IST-2001-32133
GridLab - A Grid Application Toolkit and Testbed

Small-bandwidth optimized adaptive grid services

Author(s):	Piotr Grabowski, Bartosz Lewandowski
Document Filename:	GridLab-12-D.3-0001.tex
Work package:	Work Package 12 - Access for Mobile Users
Partner(s):	ZIB, Poznan Supercomputing and Networking Center
Lead Partner:	ZIB
Config ID:	GridLab-12-D.3-0001-1.0
Document classification:	GridLab Deliverable, Prototype, Internal

Abstract: In this document we present our approach to giving users the opportunity to access grid resources, developing appropriate adaptive grid middleware optimized for small-bandwidth networks used by mobile users devices.

Last amendment date: 2003/12/31 & time: 13:17:56

Contents

1 Introduction to small-bandwidth optimized adaptive grid services and short document description

In the first months of the GridLab project existence, the WP12 people were focused on the Grid User needs. After several discussions it was clear that users would like to have access to their grid applications even when they are not on-line via the standard network. The users are also interested in notifications: they want to know if something has happened with their simulation/application even when they are on their way or at home. They would like to be notified in different ways: using Email, Fax or SMS that could be sent to their cell phones. Although the notification infrastructure takes into consideration mobile networks disadvantages (e.g. small bandwidth, intermittence) the ready-to-use solutions for grids do not allow us to deliver the demanded grid functionality to the user. Analyzing the user needs and the constraints of the mobile devices and mobile networks, we have decided to use client - server architecture (with the main intelligence on the server side to avoid the weaknesses of mobile devices), and optimize the amount of data sent through the network. We developed web/grid services that can store application messages and send them as notifications to the user, we are also developing the service that can allow the mobile user access to the grid. This service is working as a Grid Gateway for Mobiles - the client request is processed in the service and then forwarded to the rest of the grid - the response can be translated for the Mobile Device in the gateway or prepared in Services according to the Mobile Device constraints the gateway is aware of. This document describes in detail the mobile services we are developing, the architecture, the technologies and our contribution in developing small bandwidth, optimized adaptive grid services for mobile users.

2 General system architecture

During internal meetings it was agreed that we need one common solution that would suit our users needs (and allow us not to rewrite a given functionality for different platforms). Due to mobile devices constraints we also agreed to use the client-server architecture (with the main 'intelligence' on the server side). On the server side we could choose from different technologies; however, the final solution had to be easily accessible from other services and from the mobile client side, too (for client we finally decided to use Java 2 Platform Micro Edition Mobile Information Device Profile - J2ME MIDP) which is widely used in mobile devices(such as cell phones).

2.1 Server side technologies

We would like to use one of the standard technologies that could be found nowadays. Taking into consideration the fact that we are going to write codes for Java 2 Platform, we want to use the existing mechanism for it. The first technology is standard Servlet/JSP Pages (with Jakarta Tomcat as a container), the second one is Portlet technology and the third is the Web Service.

2.1.1 Servlets

Using this solution we had to write a dedicated Servlet/JSP framework supporting XSL transformations (XML-> WML, XML->HTML, XML->XML; the output from the transformation is sent to a client device). This was our choice for the beginning, just because the Portlet framework was not ready yet and we needed a temporary solution for mobile side tests.

2.1.2 Portlets

The Portlet framework is being made by the GridLab Portal Workpackage (WP4). With this solution we can use XSL transformations for serving content to the mobile side, too. This is our choice for the final server side application - it will allow us to avoid the redundancy of building two separate portals (a common one and a dedicated one). The final solution uses GridSphere (a portlet framework) as the container for the mobile portlet/servlet which can communicate with the GridSphere services, web/grid services and serve mobile client requests.

2.1.3 Specialized Portlet/Servlet in GridLab: Mobile Command Center

To avoid the redundancy in writing codes, we developed a specialized server for mobile clients. It is called the Mobile Command Center (MCC). The MCC is written as a servlet which has to be deployed on the same container as the existing GridSphere instance. This structure is forced by the special needs of mobile clients from one side and the necessity of calling the GridSphere services such as the login service (this allows us to use the same user and user profile repository as used by the GridSphere, which gives the mobile-user opportunity to e.g. manage his/her profile via different interfaces). The current functionality of the MCC allows the mobile user to browse the user folders, messages and mobile visualizations. In the next version we are focusing on giving the user grid functionality like starting, migrating or ending/suspending applications or simulations. All of the communication with the service uses the HTTP(S) protocol, a minimized set of commands and own data structures, which ensure the minimal amount of data sent via GSM/GPRS connection (which is paid for data amount - the expenses are lower in this case).

2.1.4 Web Services

The GridLab project as a whole decided to make an extensive use of a new emerging technology: Web Services (WS). We were surprised by the fact that Web Services can be accessed from a mobile phone with a satisfactory connection and parsing speed. The question in this case was only if we were going to use WS or not. Now we can say that it is possible to benefit from Web Services, writing software for using mobile devices with MIDP, and we are going to do it for devices with stronger processors and larger memory without package size limitations. All services without the presentation layer which do not require Portal/Portlets can be implemented and served using Web Services. However, the developer should be aware of disadvantages of Web Services for the mobile appliances - the amount of metadata increases and the needed processor power is bigger. This is causing a larger amount of data sent via network (in comparison to pure XML or plain Text/Data) and slower data decoding on the Mobile client side.

2.1.5 Specialized Web Services in GridLab: Visualization Server

The WP8 GridLab workpackage (Data Handling and Visualization) is developing the Visualization Server which can produce large visualizations from data gathered in the grid. Regarding the sizes of result visualizations, needed processor power to decode the images or frames and variety of visualization formats, the visualization produced by visualization server are not useful for mobile clients (however, the mobile users are still expecting that the visualization can be viewed on their mobile device). This fact convinces us to propose a specialized service that could serve visualizations in form, size and format that are more appropriate for mobile devices capabilities. The visualization service for mobiles can produce custom images from original visualizations. It can be done with the desired color depth and picture resolution for the whole image or only the desired part of it.

2.2 Client side technologies

There are plenty of technologies that are emerging in the mobile world nowadays. However, many of them work only on one or several hardware platforms - we have decided to choose between two that are widely distributed and seem to be the leading software platforms of the mobile future in the next years: WML browsers(using WAP) and JAVA 2 Platform Micro Edition. JAVA J2ME enabled devices (communication via HTTP/HTTPS/SOAP with server side) gives us better offline work, data can be stored in small internal repositories (the size of this repository is device dependent and is rapidly growing, it ranges from several kilobytes in old devices to megabytes in the newest ones). Targeted devices for this solution are mobile phones and PDAs.

2.2.1 JAVA 2 Platform Micro Edition Client together with Mobile Command Center

The mobile client written in JAVA 2 Platform Micro Edition can communicate directly with web services if the device is powerful enough. However, for mobile devices with some constraints we are implementing the Servlet Gateway for Mobiles - the Mobile Command Center Server. Due to small processor power, memory shortages or provider's policy (e.g. Nokia), there is a certain number of devices that can not use Web Services to connect to the server - they are too slow or do not give a sufficient amount of memory for Web Services packages like kSOAP (additional constraints in Midlet (MIDP application) size). For such devices we are going to implement a gateway that will be accessed from a mobile device with HTTP/HTTPS and then information is 'forwarded' via Web Services to the next service. Support for HTTP is an essential part of MIDP compliant devices so there is always an open way to connect to the service. With the MCC we can redirect the request from a mobile device to next services. In this way we can allow user to access the grid services, for example it can be the request for migrating an application to another host or suspending it.

2.2.2 JAVA 2 Platform Micro Edition Client together with MCC and Visualization Server

Another service we are implementing for mobile devices is the Visualization service for mobiles. The visualization request sent from a mobile device can reach the Visualization service in two ways. The first one is direct and implies the usage of the Visualization Server web services interface from a mobile device with enough power to use the web services. The second one is indirect and uses the Mobile Command Center as a gateway. In the latter case the mobile request for visualization is sent to MCC, there the device capabilities are decoded from the request and together with the user's wish regarding cropping is sent via web service to the visualization server. The desired output image is put by the Visualization server in a place that is reachable from the mobile device and information about placement is sent as a response to the caller (directly to a mobile device or via MCC if it was used).

2.2.3 JAVA 2 Platform Micro Edition Client together with MCC and Message Box Web Service

The first service developed by WP12 was the Message Box. This one based on relational database storage of user messages was developed as Java API first and then the web services interface was added. The external application or portal can store messages for users in the Message Box folder structured repository. The messages can also be sent to the user in a desired way (SMS, EMail, Fax - depending on a user profile, parameters and implementation of sending modules).

The stored data can be browsed and managed via Java API and Web Services. To access the Message Box service from a mobile device we have to either call directly web service functions or use the Mobile Command Center as gateway. Both ways are possible; however, the second one gives the developer chance for providing the mobile application for a larger set of mobile devices due to their constraints.

2.2.4 Web Service Clients in GridLab

There are plenty of services in GridLab and WP12 is going to use some of them (e.g. we are going to provide information about messages into WP8 MetaData, check WP6 Authorization Server for security information or steer WP8 Visualization Service for scaled down Visualization for Mobile Devices). The clients for our services will be the GridLab Applications that want to store messages or send notifications; these can be Portals or Cactus thorns or user applications.

2.3 Optional packages and 3-rd parties solutions

These are described in our previous deliverable (see "Mobile-enabled grid middleware and/or grid gateways").

3 Development

The development in WP12 can be divided into 3 branches. The first branch are services for other applications (e.g. notification, message box services), the second branch are specialized services for mobiles (Mobile Command Center and Visualization for Mobiles) and the last one is to develop an efficient and common solution on the mobile client side.

3.1 Mobile Command Center Service Introduction

The MCC first version was HTTPMessageBoxServer (see deliverable D12.1:"Mobile-enabled information exchange"). The service was very customized to handle only mobile requests to the Message Box API. It was implemented in a form of a servlet that used the ASC portal user and user profiles repositories. The introduction of GridSphere 1.0 allows us to move into a new framework and use its services. Initial problems with specific presentation requirements for a mobile are solved and now the MCC is a separate servlet that has access to the GridSphere services.

3.2 Message Box Web Service Introduction

The usage scenario (see D12.2:"Mobile-enabled grid middleware and/or grid gateways") requires that the WP12 developers should provide a new service that could send and store different kinds of messages. To fulfill the scenarios requirements we developed 3 applications: notification server - Message Box, Mobile Client - Message Reader and Dedicated server for Message Box - Message Box Server (HTTP/HTTPS). The Message Box application was designed as Java API that can be used from an external package like Portlets or dedicated servlet server. The Web Service extension for Message Box API was added in the recent months. The Message Box database was improved to contain also the user and user profile data. This allowed us to have unique identifiers of users regardless of the client that uses our service which is now a stand alone service and does not need to be deployed in the ASC portal environment. An example Message Box Web Service Client was also developed, it tests the connection to Message Box server and tries almost all functions of service interface. This can be used by developers to test if the Client Web Service environment is set properly and connection works before they actually write their own codes that are using the service.

3.3 Visualization Server Web Service Introduction

Working together with the WP 8 Data Management and Visualization, WP12 Access for Mobile Users work package can deliver a specialized service accessible from mobile user devices. WP12 users expressed the need of displaying scaled down visualizations (mainly pictures) on their mobile phones or PDAs many times. Although the large resolution images in many different formats can be easily accessed and displayed on desktop computers with high bandwidth networks, they are completely unusable working with a mobile device. Small screens with low resolutions and colors depth together with intermittent, small bandwidth network access in case of mobile devices implies the need of developing software infrastructure that could scale down 'heavy' visualizations into more acceptable for small devices formats. The subject of this subproject perfectly meets the WP12 deliverables plan especially with the "Small-bandwidth optimized adaptive grid services" and "Mobile-enabled grid middleware and/or grid gateways". In order to fulfill the aforementioned expectations and taking into consideration the fact that client - server architecture suits best the mobile devices needs (due to the limitations of memory, processor power, etc), and additionally chosen in the GridLab model of Grid/Web Services, it seems that

the best solution will be to implement the new Visualization service as a standard GridLab grid service. The model of communication for this service can be found in the next picture. The main idea of a new service is getting new versions of visualization appropriate to the device constraints and features. It is essential that the Visualization Service could change the file format of visualization - in case of mobile devices we are not interested in large dimension not compressed files. The most suitable format would be PNG, therefore the service has to have a possibility to recode images into PNG files using dimensions of mobile device screen and colors as a base for new image dimensions. Taking into consideration all the above, a new method for mobile clients was added to the existing service interface. The parameters of this method are: source and output visualization placements, desired color depth, output picture size and cropping parameters. This service method can be called by any web service client and the output image can be accessed via HTTP(S) from any place and device. The call parameters should be set by user/developer according to device capabilities. This can be done automatically by the MCC if the request for visualization is sent indirectly (except cropping properties which have to be set by the user - see Mobile Client Introduction).

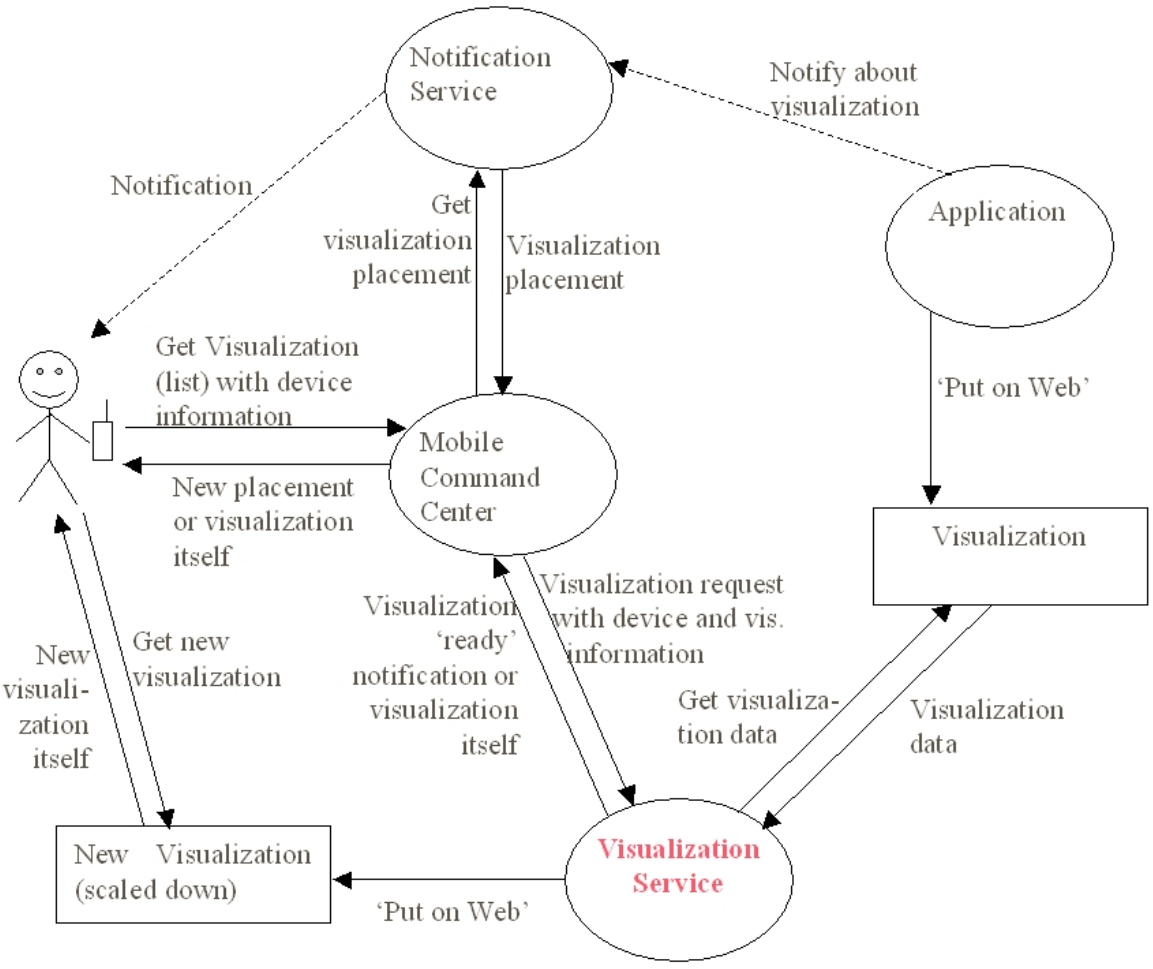

Figure 1: Diagram of interactions between entities with Visualization Server involved.

3.4 GridLab Mobile Client Introduction

The Message Reader application (Java J2ME Midlet) which was prepared for the first two deliverables has to be extended with capabilities to steer applications in the grid and view and manage the visualization images. The standard functionality of the Message Reader (browsing the Message Box for user folders and messages) will be moved to the new GridLab Mobile Client without changes to avoid developer work overhead. The new Client will allow the user to start simple applications, to migrate applications and to suspend them. The visualization browser part will allow to view images from URL given by the user and URLs stored in messages obtained from the Message Box for the user. The user can view the picture as a whole or display the zoomed part of the main picture. The navigation between the parts of the main image will be done with a movable frame whose size can be changed by the user. The ratio of the frame should be constant and set to the device screen ratio. The first pictures of the new client prototype can be viewed in the next figure.

Figure 2: GridLab Mobile Client prototype pictures.

3.5 Software collaboration

After the Super Computing 2002 some users have expressed their need for using the Message Box API from other GridLab modules than Cactus ASC Portal. Looking at the overall development progress in GridLab and taking into consideration technologies chosen by the GridLab Technical Board, we have decided to use Web Services as the leading technology for our client-server architecture. There are two main services developed by the WP12 team: the first one is the Message Box Web Service which will store, send and retrieve different kinds of messages; the second one is the Mobile Command Center Service which will serve as a Mobile Gateway to other Grid Services (which includes simulation steering and visualization requests to the Visualization Service). The Mobile Command Center will serve requests sent from the Mobile Side via HTTP/HTTPS for the mobile devices processor/memory weakness reason. This solution allows us not to require possession of the strongest devices on the market from the user. In the next figure one can find how our services work together, and see the communication schema between all affected entities.

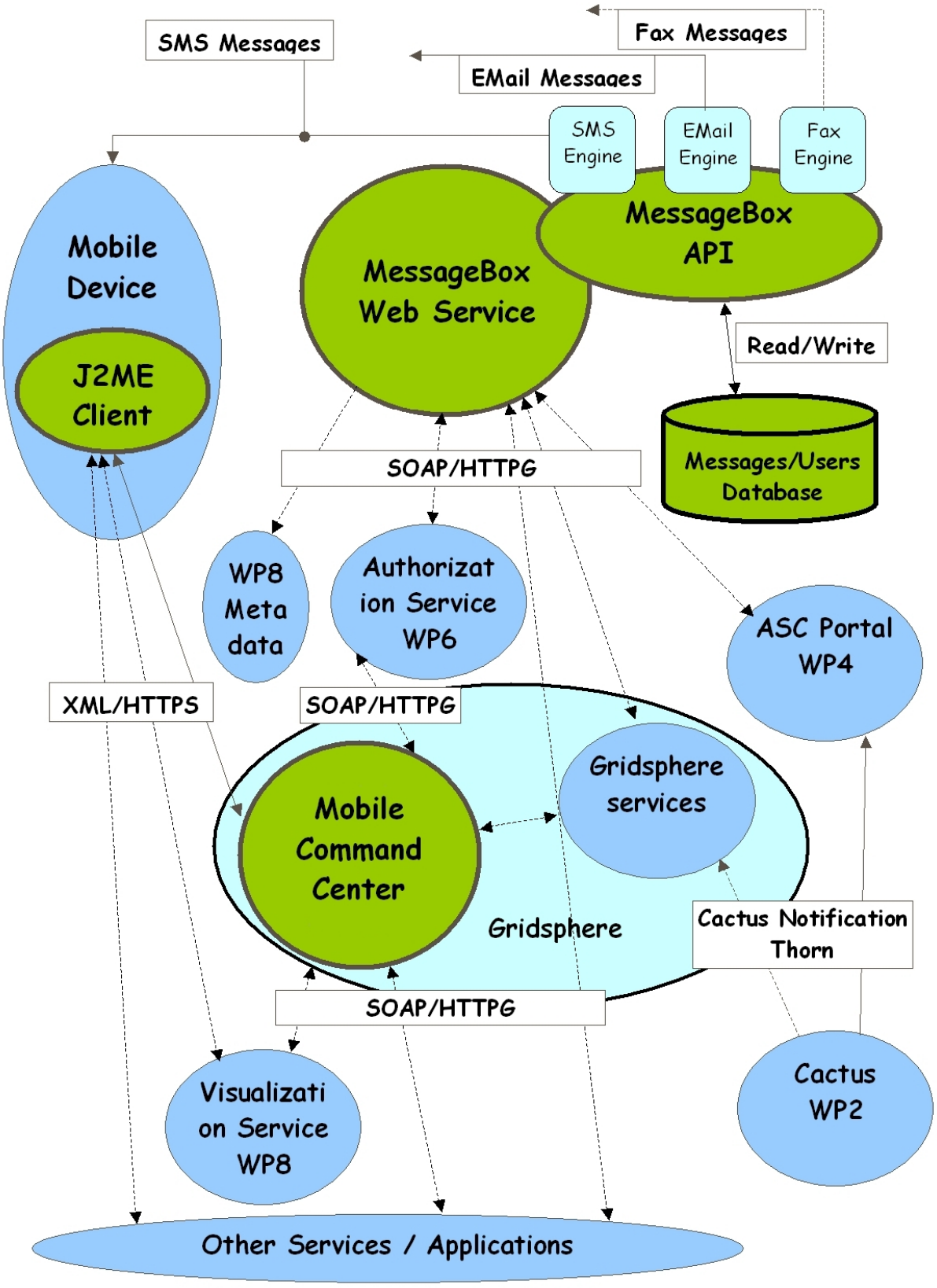

Figure 3: Diagram of Mobile, Web & GridSphere Services interactions between entities.

3.5.1 Current state - development

We developed and extended the Message Box Java API and the Message Box Web Service and deployed it on rage1.man.poznan.pl machine. Currently we are enhancing the Mobile Command Center and deploying it together with GridSphere on rage1.man.poznan.pl. In the same time we are implementing the Mobile Client to test the implemented functionality in MCC (e.g. browsing images). The next step will be establishing a connection to the WP6 GAS server from all our services to improve security and also storing information about the Message Box entities in the WP8 Metadata Storage Box.

References

[1] Krzysztof Kurowski, Jaroslaw Nabrzyski, Bogdan Ludwiczak. *Document Publication Rules*. GridLab 2002.

[2] Eric M. Burke *Java and XSLT*. O'Reilly 2001

[3] http://java.sun.com/j2me/ *JavaTM 2 Platform, Micro Edition (J2METM)*. Sun Microsystems, Inc..

[4] Vipul Gupta and Sumit Gupta *KSSL: Experiments in Wireless Internet Security*. http://playground.sun.com/ vgupta/KSSL/SMLI-TR-2001-103.pdf

[5] Apache Software Foundation *The Apache Jakarta Project*. 1999-2002, Apache Software Foundation http://jakarta.apache.org/tomcat/

[6] Apache Software Foundation *The Apache AXIS Project*. 1999-2003, Apache Software Foundation http://ws.apache.org/axis/

[7] Apache Software Foundation *Log4j Project*. 1999-2002, Apache Software Foundation http://jakarta.apache.org/log4j/

[8] Borland Software Corporation *JBuilder(R)*. http://www.borland.com/jbuilder/