

# Grid-based Interactive Virtual Scientific Experiments for Distributed Virtual Communities

Ioannis T. CHRISTOU, Thanassis TIROPANIS, Sofia TSEKERIDOU

*Athens Information Technology  
19Km Markopoulou Ave. P.O. Box 68  
Peania 19002, Greece  
{chr, ttir, sots}@ait.edu.gr  
fax: +30-210-668-2703*

Konstantinos ROUSSOS

*INFOTE S.A.  
Mihalakopoulou 3, Athens Greece  
[konrous@gmail.com](mailto:konrous@gmail.com)*

## Abstract

E-learning technologies have matured to a point where distance learning classes are commonly offered from many leading Universities around the world. A major challenge in such distributed classrooms is the formation of virtual communities among the participating students, enhancing the overall learning experience. *Shared* virtual laboratories offer the possibility of forming such virtual communities as students form lab teams to run the same interactive simulation and in the course of such experiments learn to interact and understand each other better. We have designed and implemented a Virtual Scientific Experiment architectural framework on top of a Grid infrastructure for running interactive virtual laboratory experiments for such distributed student communities with visualization capabilities. The architecture is based on Web Services standard protocols such as WSDL and WS-Notification as implemented in the WSRF specification. For the first concrete instantiation of this architecture, we ported a stand-alone Wireless Sensor Network simulator written in Java in our Grid-based architecture and extended it to allow for initial collaborative parameter setup and on-the-fly visualization of the simulation execution and interaction with it, a capability not present in the original simulator. We report on results from running such simulations on a local Grid infrastructure. System evaluation results from a distributed pool of students show the added value of our system in enhancing distance-learning programs and Virtual Classes with extensible collaborative and interactive Virtual Laboratories sessions.

**Keywords:** Virtual Scientific Experiments, Virtual Communities, GRID-based Shared Virtual Laboratories, Distance Learning, Collaborative Learning, Distributed Computing, Network Simulations, Web Services.

## 1. Introduction

The emergence of broadband communications as a commodity combined with advances in computer hardware and specialized video-conferencing software has allowed for a significant growth in life-long learning, distance learning and e-learning technologies, products and services in general. As a result, some of the top Universities in the world are offering degree-granting programs with geographically dispersed campuses (virtual campuses) around the globe. Students from all these campuses form one virtual class and community to attend the lectures of the institutions' instructors in real-time. The MSIN program ([www.ini.cmu.edu](http://www.ini.cmu.edu)) is such a graduate program in Information Networking offered jointly by the Carnegie Mellon University in Pittsburgh, PA, USA and the Athens Information Technology ([www.ait.edu.gr](http://www.ait.edu.gr)) operating as the CMU campus in Athens, Greece. Several courses are offered as distance-learning courses through e-learning technology. Such courses are taught in auditoriums equipped with large high definition video-walls and cameras' transmitting between the auditoriums what is happening in the other class in real-time. As a result, instructors and students at both auditoriums participate in an essentially unified but distributed virtual classroom where they enjoy full interactivity with every participant in the auditoriums at all times. Real-time audio-visual interaction is achieved through several dedicated high-speed communications links between the two campuses. The educational process is complemented with the use of an appropriately configured document repository where course related documents are placed and discussion forums for course related subjects are supported. Several courses expect students to do their coursework in teams in a laboratory environment. These teams are formed locally, meaning that students at the A.I.T. campus form their own teams and students at the CMU campus form their own teams but teams across the two campuses can not be easily formed because of lack of support of a distributed laboratory infrastructure that would enable closer student interaction and collaboration (Christou et. al. 2007).

Interactive virtual laboratory experiments run together by teams of students unrestricted by geographic location would enhance the formation of community ties between all members of the same class independent of their location. For these reasons, within the context of the ELeGI project (European Learning GRID Infrastructure) we have designed and implemented a prototype architecture for running such virtual scientific experiments, and demonstrated its power by plugging-in the SENSASIM simulation engine (Rayess et. al. 2006), a Wireless Sensor Network simulator written in Java that has been used locally in Athens Information Technology for a graduate course on Wireless Sensor Networks that is offered in the MSIN program. The tool was appropriately modified so as to allow users to alter the execution of the simulator in real time.

The deployed virtual laboratory architecture has been designed to be both *scalable* and *extensible*. Scalable in terms of enabling a large number of participants to collaboratively setup, execute and discuss about virtual scientific experiments and extensible in terms of enabling the addition of virtual scientific experiments that make use of the deployed collaborative parameter setup and GRID infrastructure by mainly customizing existing generic components and converting them to Web Services.

This is achieved by decoupling computationally intensive parts of a simulation from the non-intensive parts. The intensive parts involve the simulation logic and the visualization of the results both of which are undertaken by the GRID or the user browsers. The non-intensive parts involve the collaborative setup and execution of

simulations by users of learner communities which are undertaken by customizable web applications.

In the rest of the paper, we first describe some recent related work in the field of Virtual Laboratories for educational purposes. We then proceed to describe the proposed GRID-based system architecture, which allows for the necessary scalability to tackle the computational demands of heavy engineering or scientific computing experiments with a focus on network simulations ranging from Wireless Sensor Networks to Wi-Fi (and many other types of networks). We describe in particular detail our *generic collaborative simulation management architecture* which lies at the heart of the whole system and enables teams of students and researchers logged-on via a web-based interface to the application, to collaborate in setting up the virtual experiment. Then we describe the *real-time interaction capabilities of our architecture*, and after that, we proceed with details of the first prototype implementation using SENSASIM as the core simulation engine. We provide some initial results from running a number of experiments with this tool, and then discuss our conclusions and some future directions this research might take.

### *1.1. Related Work*

Even though Virtual Scientific Experiments have been studied thoroughly and very extensively over the past 15-20 years within fields such as simulation software, scientific computing and visualization or even Virtual Reality, the use of Virtual Laboratories as a means for enhancing education with particular emphasis on team-formation and students community building has not received much attention until recently. This situation is rapidly changing though, with the advent of distance learning technologies and the formation of distributed virtual University classrooms. Within the ELeGI project, the use of Virtual Reality tools and simulations was studied in order to provide GRID-enabled physics simulations (Szczytowski, 2005). Simulations of physical phenomena can be written VCLab and be visualized in the learner's browser using these technologies. While it is possible that two or more students share the same simulation session, the focus is on individual learning and not so much on team-formation or community building. Community building –albeit without focus on formal learning environments- takes a more prominent role in the work described in (Eisenstadt et. al., 2005) where the authors advocate a “light” approach to learning through peer-interactions. Tzafestas (2000) presents a stand-alone simulation software for educational purposes, and in particular for engineering education, however no distributed or GRID infrastructure is utilized in any way. Our work is different in that we place heavy emphasis on the optimal utilization of GRID infrastructure so as to support multiple collaborative lab sessions of geographically separated student teams. Mikropoulos (2003) describes in detail requirements for educational software for elementary school-children, with emphasis on the user interface mechanisms that enhance and ease the learning process. In the same context of elementary education, Kfir (2001) discusses how children can benefit from using an interactive Computer Graphics application. That paper presents a project that attempts to produce a generic tool to advance the educational use of virtual laboratories, but the focus is on school children whereas our work focuses on advanced learner groups such as electrical and computer engineering graduate students. In (Ferry et. al., 2006) an online simulation tool is described that was developed to enhance the pre-service teacher practicum experience in the area of literacy teaching. And in (Wong et. al. 2006), the authors

present an edutainment approach to the problem of online life-long learning, with the introduction of a Role-Playing-Game that is designed to sharpen players' skills in management and banking. The game uses Rich Internet Application technologies, but without any focus on GRID technologies. We envisage that the infrastructure for Virtual Laboratories that we have built and which is based on turn-taking order, will be easily extended to provide management experiments in role-playing management games, which will simulate the course of a virtual business under the control of the students playing the roles of prospective executives.

## 2. System Architecture

Simulation based experiments that are collaboratively setup, executed and discussed in a group share certain functionality that is not particular to the simulation itself but to the workflow of a Virtual Scientific Experiment (VSE). This very functionality can be decoupled from simulation-specific features and made available in a generic component for the collaborative execution of VSEs.

### 2.1. Collaborative Simulation Management Generic Architecture

It has been possible to provide an architecture for the collaborative parameter setup as well as the visualization and the interaction management of simulation-oriented VSEs that includes:

- (i) a number of simulation components, which can be computation-intensive and can be executed using a GRID infrastructure;
- (ii) a generic component to support the collaboration mechanisms of VSEs that is not computation-intensive and can be executed as a Web application;
- (iii) support for the addition of simulation specific components on the Web application for parameter setup, visualization and the communication between the Web application and the computation-intensive GRID services that perform the simulations.

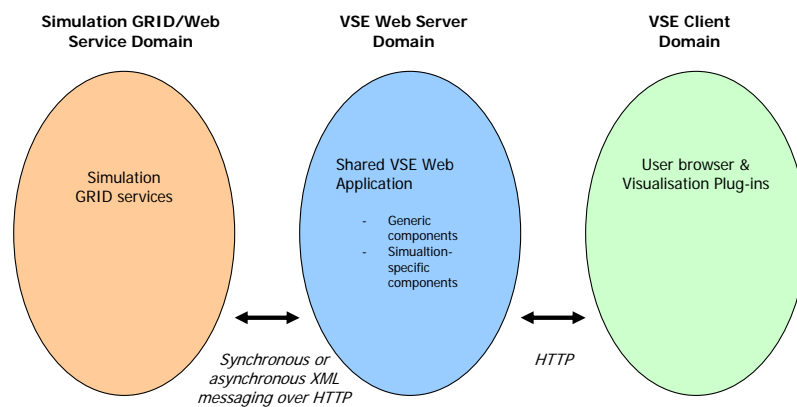


Figure 1: Collaborative simulation-based VSE domains

As shown in Figure 1, this architecture considers three distinct domains: the Web Services/GRID domain with the computation-intensive simulation GRID services, the VSE Web Server domain with the shared Web application for collaborative parameter setup and simulation visualization and the VSE client domain with the browsers of the users of the VSE applications including the necessary visualization plug-ins.

The requirements for this architecture have been:

- To enable learning communities to collaboratively setup and execute VSEs taking into consideration learning community roles, authentication and authorization.
- To allow for collaborative interaction and visualization of experiments in real-time.
- To provide flexible access to VSEs over the Web.
- To leverage the flexibility of GRID and Web Service technologies for the efficient execution of VSEs.
- To enable extensibility of the framework by supporting different types of simulation-based VSEs.
- To enable scalability of the framework in terms of participating community members, number of communities and number of supported simulation VSEs.

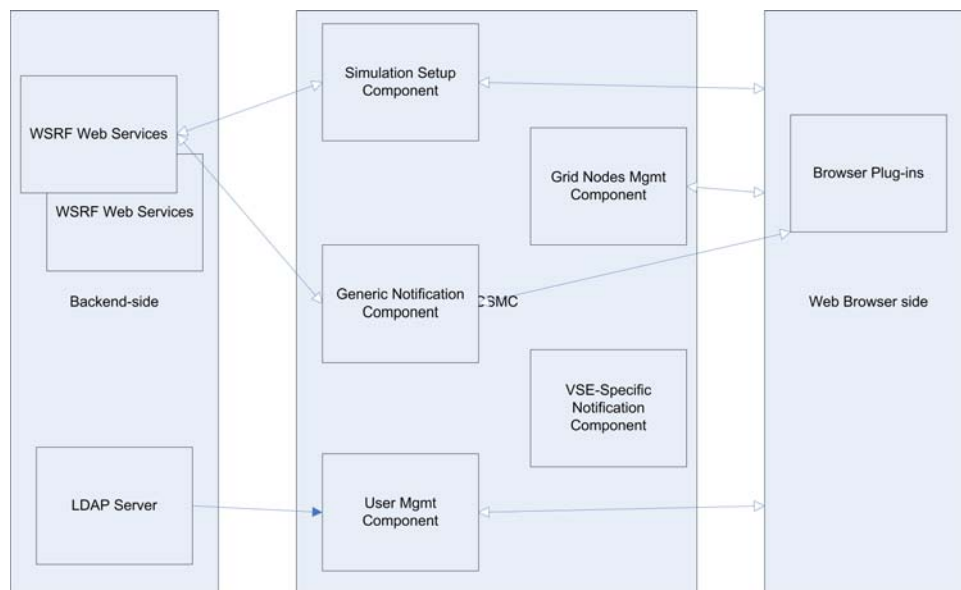


Figure 1: Collaborative Simulation Management Component Architecture

A more detailed diagram of the architecture is shown in Figure 2. It illustrates a *Collaborative Simulation Management Component (CSMC)* that enables the collaborative as well as private setup, execution and presentation of different types of VSEs to virtual communities or individual users, which is hosted on a Web server.

The CSMC implements functionality allowing a virtual community to collaboratively setup generic and specific parameters of simulation-based VSEs and the mechanisms for the communication between the CSMC and the simulation GRID

services as well as the users. In addition, it enables the management of users and GRID nodes that can be used to host simulation GRID services. The CSMC provides the means to extend the repertoire of the simulations it supports by enabling a flexible addition of simulation GRID services and configuration of the visualization mechanisms and of the simulation-specific parameters for these services.

The architecture of the *CSMC* itself as depicted in Figure 2 includes the following components:

**Generic Notification Component;** which is able to receive notifications regarding the status of a simulation from the GRID-based simulation services, to process them and to forward them to the Browser-side visualization plug-ins.

**Simulation Setup Component;** which sets-up the appropriate GRID-based simulation services given a set of collaboratively determined parameters. The parameters are used to populate an XML Settings file that describes a set of attribute-value pairs. This set of attribute-value pairs (see Figure 3) describes the values given to each VSE-specific parameter by the users from the GUI rendered in their browser. Accordingly, the defined WSDL for the back-end simulation engine web service defines an XML file that must be passed in as input to the initialization of a service execution. In this way, the front-end component remains completely agnostic of the settings themselves or the way they should be organized and rendered in the GUI.

**User Management Component;** which allows the initial authentication of users (in collaboration with LDAP servers), and is also responsible for enabling the administrator to add/delete/enable/disable users from running simulations. It also allows the administrator to force token-passing from a slow user to any user that they choose, which can be useful in cases of users that are un-decided or un-responsive during collaborative parameter set-up.

**GRID-nodes Management Component;** which allows the management of the available GRID nodes.

**VSE-specific Notification Component;** it is optional and allows handling of notifications from legacy GRID-based simulation services to the VSE Parameter Setup Application.

```

<?xml version="1.0" encoding="UTF-8" ?>
- <settings>
- <group id="general" label="General" folded="false">
- <setting id="scenario" type="Integer" value="7">
- <htmlRender label="Scenario" displayAs="comboBox">
  <option label="Broadcast Flooding" value="1" />
  <option label="Normal Flooding" value="2" />
  <option label="Bidirectional Flooding" value="3" />
  <option label="Speed Broadcast" value="4" />
  </htmlRender>
</setting>
- <setting id="thetaTheshold" type="Double" value="130.0">
- <htmlRender label="Theta threshold" displayAs="textbox">
  <validation min="100.0" max="200.0" required="true" />
  <helpTooltip>This a help tooltip for the Theta threshold setting</helpTooltip>
  </htmlRender>
</setting>
- <setting id="broadcastMode" type="Boolean" value="false">
- <htmlRender label="Broadcast mode" displayAs="radioButtons">
  <option label="Enabled" value="true" />
  <option label="Disabled" value="false" />
  </htmlRender>
</setting>
</group>
<setting id="bpsymbol" type="Integer" value="8" />
</settings>

```

Figure 3: VSE Settings.xml instance. Each VSE has its own settings, but each VSE type adheres to the same schema, which is used to render the parameter setup part of the GUI

The CSMC architecture is highly reusable for different web-based applications for collaborative setup of VSE simulations. The only programming that may be required when this framework is to be used for a specific simulation VSE is to provide a visualization plug-in for the browser-side part of the application and/or a VSE-specific notification component if necessary. Of course, providing a Web Service that adheres to the WSDL defined by the architecture is a must, otherwise, a plug-in component that invokes the WSRF service that does not comply to the proposed WSDL definition is required.

In addition to the visualization plug-in the setup of a simulation-specific configuration file is required. Specifically, the whole system can therefore be configured by means of two configuration files:

**Generic Configuration File;** there is a single instance of this file. The schema of the configuration file provides for the identification of:

- The role/community/authentication services with which the application will be able to communicate.
- The directory in which VSE-specific Configuration files can be found.
- Other parameters concerning the look and feel as well as configuration of the component.

**VSE-specific Configuration File;** see Figure 3. There is one instance for *each type* of VSE supported by the framework. The schema of the VSE-specific configuration file provides for elements such as:

- A unique identifier of the VSE (URI)
- Short Description of the VSE

- The names of the *tabs* that will be presented for parameter setup. A ‘*General*’ Tab that will enable *Submitting, Saving, Deleting* Parameter Sets will be available for all VSEs. Additional (optional) tabs for setting up will be identified and described by additional elements.
- The URL(s) of the Simulation Web Service(s) to which parameters will be submitted for execution.
- A pointer to an application function that will invoke the Simulation Web Service(s) using the values in the parameter setup tab(s) when parameter submission is selected.

## 2.2. User Access Control

The users of the virtual community participating in such an experiment have different access levels. These levels depend on the role that each member has in the community. Typically, instructors and administrators have full privileges to modify parameters, start simulation jobs, and/or modify simulation execution in real-time. Students on the other hand can be granted or denied modify and submit privileges on the fly by the administrator-instructor of the session. The user-interface of this tool is shown in Figure 4.

Sensasim users					
Type	Username	Login at	Start sim	Has token	Actions
	ichr	13:42:26 23-07-2007			
	user1	13:44:07 23-07-2007			<input type="button" value="Kick"/>

Click on and in order to change the settings

Figure 4: System Administrative Interface for User Management

The Administrator can perform the following actions:

- Enable/Disable the ability of a simple user to start a private simulation.
- Pass the Token of the public shared simulation to the next user or give the Token to a specific user in order to by-pass turn-taking.

## 2.3. Using the Collaborative Simulation Management Component

The *CSMC* is the entry point of virtual community users to a variety of virtual scientific experiments that may be available. The steps that need to be followed when a virtual community member accesses this application can be outlined as follows:

The user is prompted to login and select the *type* of VSE they wish to run. The *CSMC* may subsequently communicate with role/community services to determine the access rights of the user to different simulations. A number of VSEs may be available for access.

The user is prompted to select the *instance* of a VSE type they want to join (collaborative public simulation) or to *create a new instance* (run a private experiment). A VSE instance is a running VSE simulation of a specific kind with a specific number



of participants and a particular state. Users are also allowed to leave a VSE instance or (given they have the rights) to delete VSE instances.

For a selected instance of a VSE type and subject to rights a user can:

- Setup parameters shared by all the users joining the same instance of the VSE type *in a turn-taking manner* simulated by a token-passing scheme.
- Submit parameters for execution to the Simulation Web Service(s).
- Save parameter sets for future use.
- Delete previously saved parameter sets.
- Be alerted when simulation results are available.
- View and interact with simulation progress (making use of a browser-side plug-ins).
- Kick a user out of the simulation.

#### *2.4. User Interaction with the Running Simulation*

Besides allowing users to collaboratively set-up the initial parameters of the experiments, the system also allows for the on-the-fly collaborative modification of simulation state so as to allow for true interaction with the system and better understanding of system dynamics and underlying principles. This is accomplished via WS-Notification mechanisms.

WS-Notification is fully supported in GT4 WS-core (Sotomayor & Childers, 2005). The simulation engine was implemented as a NotificationProducer grid web service that is part of the WS-Notification standard implemented by GT4, so that it can spread the information to the clients that sent subscribing request messages without having a prior knowledge about them. The service can then accept a subscribe request which will contain the reference for the NotificationConsumer which is the endpoint designated to receive the Notify message that will actually represent an event of the simulation so whenever events for the running simulation are produced all the consumers – participants of the virtual experiment – will get notified automatically.

In the rest of this section, we describe in detail the interactions of the users with the system. Figure 5 shows the overall Graphical User Interface, which consists of 5 major areas.

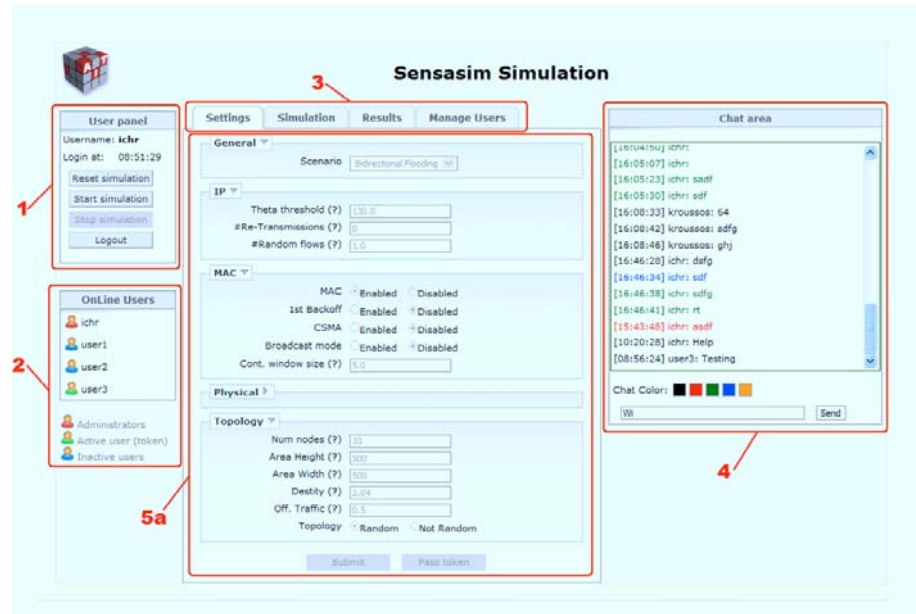


Figure 5: Overall System Graphical User Interface

**1. User panel:** Provides user information and actions. The user can **Start**, **Stop**, **Reset** the simulation (for the Administrator user the public simulation, for simple user the private one) or **Logout**

**2. Online users:** Displays the Online users status. The Administrator user/tutor is displayed in red color. The simple users/students are displayed using blue color except for the user who has the token (can enter settings) which is displayed in green color.

**3. Application tabs:** Displays the Application tabs. These are **Settings**, **Simulation** and **Results** for the student users. The Administrator user also has access to the **Manage users** (and **Manage Grids**) tab.

**4. Chat area:** Provides a simple global broadcasting chat. The user can change his/her displayed message color.

**5a. Settings tab:** This is the area where the users can enter settings for the simulation. Only one user at any given time can enter settings as the system implements a turn-taking approach similar to many other major collaborative internet-based tools. When a user gets the token (based on a Round-Robin, first-come-first-serve manner) the settings fields and buttons are enabled. Then the user can either **Pass token** to the next user or change and **Submit** the settings. The token can be passed many rounds and it is up to the decision of Administrator when the simulation can be started.

**5b. Simulation tab:** This tab is enabled when the Administrator starts the simulation. In this type of experiment, the user can click on a node in order to simulate a random event that causes the particular node to go off-line.

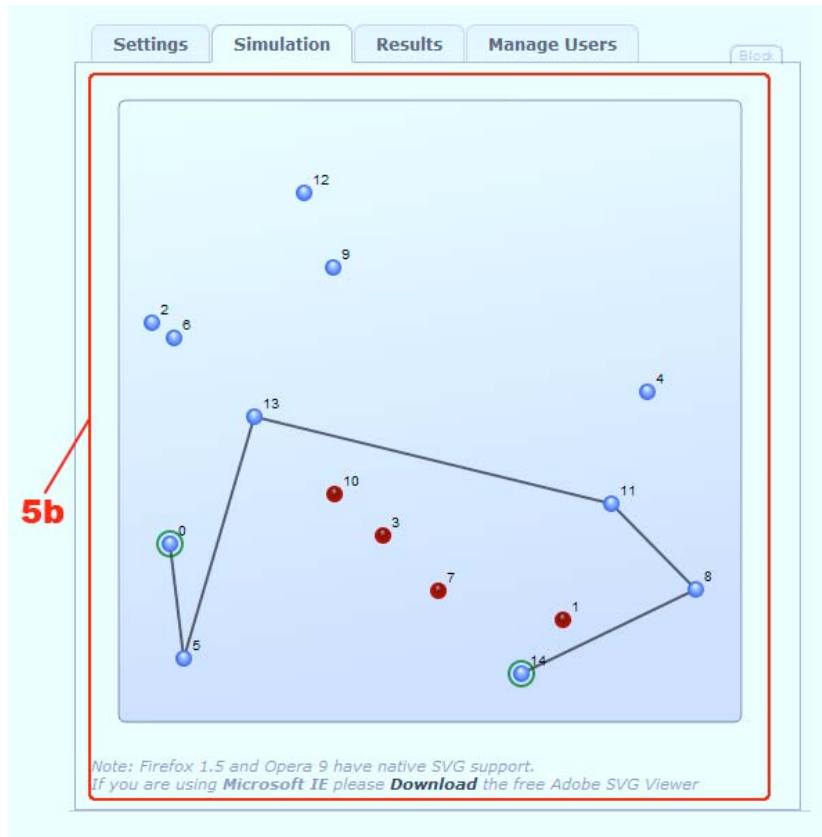


Figure 6: Simulation Area where the user can visualize the state of the simulation and also interact with the running simulation

**5c. Results tab:** Once a simulation finishes, statistics can be collected from the Results tab of the GUI, as shown on Figure 7.

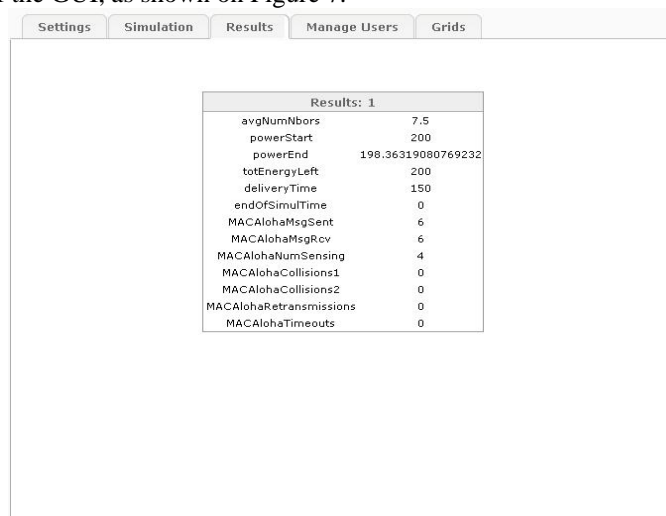


Figure 7: Results Area of the GUI. The results can be copied to the clipboard for further processing

### 3. Implementation and Deployment of the Generic VSE Infrastructure

A high-level diagram illustrating the deployment of the implemented system architecture is depicted in Figure 8. As can be seen, access to the application is provided via a web browser interface. A user logs in to the web-site and after she/he is authenticated via LDAP authentication, she/he may choose to join a collaborative simulation session (default choice) or, assuming there is an idle GRID node, she/he may choose to enter a “private simulation session” to which no other common user will have access.

A front-end Apache Tomcat server runs a J2EE application that sends a JSP with a number of components in it to the user browser. The user browser establishes a connection back to the front-end server for real-time synchronized communication with the system via AJAX technology. The user interface includes a presence window showing the team members that are currently on-line, a chat interface for instant messaging between all team members, and the main window showing the simulation environment. These built-in instant messaging and presence mechanisms offer a significant advantage for the virtual community members that have to interact heavily during the experiment. Usually during such an experiment other videoconferencing tools are also running in each user’s machine in order to enhance communication between team members.

The actual simulation engine runs on any of a number of Grid nodes running the Globus Toolkit 4 (GT4) WS-core package which is platform independent as it is completely written in Java (Sotomayor & Childers, 2005). Each Grid node is connected to a Grid Server Wrapper that distributes simulation jobs between the Grid nodes.

The simulation engine itself is implemented as a WSRF Web Service with a simple Web Services Description Language (WSDL) file associated with it, allowing remote clients to call the service (Joseph & Fellenstein, 2004), and to also interact with it, changing its state on-the-fly.

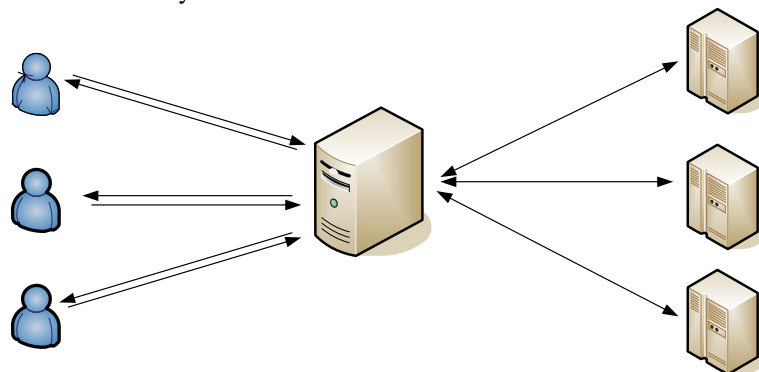


Figure 8: Collaborative Interactive Virtual Scientific Experiments Architecture

As shown in Figure 8, the CSMC co-operates with Browser-side plug-ins via AJAX technology for the collaborative set-up of the simulation parameters and forwards them to the GRID-domain Simulation Web Services that perform the

simulation logic. It also connects to a number of Grid Services to enable Authentication, Virtual Community Management and Role-based access control to VSE resources. It also communicates with other Browser-side plug-ins – such as Adobe SVGViewer for the Internet Explorer – in order to forward the state of the experiment to the browser-side renderer that renders the simulation progress via standard WS-Notification mechanisms in a callback fashion. Conversely, it also accepts user interactions indicating a request for a change in the state of the simulation and relays them back to the Web service. It is an event-driven model that is employed for the submission of the simulation setup parameters and state notification. Thus, after simulation parameters have been submitted for processing to the GRID-based simulation services, notifications are sent back to the CSMC as the component has subscribed itself as a listener for the execution of the Web services; these notifications, in turn, are forwarded to the Browser-side for display by the appropriate plug-ins.

Communication between the different components is over SOAP/HTTP or XML/HTTP.

#### **4. Prototype Implementation of Sensor Network Simulation Service (SENSASIM)**

As a first proof-of-concept of our architecture for Grid-based interactive collaborative virtual scientific experiments for virtual communities, we chose as our simulation engine an Open-Source tool for Wireless Sensor Networks that is written entirely in Java and that can support up to thousands of sensor nodes in the network topology. The tool, called SENSASIM (Rayess et. al., 2006) as already mentioned is a simulator for wireless multi-hop networks with multiple directional antennas. The simulator can be used for any ad hoc wireless network, but the focus is on the more energy-limited and constrained sensor networks. The supported physical model includes use of directional antennas, enabling the switching of the antennas' lobe at will and the finding of direction of arrival (DoA) of the signal. Two variants of Aloha and one CSMA (Carrier Sense Multiple Access) are the major MAC (Medium Access Control) protocols implemented. At the network layer, emphasis was given on SPEED (Scalable Protocols for Efficient Event Delivery in Sensor Networks) (Dimitriou et. al., 2004) and its enhancements.

The support for all these layers of the network with detailed models is a good fit for a successful laboratory experiment in the form of a VSE (Virtual Scientific Experiment) for a course in Wireless Sensor Networks that is offered in the MSIN program. The simulation engine architecture is shown in Figure 9.

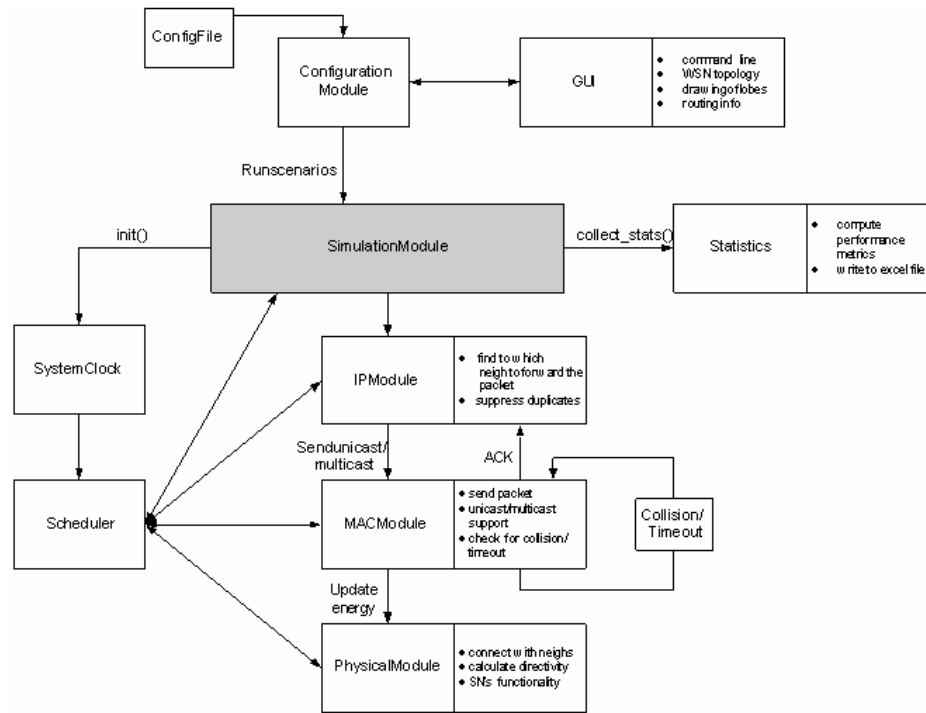


Figure 9: SENSASIM Engine Architecture

We modified the simulation engine to allow for on-the-fly modifications of the simulation state and sending/receiving and responding to events such as users clicking on sensor nodes visualized on their browser to activate/deactivate them. The engine simulates the process of an event generated in a random node in the sensor network – such as a measurement reading of a quantity- and the propagation of this event to a final “aggregator” node (Rayess et. al. 2006).

The static structure of the design of CSMC is shown in Figure 10. This figure shows the major classes used to build the main module.

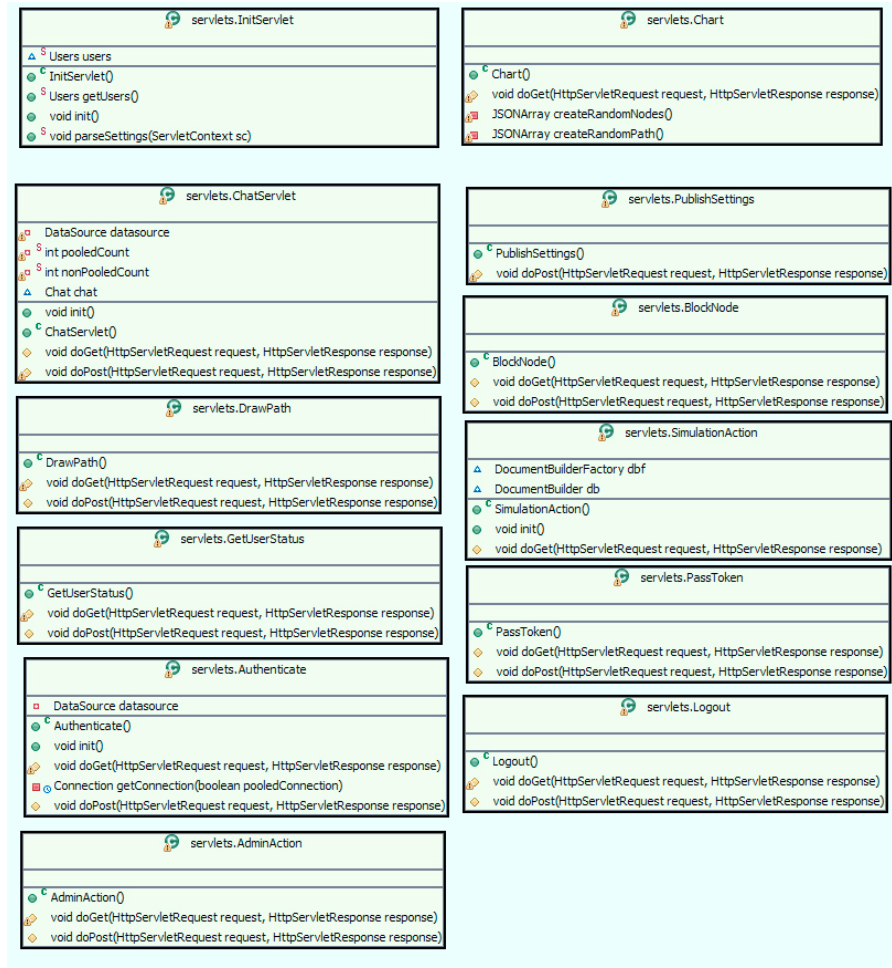


Figure 10: UML Class Diagram produced by the Eclipse IDE ® showing most of the major classes in CSMC module

A grid cluster of three SUN/Solaris nodes running GT4's "globus-start-container" and another cluster of two Windows nodes running the WS-Core component of GT4 were created. Each Windows node is equipped with dual Pentium4 processors running at 3.2 GHz , 1GB of RAM and 120GB disk drives. These clusters are located, respectively, in the Software Systems and Web Applications Laboratory and in the Wireless Networks Lab of AIT. Another cluster located at CMU that will connect to create a single even more powerful distributed laboratory is under way, but this extra power is needed mainly for research purposes. For the educational aspects of the virtual lab, five grid nodes were found to be more than adequate. Each of the GRID nodes may start a simulation instance upon request from the front-end node of the architecture assuming it is idle at the time of the request.

## 5. System Evaluation

An iterative approach was adopted for the evaluation and assessment of the deployed Grid-based VSE solution, which involved two major iterations: (i) the deployment and evaluation of the first prototype and (ii) the deployment and evaluation of the second prototype, further developed and enhanced based on user feedback during the first prototype evaluation. The results of each iteration cycle are presented in the following sections. Although further iterations are envisaged in the future, the evaluation results of the second iteration already showcase increased acceptance of the Grid-based interactive SENSASIM VSE by the participating students from remote collaborating institutions.

The Grid-based VSE for Sensor networks was deployed locally at Athens Information Technology (AIT), Athens, Greece and was available for evaluation to groups of students located at AIT and at the CyLab Japan site of the Information Networking Institute of Carnegie Mellon University in Kobe, Japan, during the first prototype evaluation, and to a single group of students located at AIT during the second prototype evaluation

### 5.1. First prototype evaluation

The evaluation of the first prototype of the VSE was organized in two stages (Christou et. al, 2007). The first stage involved the evaluation of the prototype by a group of 25 students following a demonstration of the VSE to the group and a Q&A session following that. Subsequently, the students filled out questionnaires detailing their evaluation of the first prototype. The results of this evaluation showed that almost two in three students appreciated the availability of communications facilities during the VSE session and they considered them a very important element of the VSE. Over eighty percent of the students preferred Web-based access to the VSE and voice and text based facilities for communication with each other and with the tutor during the setup or execution of simulations. From this first stage of the evaluation of the first prototype there was clear demand for three additional features in the SENSASIM VSE: (i) interactivity with the VSE in progress by modifying running simulations, (ii) real-time graphical interfaces showing the simulation progress and (iii) turn-taking for the setup and modification of the simulation parameters. The students were content with the performance of the simulation over Grid.

A second stage of the initial evaluation involved a geographically distributed session between graduate students at AIT in Athens, Greece and graduate students at CyLab, Kobe, Japan. During the deployment of the Virtual Lab Course on Wireless Sensor Networks involving the execution of the first prototype of the SENSASIM-based Virtual Scientific Experiment (VSE) over Grid nodes, A.I.T. and CyLab Japan students were given the opportunity to form remote laboratory teams attending a common lab course, perform virtual scientific experiments in Wireless Sensor Networks to observe their simulated behaviour after changing a number of varying parameters and collaborate via chat mechanisms to exchange opinions on parameter values and justify the obtained results. During the virtual lab session, the responsible teacher presented the required theoretical concepts of Wireless Sensor Networks and explained the entire process of the simulation using the SENSASIM simulator, setting the values of the varying parameters and justifying the behaviour to students.



The trial involved initially students and tutors (1 teacher at A.I.T., 1 teacher at CyLab Japan and a number of assisting tutors) from both A.I.T. and CyLab Japan gathering at their local lab classes at a prescheduled time for the virtual lab course – announcements for this event were made well in advance to interested students. All participants were equipped with PCs, web cameras, speakers and microphones. All participants connected to the indicated Flash Meeting Booking URL (<http://flashmeeting.open.ac.uk/>) to enter the virtual meeting room and thus initiate communication and collaboration among remote lab classes via videoconferencing facilities. In this way students were able to ask questions and receive answers from teachers/tutors during the hands-on exercises. At the prescheduled time, the virtual lab course was initiated by the teachers at A.I.T. and CyLab Japan. The teacher at CyLab Japan made a 30 minute introduction focusing on challenges. Then the teacher at A.I.T. presented the networking challenges and introduced the SENSASIM simulator. Access to the slides of presentations was possible prior to the event from a web site where teachers uploaded them. Then, students were asked to access the SENSASIM server from their PCs through a specified URL from a Web page. Students had the chance to download the stand-alone version of SENSASIM prior to the event from an indicated Web site in order to get acquainted with it. As soon as students logged in to SENSASIM, the demonstration started by a tutor at A.I.T. A number of hands-on exercises were run to demonstrate the simulated behavior of wireless sensor networks under varying parameters of its model. Students asked questions and received answers during that part.

After the session ended, the participants were asked to complete an appropriately formatted Questionnaire that included questions both on system usability as well as on the collaboration and learning added value of the VSE prototype and its usefulness in allowing for efficient execution of virtual lab sessions and collaborative interactive simulations. The evaluation results provided a rating of 2.71 for the collaboration process in a scale 1 (process worked very well) to 5 (process did not work at all). The rating for the interest of the VSE was 2.0 in a scale 1 (very interesting) to 5 (not interesting at all), while the rating for the communication tools was 2.28 in a scale 1 (appropriate) to 5 (not appropriate). The overall rating of the system was 2.4 as far as the integration of its components was concerned in a scale 1 (strongly agree) to 5 (strongly disagree).

In addition to the evaluation of the Grid-based VSE simulator in terms of learning and collaboration, a properly structured questionnaire aimed to allow for the calculation of the system usability score (SUS) of the SENSASIM VSE simulator. Due to network congestion problems during the execution of the lab session among the remote collaborating institutions, communication and collaboration among the remote sites was bad, a fact that affected the evaluation result. The SUS score was almost 70% (in a scale 0-100).

An analysis of the evaluation results of the above two stages indicated that *the additional features enabling interactivity, turn-taking and real-time graphics would improve the performance of the VSE in terms of supporting collaboration processes, while the communication facilities could be improved, despite the fact that Internet congestion that was experienced during the second stage of the evaluation may have been the prime cause for the low rating of the communication tools rather than the functionality of the tools themselves.*

## 5.2. Second prototype evaluation

Taking into account the user feedback during the evaluation of the first prototype, a second Grid-based SENSASIM prototype was developed as detailed in the previous sections. The second prototype addressed the issues that were identified during the analysis of the evaluation results of the first prototype. The initial version of this second prototype had a functional but less pleasing Graphical User Interface and it was loosely integrated with the Grid-aware Intelligent Web Teacher learning portal IWT-GA, in order to provide an integrated approach to management of learning resources within the context of the ELeGI project. IWT-GA is available at <http://elearning.diima.unisa.it/iwtportal/DesktopDefault.aspx>. With this integration, the students actually obtained a more holistic laboratory experience since from one portal, they could first access related Units of Learning (UoL) and study the required theoretical material and then join the Virtual Laboratory without leaving the learning portal. Access to the initial version of our second prototype through the IWT-GA portal is shown in Figure 11.

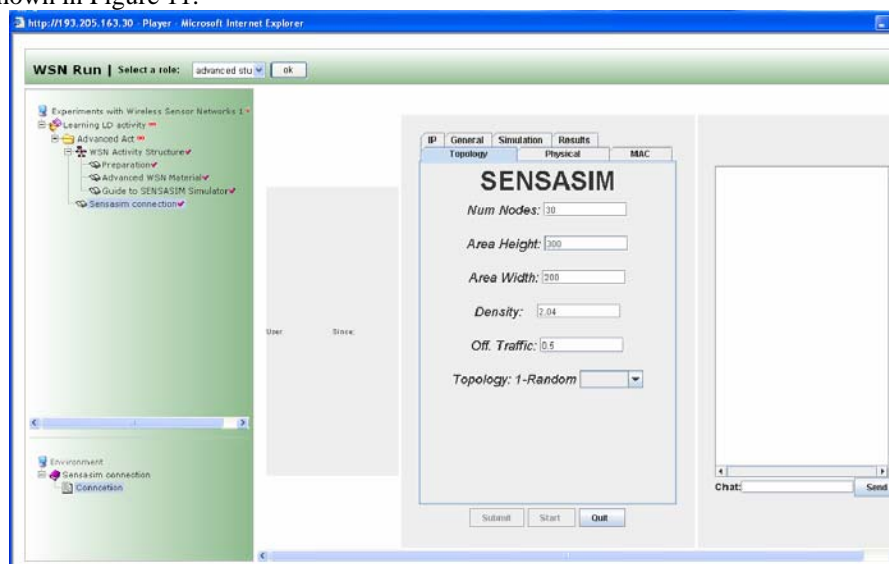


Figure 11: GRID-SENSASIM Access Through IWT-GA Learning Portal

A group of five students who were attending an elective course on wireless sensor networks at AIT were asked to participate in an in-depth evaluation of the second prototype, following similar deployment details of the 1st evaluation stage, and to fill out properly constructed questionnaires to evaluate both system usability and learning and collaboration potential of the simulator.

Based on the evaluation results, the SUS score of the second prototype was 80%, which was considered encouraging and was increased with respect to its initial value for the first prototype. The rating of the appropriateness of the communication tools was 2.00 in a scale of 1 (appropriate) to 5 (inappropriate). The improvement of this rating can be explained by the better performance of the local network for voice and text based communication and certain improvements in the deployed tools.

The evaluation group was asked, among others, to evaluate the appropriateness of media presentation of the VSE providing feedback on the new features enabling

interactivity and real-time graphics presentation. The responses, which are illustrated in Figure 12, provide a rating of media appropriateness of 1.00 in a scale of 1 (appropriate) to 5 (inappropriate), which indicates that the improvements in the second prototype were well received.

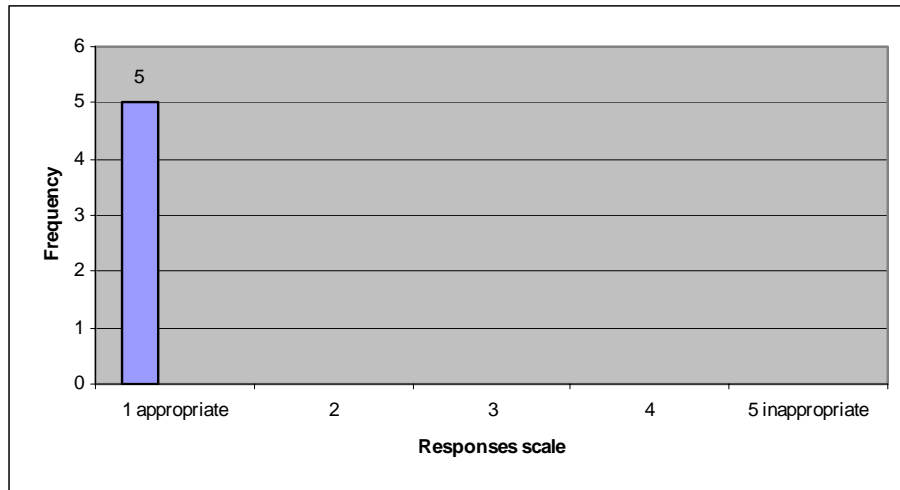


Figure 12: Evaluation of the appropriateness of media presentation in the second prototype.

In addition, students were asked to evaluate the collaboration process as implemented in the second prototype with increased interactivity and turn taking. The evaluation results, as illustrated in Figure 11, provide a rating of 1.80 in a scale of 1 (worked well) to 5 (did not work well at all), which is a clear improvement of the rating of the first prototype.

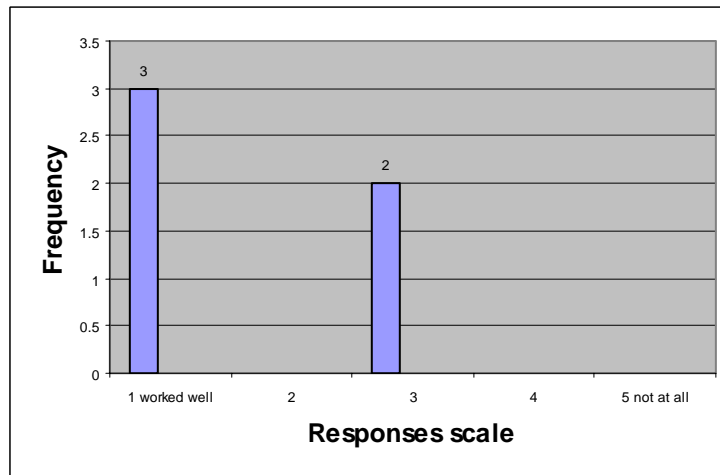


Figure 13: Evaluation of the degree of collaboration process performance in the second prototype.

The comprehensibility of the VSE was rated 1.40 in a scale 1 (easy to understand) to 5 (difficult to understand), while the clarity of the learning experience to the students was rated 1.60 in a scale 1 (strongly agree) to 5 (strongly disagree).

Since the evaluation of the second prototype, further improvements were applied on the SENSASIM VSE regarding text-based communication (improved chatting features, color-coding of the text of each participant) and media presentation (use of scalable vector graphics – SVG) to further improve the usability, performance and interoperability of the SENSASIM VSE with different browsers as well as the reuse of the collaborative parameter setup component with different VSEs. The current Graphical User Interface of the system is shown in Figures 4 through 7.

## 6. Conclusions & Future Directions

We have described a generic Grid-based architecture that enables distributed learner communities to conduct virtual laboratory experiments via the web in a fully collaborative manner, thus enhancing community building among participants and resulting in overall better learning experiences. Team members interact with each other via integrated instant messaging and complementary standard video-conferencing facilities. More significantly however, the architecture allows team members to participate in the setup of the laboratory experiment's parameter set-up, and then to fully visualize the simulated lab experiment progress in real-time and interact with it modifying its execution path while it is actually running. The added value and even necessity of this interaction feature was anticipated even before the first prototype assessment phase by Prof. Takeshi Utsumi in private communications within the ELeGI project, and indeed it proved to be so, as it enables students to gain much deeper understanding of the principles being demonstrated by the experiment, by taking full advantage of the built-in notification mechanisms of Grid Web Services.

The architecture has been instantiated in a first proof-of-concept with a Wireless Sensor Network simulation engine and was well appreciated by teams of students that evaluated its features and performance in Athens Information Technology. We are working towards instantiating the same architecture with a Wireless Networks simulator based on the ns2 engine (ns2, 2007) that will allow labs to complement theoretical lecture material in courses on Wi-Fi Networks (Sturgeon et. al. 2006). The current version of our Virtual Laboratory is available in <http://palantir.ait.gr:8888/Sensasim>. Users can log in with the guest login “user1” and password “user1”.

Currently, the power of the Grid is used to run simultaneously different simulation sessions. We are also working towards parallelizing the two simulators to take advantage of the Grid so as to distribute a single large-scale simulation among the Grid nodes to achieve near-linear speedup in simulation execution.

### References

- I. T. Christou, S. Efremidis, T. Tiropanis and A. Kalis, 2007: “Grid-based Virtual Laboratory Experiments for a Graduate Course on Sensor Networks”, IEEE Transactions on Education, Volume 50, Issue 1, pp:17 – 26.
- G. Rayess, I. Tsirilakis, C. Kakogiannis, A. Kalis, A. G. Kanatas, Ph. Constantinou, 2006: “Java-based simulator for wireless multi-hop networks using directional antennas”, European Wireless Conference 2006, Athens, April 2-5, 2006
- B. Sotomayor, L. Childers, 2005: “Globus Toolkit 4: Programming Java Services”, Morgan-Kaufmann, San Francisco, CA.
- J. Joseph, and C. Fellenstein, 2004: “Grid Computing”, Prentice-Hall, Upper Saddle River, NJ.

- T. Dimitriou, I. Krontiris, F. Nikakis and P. Spirakis, 2004: "SPEED: Scalable Protocols for Efficient Event Delivery in Sensor Networks", Networking 2004, Athens, Greece
- ns2. 2007: "The Network Simulator ns-2". [cited 2007 January]; Available from: <http://www.isi.edu/nsnam/ns/index.html>.
- T. C. Sturgeon, C. Allison, and A. Miller. 2006: "A WiFi Virtual Laboratory". in 7th Annual Conference of the Subject Centre for Information and Computer Sciences. 2006. Dublin, Ireland: HE Academy.
- E.S.Tzafestas, 2000: "Experiences from the Development and Use of Simulation Software for Complex Systems Education", Proceedings of the World Conference on the WWW and Internet (WebNet-2000), November 2000, San Antonio, TX.
- T.A. Mikropoulos, 2003: "Educational Software", Klidarithmos, Athens, Greece, (In Greek)
- R. Eli Kfir, 2001: "Virtual Laboratories in Education", Proceedings of ACM 1<sup>st</sup> Intl. Conference on Computer Graphics, Virtual Reality and Visualization (AFRIGRAPH), Cape-town, South Africa, pp. 27-31.
- P. Szczytowski, 2005: "VCLab as an Example of GRIDifying Virtual Scientific Experiments", Proceedings of ELeGI 1<sup>st</sup> Intl. Conference on Advanced Technology for Enhanced Learning, March 2005, Vico Equense, Italy.
- M. Eisenstad, J. Komzak, & S. A. Cerri, 2005: "Peer Conversations for e-Learning in the Grid", Proceedings of ELeGI 1<sup>st</sup> Intl. Conference on Advanced Technology for Enhanced Learning, March 2005, Vico Equense, Italy.
- B. Ferry, L. Kervin, & K. Rudd, 2006: "Use of Rich Internet Applications to develop an online classroom-based simulation", International Journal of Continuing Engineering Education and Life-Long Learning - Vol. 16, No.3/4 pp. 200 – 214.
- W.-C. Wong, M. Sheu, E. Moy, 2006: "A Rich Internet Application edutainment training system", International Journal of Continuing Engineering Education and Life-Long Learning - Vol. 16, No.3/4 pp. 280 – 296.