interface type and description):

- Services that provide specific functionality, like for example access to a proprietary database, access to a sensor or a vector computer, access to other specialized hardware or software. Similar services –i.e. that offer the same functionality or implement the same interface, will be aggregated and presented through a single service interface deployed at the proxy. All requests arriving at the proxy for a specific type of service will be forwarded to an available node that supports it. This technique will give us more flexibility and will allow us to implement collective operations on all nodes offering this functionality, parallel execution whenever this is feasible and supported by the application, or even mirrored execution for higher reliability.
- Services that provide controlled access to raw resources like processor cycles, memory, secondary storage etc. Again resources are aggregated and presented through a single interface, but the aggregator service implementation is now different. Now we only publish a fraction (70-80%) of the total amount of aggregated resources, in order to expose a virtually more stable environment: there are spare resources available in the cluster to cover up any possible node failures and avoid unnecessary disturbance of the Grid by reallocating resources from within the cluster itself. Failures are now masked and dealt with locally by the “cluster” community scheduling and recovery components, thus avoiding putting more load on the higher level Grid scheduling systems, monitoring and failure recovery mechanisms.

We are implementing early prototypes of the proxy engine components, mainly the service/resource aggregation and indexing components. These early prototypes are somewhat restrictive at this stage, and we haven't included functionality that will be present at the final version –like support for mobility, failure resilience and more. The purpose of this prototyping is to give us a better insight of the actual needs and challenges that we will have to face on our way to the final deliverable engine.

After the registration and service aggregation procedure, the engine indexes the services based on static service information (like functionality offered, hosting device's characteristics, available resources and other) and keeps a registry that closely cooperates with the community scheduler and the information and monitoring services to allow for dynamic resource allocation and reservation. It is important to mention that advanced resource reservation can neither be supported nor guaranteed due to the dynamic nature of the cluster. A lower level description of the whole service aggregation procedure follows and is also depicted simplified in Figure 2.

1. The mobile node that wishes to expose a service sends the WSDL and an XML file containing service and device information to the engine by using a convenient web interface. The XML file will be used for indexing purposes by the Cluster Service Registry.
2. The WSDL file is parsed using JAXP (Java API for XML Processing) in order to extract all necessary service information, and is modified so that we can then deploy it at the proxy.

3. Both the original and the modified WSDL files will be fed to a generator that will automatically create a Java implementation for the aggregator service, the deployment descriptors and all the necessary helper Java classes like the Server-side skeleton, the client-side proxy, the Service Locator and so on.
4. Next, all these files are collected and put into a Grid Archive File, which is then deployed to the Grid hosting environment using Apache Ant tools.
5. Finally, the newly deployed service at the Proxy can be published to a UDDI registry or GT's MDS (Monitoring and Discovery Service) or any other index service.

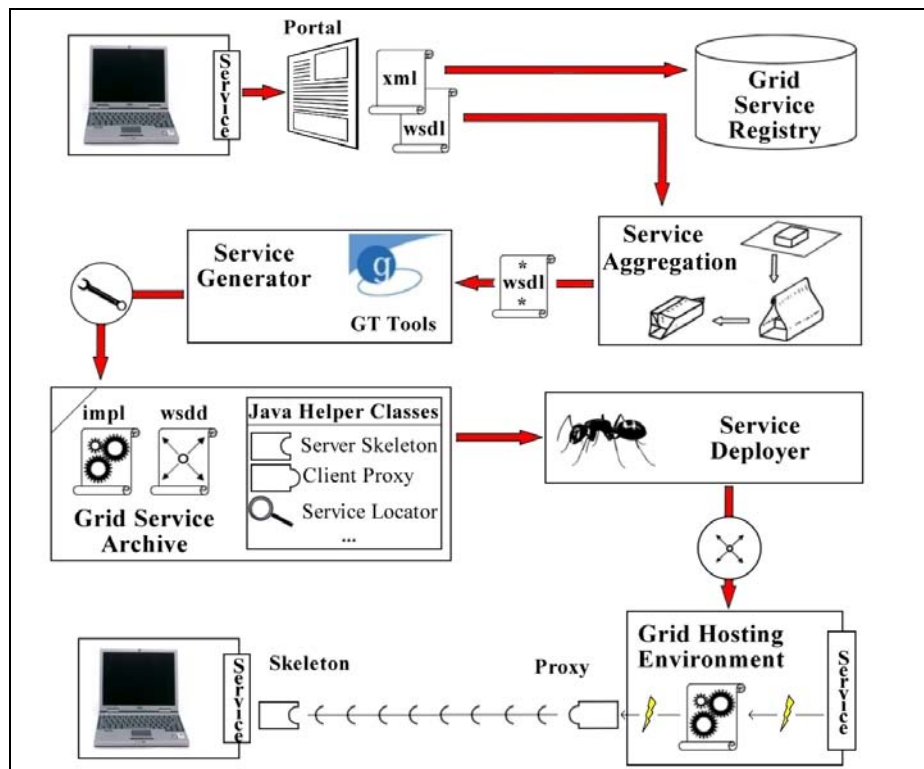


Fig. 2. The service aggregation and service generation procedure

As is easily observed from the description, both the mobile user and the cluster administrator have minimal work to do in order for the aggregation and registration to occur and for the service to be deployed and published. The mobile user merely has to fill in a small form in the web interface and submit it, while the administrator only has to deploy the service. In the near future, we plan to fully automate this procedure by making use of a dynamic resource discovery infrastructure like Jini [13].

The resulting aggregator service deployed at the proxy provides a single interface to all aggregated resources in the “cluster”. This abstraction makes job submission easier for inexperienced users and simplifies the job of Grid scheduling systems, since they can now delegate finer control to the proxy engine which will take care of distributing the tasks or forwarding service requests to the mobile nodes in the “cluster”.

When a request for a service arrives at the proxy, an available mobile node that match the requirements is found by querying the Cluster Information and Monitoring Service and the Cluster Service Registry, a new service instance is created and all subsequent method calls are forwarded to that node. If no nodes can offer the requested functionality a “service unavailable” response is forwarded. Our engine also takes care of state and lifetime management and any possible notification subscriptions automatically.

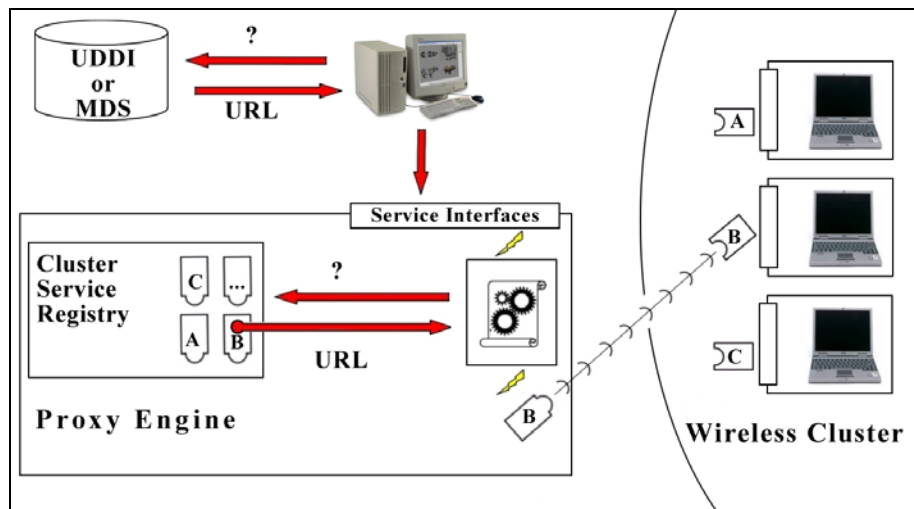


Fig. 3. Servicing requests.

5. Simulative Network Performance Evaluation

In order to ensure that this architecture will not present a bottleneck and won't cause network performance degradation, we have performed several simulations using OPNet®. We have modeled a very heavily loaded data-centric Grid environment following regular Grid behavior, i.e. light traffic initially querying service registries, followed by point-to-point data transfer, request for CPU usage, database queries and updates etc. A brief description of the simulation environment is given below, followed by some of the most important results that were generated.

- 4 Basic Service Sets each one servicing 10 mobile devices, 50% of which contribute resources to the Grid community. These sets represent the mobile resource limited edge of the Grid.
- A LAN consisting of 50 nodes all of which contribute resources to the community. The LAN acts as the main Grid infrastructure into which we want to integrate mobile devices.
- A couple of proxy servers residing in between the 4 WLANs and the LAN, acting as the interface point between the mobile “cluster” and the Grid.
- All Wi-Fi interfaces operate at a rate of 11Mb/s.

- All Ethernet interfaces operate at a rate of 10Gb/s.

During the 60 minutes of the simulation, all nodes (both wireless and wired) request services that are available either in the WLANs or the LAN. When a Grid node wants to access a service available in the cluster it sends the request to the relevant proxy aggregator interface (since the cluster is hidden and access to its services is provided through the proxies). The proxy will then forward the request to an available node along with the necessary data or code to execute. In this environment we assumed that no failures occurred as the main goal of the simulation is to measure the network load and the proxy response time, not the actual engine functions or performance. We were mainly interested in the wireless media statistics and the proxy utilization and response time, in order to ensure that no performance or network traffic bottlenecks occurred because of the narrow proxy design – “narrow” because all traffic between the Grid and the “cluster” goes through the proxies.

- Wireless Media

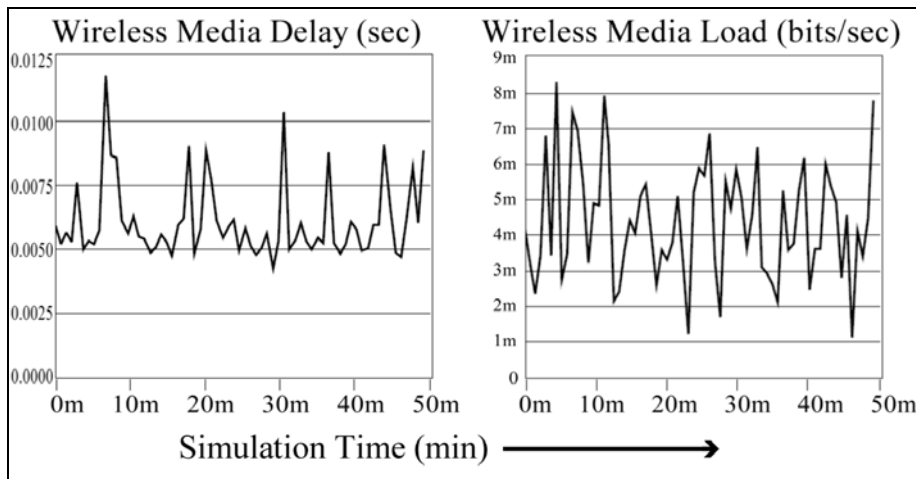


Fig. 4. Wireless Media Delay and Load

We can see from the second graph that at no point during the simulation there was any data dropped due to network traffic as the wireless media never reached its full potential (averaged at around 5Mbits/sec \cong 45%) and the media access delay was kept to very low levels at around 7msecs. Our architecture, doesn't introduce any network performance bottlenecks even though the simulation environment was very heavily loaded.

- Proxy Performance

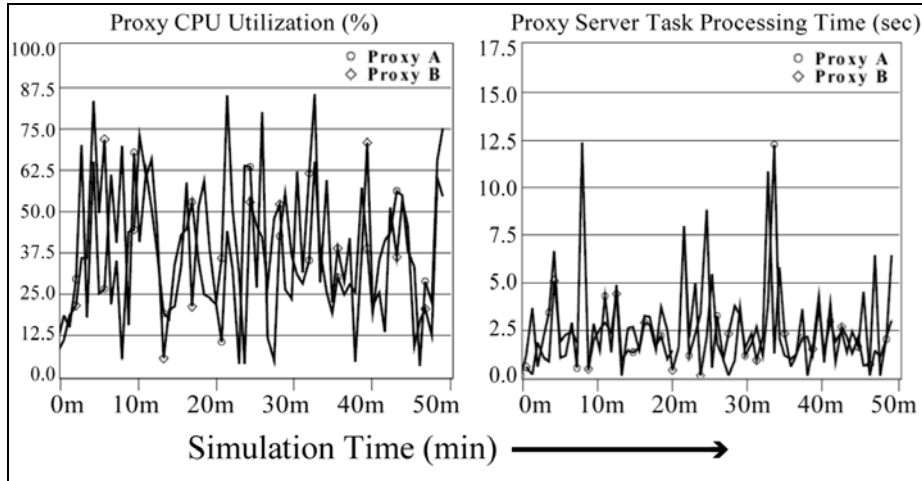


Fig. 5. Proxy CPU Utilization and Server Processing Time

The results showed that the two proxies handled very efficiently the heavy traffic we injected to the network (utilization was kept around 50% following a bursty nature) and the server performance was kept to very good levels despite the big number of service requests arriving from both the WLANs and the LAN.

6. Conclusions and Future Work

So far we have implemented the components responsible for service aggregation and indexing, and those dealing with servicing requests. Everything is achieved with minimal intervention from the mobile user and the cluster administrator. This is very important since we do not want to discourage mobile users from contributing to the Grid community by requesting lengthy and complicated procedures. Also, a cluster membership scheme – if one is deemed necessary – will be built on top of already installed Grid security infrastructure making use of available certificates so that no extra inconvenience is caused and alignment with existing authentication and authorization mechanisms is achieved.

We have tested our engine's aggregation and indexing components against a wide variety of services and it has been proved to perform efficiently. Nevertheless, there is still plenty of work to do and we have to implement failure prediction, detection and recovery components that will take us one step closer to our ultimate goal of autonomy and high service availability in this wireless environment.

The simulation reports we got were more than encouraging and showed that even for a very heavily loaded hybrid environment our proxies responded very well and there were no performance or network traffic bottlenecks introduced due to the “thin”

proxy layer architecture. Near future plans include experimental evaluation of our design and addressing mobility and roaming issues between two wireless “clusters”.

As we are dealing with resource limited devices, we need to develop a lightweight Grid platform suitable for such devices. The authors also participate in ongoing research regarding a generic components based Grid platform and hence we are confident that we will have results in this field in the near future [12].

Finally, we need to investigate available community schedulers suitable for wireless and very dynamic environments in order to select and integrate one in our proxy engine. The scheduler will cooperate closely with the wireless “cluster” service registry and the monitoring and information components of the engine in order to deal with dynamic resource allocation and load distribution whenever this is feasible.

References

1. L. Cheng, A. Wanchoo, I. Marsic, “*Hybrid Cluster Computing with Mobile Objects*”, Proc. of Fourth International Conference on High-Performance Computing in the Asia-Pacific Region, 2000.
2. T. Phan, L. Huang, C. Dulac, “*Challenge: Integrating Mobile Wireless Devices into the Computational Grid*”, ACM MOBICOM, 2002
3. G. H. Forman, J. Zahorjan, “*The Challenges of Mobile Computing*”, IEEE Computing Milieux, 1994
4. Chlamtac, J. Redi, “*Mobile Computing: Challenges and Potential*”, Encyclopedia of Computer Science, 4th edition, 1998
5. M. Franz, “*A Fresh Look at Low Power Mobile Computing*”, <http://research.ac.upc.es/pact01/colp/paper15.pdf>
6. B. Chen, C. H. Chang, “*Building Low Power Wireless Grids*”, http://www.ee.tufts.edu/~brchen/pub/LowPower_WirelessGrids_1201.pdf
7. D. Bruneo, M. Scarpa, A. Zaia, A. Puliafito, “*Communication Paradigms for Mobile Grid Users*”, IEEE/ACM International Symposium on Cluster Computing and the Grid, 2003
8. K. Czajkowski, D. Ferguson, I. Foster, J. Frey, S. Graham, T. Maguire, D. Snelling, S. Tuecke, “*From Open Grid Services Infrastructure to WS-Resource Framework: Refactoring & Evolution*”, http://www.ibm.com/developerworks/library/ws-resource/ogsi_to_wsrf_1.0.pdf, 2004
9. The **AKOGRIMO** project: <http://www.akogrimo.org>
10. J. Hwang, P. Aravamudham, “*Proxy-based Middleware Services for Peer-to-Peer Computing in Virtually Clustered Wireless Grid Networks*”, in Proceedings of International Conference on Computer, Communication and Control Technologies, 2003
11. D. Chu, M. Humphrey, “*Mobile OGSINET: Grid Computing on Mobile Devices*”, <http://www.cs.virginia.edu/~humphrey/papers/MobileOGSI.pdf>, 2004
12. J. Thiyagalingam, S. Isaiadis, V. Getov, “*Towards Building a Generic Grid Services Platform: A Component-Oriented Approach*”, in V. Getov and T. Kielmann (Eds), “*Component Models and Systems for Grid Applications*”, 39-56, Springer, 2005
13. J. Waldo, K. Arnold, “*The Jini specifications*”, Jini technology series. Addison-Wesley, Reading, MA, USA, Second edition, 2001
14. R. Chinnici, M. Gudgin, J. Moreau, S. Weerawarana, “*WebServices Description Language (WSDL) 1.2*”, World Wide Web Consortium. <http://www.w3.org/TR/wsdl12/>, 2003
15. The **Globus** Project, <http://www.globus.org>