Grid-Based Virtual Laboratory Experiments for a Graduate Course on Sensor Networks

Ioannis T. Christou, *Member, IEEE*, Sofoklis Efremidis, Thanassis Tiropanis, *Member, IEEE*, and Antonis Kalis, *Member, IEEE*

Abstract—This paper presents the pedagogical and technical challenges the authors faced in developing a distributed laboratory for the execution of virtual scientific experiments (VSEs) superimposed on a Grid infrastructure, for a course on sensor networks that is part of the Master's in Information Networking (MSIN) program jointly offered by Carnegie Mellon University (CMU), USA and Athens Information Technology (AIT), Athens, Greece. The MSIN program utilizes virtual classroom technologies because of its strong distance learning component. Courses taught by CMU faculty are attended in real-time by students in Athens, Greece, via video-wall teleconferencing sessions. Vice versa, visiting CMU faculty to AIT teach classes that are attended by students at CMU. Students in both institutions enjoy full interactivity with their classmates on the other side of the Atlantic Ocean. A distributed shared virtual laboratory is needed for many of the more empirical courses. This paper describes the challenges and issues the authors faced in developing such a lab.

Index Terms—Discrete-event simulation, distributed computing, learning systems.

I. INTRODUCTION

The Master's in Information Networking (MSIN) program (www.ini.cmu.edu) is a graduate program in information networking offered jointly by Carnegie Mellon University, Pittsburgh, PA, USA, and the Athens Information Technology (www.ait.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 in real-time what is happening in the other class. As a result, instructors and students at both institutions 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 course work in teams in a laboratory environment. These teams are formed locally, meaning that students at the AIT campus form their own teams; and students at the CMU campus form their own teams; however, teams across the two campuses can not be easily formed because of lack of support of a distributed laboratory that would enable closer student interaction and collaboration.

Consequently, the need has been identified for a virtual laboratory that would enable students from both campuses to join in teams to perform lab work essential for many engineering courses offered in the context of the program. An example of when the use of a virtual laboratory would be beneficial to students is the "sensor networks" course. For this course, the last author developed with his team of M.Sc. students a simulator for sensor networks [1] that was well accepted from the sensor network research community, because of its ease of use and its educational and research value. The simulation was originally written in Java and ran as a stand-alone desktop application.

At the same time, AIT participates in the E.U.-funded project European Learning Grid Infrastructure (ELeGI) [2], with the main role of providing requirements and a test-bed for the technologies that the project will develop. In its role, AIT is uniquely qualified especially because of its formal educational program MSIN that actually operates as a distance learning program between two institutes that are geographically dispersed. The development of a virtual distributed laboratory for a hands-on experience in some of the more practical engineering courses was one of the major requirements for the evolution of the program and fit perfectly in the context of AIT's participation in the ELeGI project. Students from both campuses would get the opportunity to interact more closely together, reaping the benefits of experiential learning, and at the same time, the two classes would become more unified and more connected through their interactions in laboratory sessions rather than just through class attendance.

The following sections describe the authors' experiences in building such a virtual laboratory, its tools, and the experiences of the users that have tested this infrastructure. In particular, first the requirements for a laboratory in sensor networks are described, which lead to the decision to run this laboratory on top of a Grid [3], [4]. This discussion borrows from ELeGI results concerning the theoretical background on the methodologies and value of virtual scientific experiments (VSEs) [5], [6]. A description of the tools implemented or used follows, namely the Grid-based Sensor Networks Simulation, and an explanation of the communication tools for geographically separated teams of students. A discussion of the challenges faced when porting a complex stand-alone simulation to the Grid is also included. Finally, the paper provides a discussion of the student feedback to

this first version of a Virtual Distributed Laboratory for Sensor Networks. User-interface issues continue to exist, and finally suggestions on how to extend this work on other virtual laboratories for conducting VSEs are presented.

II. Sensor Network Virtual Laboratory Requirements

A. Technical Requirements

This section provides a brief overview of the technical requirements for the virtual laboratory, and the experiments that are intended to run on this lab.

The Sensor Networks course works with a number of topics [7]. After an introduction to the field of wireless sensor networks, the students are introduced to motivating problems, such as user movement tracking, localization, etc. Next, sensor networks stack protocols are studied in detail as are various routing algorithms. The course also discusses infrastructure establishment and sensor network hardware, with emphasis on antennas for wireless sensor networks. Finally, the course curriculum includes sensor network databases and security and privacy issues from a theoretical and practical perspective. Overall, the course focuses both on theoretical foundations and on practical technological considerations. The mathematical foundations on which the course builds are differential and integral calculus including differential equations, analytical geometry, and applied mathematics of wave propagation and electromagnetics. These foundations are required for the study of the various aspects of wireless sensor networks. The study of these fundamental characteristics makes clear the many constraints with which sensor networks are faced, giving rise to the search for efficient algorithms (discrete mathematics) for sensor networks management. The practical considerations of sensor networks include the modeling and simulation of virtual sensor networks on a computer.

The SENSASIM tool [1] is a Java-based simulator for wireless multihop 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 carrier sense multiple access (CSMA) are the major medium access control (MAC) protocols implemented. At the network layer, emphasis was given on scalable protocols for efficient event delivery in sensor networks (SPEED) [8] and its enhancements.

The support for all these layers of the network with detailed models is exactly what is expected for a successful laboratory experiment in the form of a VSE for this course. However, to run the simulation with all these fine details of the models, the computational requirements of the software often pose significant stress on the underlying hardware even for a single simulation run for a network of more than a few thousand sensor nodes.

Further, the original simulator was written as a stand-alone desktop Java application, which has certain limitations.

- *No team collaboration can be used for the simulation setup*, unless all team members are in front of the same desktop; thus, collaborative learning is not supported. As a consequence, geographically separated teams for the lab cannot be formed, and the sense of community building

between the students taking the course at the two different campuses is limited (or nonexistent).
- If a formal lab session is to take place for this course, the computational requirements for running all the simulations concurrently by the student teams would pose very significant burdens on the (limited number of) lab computers.

For these reasons the SENSASIM tool has to be modified in a number of ways.

- To write as a Grid Service to run on top of a powerful Grid network to take advantage of on-demand sharing of resources without the need for a permanent reservation of these resources. As teams run (in the same lab session) the same code simultaneously, but with different parameters, such a lab requires the resources of a GRID.
- To allow the parallel execution of the outer loops of a single simulation run on many—dynamically brought together—host nodes, so as to speed up simulation execution time.
- To modify its user interface to allow team collaboration in the simulator setup and team coordination and visualization of the simulator results. In this regard, the Graphical User Interface (GUI) of the tool must allow for the instantaneous update of the simulator's parameters on the screens of all participants, whenever one participant modifies any parameter; it must also allow for different privileges among the participants (the team leader will be the only one who can commit the final round of parameter setup and start the simulation, while the other participants will be only allowed to modify parameters before commit time and view the simulation progress and results on their screens.)
- To allow the use of predefined simulation configurations that lock specific parameters to appropriate values depending on the team members' profiles.

B. Pedagogical Requirements

In the Sensor Networks course offered at AIT, the instructors utilize a number of instruments to enhance their lectures. Besides the standard textbooks and lecture slides, a number of assignments, a final project, a midterm, and a final exam are given to the students each semester. All course resources are accessible via the Institute's local intranets on an open source Web-based, course-ware application, called "moodle." Audiovisual content and all lecture slides are placed regularly on this site so that students have continuous access to this material. The site also supports e-forums where the registered students can discuss class material. Essentially, the only items not supported on the moodle installation are: 1) online exams and 2) VSEs.

The approach toward the creation of a virtual laboratory for this course originates from a general solution conceptualization for VSEs that comprise three distinct models: A Knowledge Model, a Didactic Model, and a Learner Model [5], [6]. The Knowledge Model consists of three, more or less expected, abstraction layers, namely the Learning Objects (LO) layer, the LO Meta-data layer used for indexing, etc., and the Ontology layer used to describe relationships about the LO meta-data. The Didactic Model, consists of two levels: the macro level at which one specifies the general structure of the whole learning experience, and the micro level which consists of a didactic approach
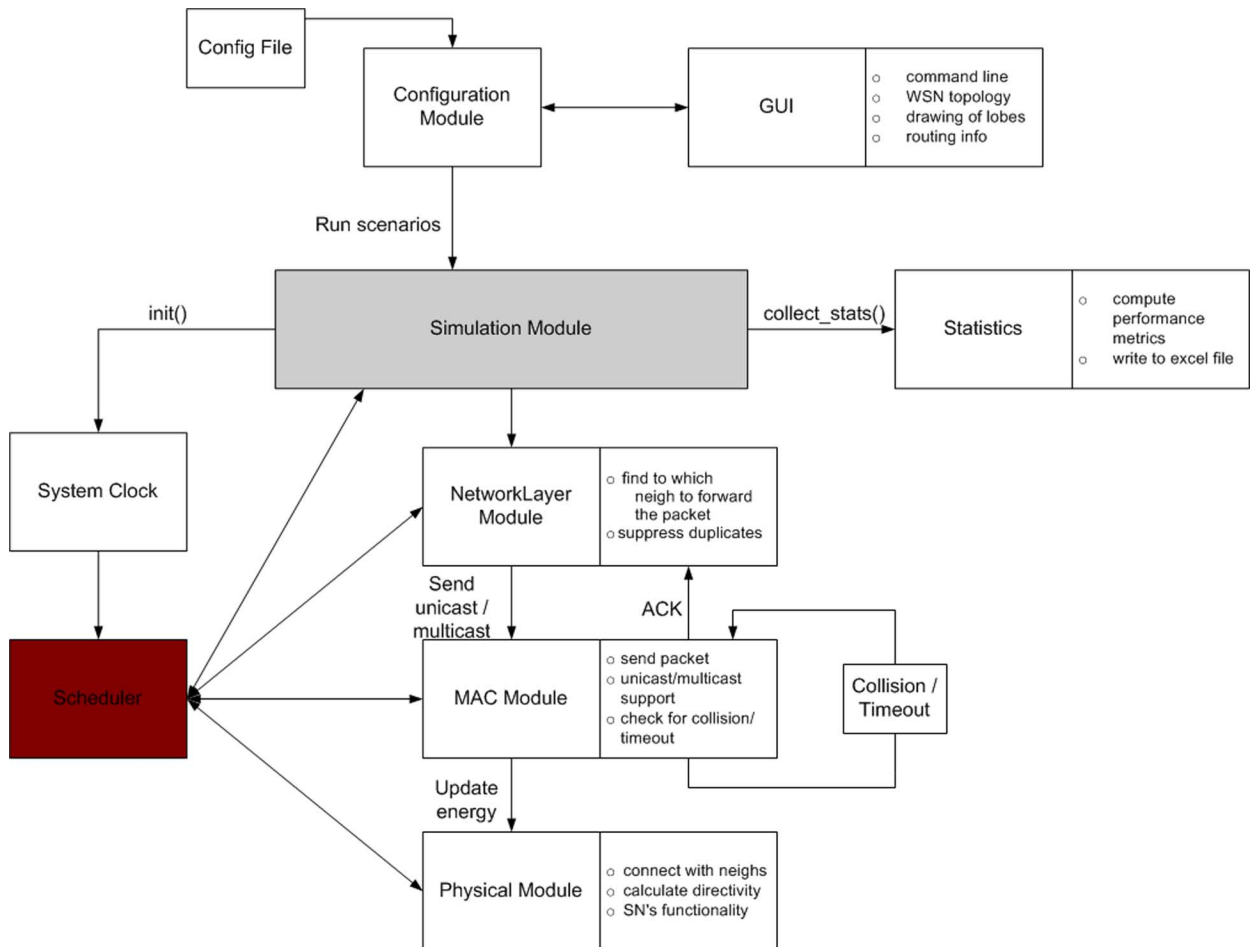
Fig. 1. SENSASIM Simulator Architecture.

and a didactic method. Finally, the Learner Model specifies a Context Profile, which eventually generates a generic contextualized ontology, and a Learner Profile that eventually generates a specific personalized ontology containing the mental structures of the team members' current knowledge about sensor networks and their interrelations. In this setting, the Learner's Profile is typically that of a graduate student with an excellent engineering background and analytical skills, a multiethnic background with a good command of the English language. Most of the students have a thorough knowledge of interacting with computers and using programming languages; however, they would not rank as most highly extroverted people, in general.

The theory of *Constructivist Learning* was used as a method for designing the laboratory scenarios. Knowledge construction is assisted by exploring experiments in a simulation of the real world. At the same time, experiential learning promotes a 'direct encounter' with the phenomena being studied rather than merely thinking about them [6], [9], [10].

By applying the above theories in the framework of the construction of a virtual laboratory for a deeply technical course such as Sensor Networks, a number of requirements was extracted.

- *The simulation setup must be adaptable to the level of knowledge of the team members*, e.g., for members with only basic knowledge of the MAC layer whose interests are in the network layer; or in the effects of wave propagation in sensor nets, the system must provide appropriate default settings for the MAC (or any other) layer. This adaptation allows the learners to focus their study on the aspects that most interest them (or on the aspects that they least understand).
- *The simulation must allow and even enforce team collaboration, especially for geographically separated teams*. This requirement has significant value on the effectiveness of team building, community building, and collaborative learning.
- *The simulation toolkit must allow for a central coordinator for each lab team*. This coordinator (usually a tutor) will be responsible for committing proposed parameter settings by the team members, and for starting and possibly interrupting a simulation run. Each team runs the simulation experiments in a cooperative but supervised manner.
- *Textual and audiovisual real-time communication must be supported by the virtual laboratory* at all times, especially during the simulation setup and execution time because it is crucial that students can talk and interact with each other to understand the experiment.
- *Visualization of the simulation runs* is of great importance toward a deeper, and easier, understanding of the experiments' results.
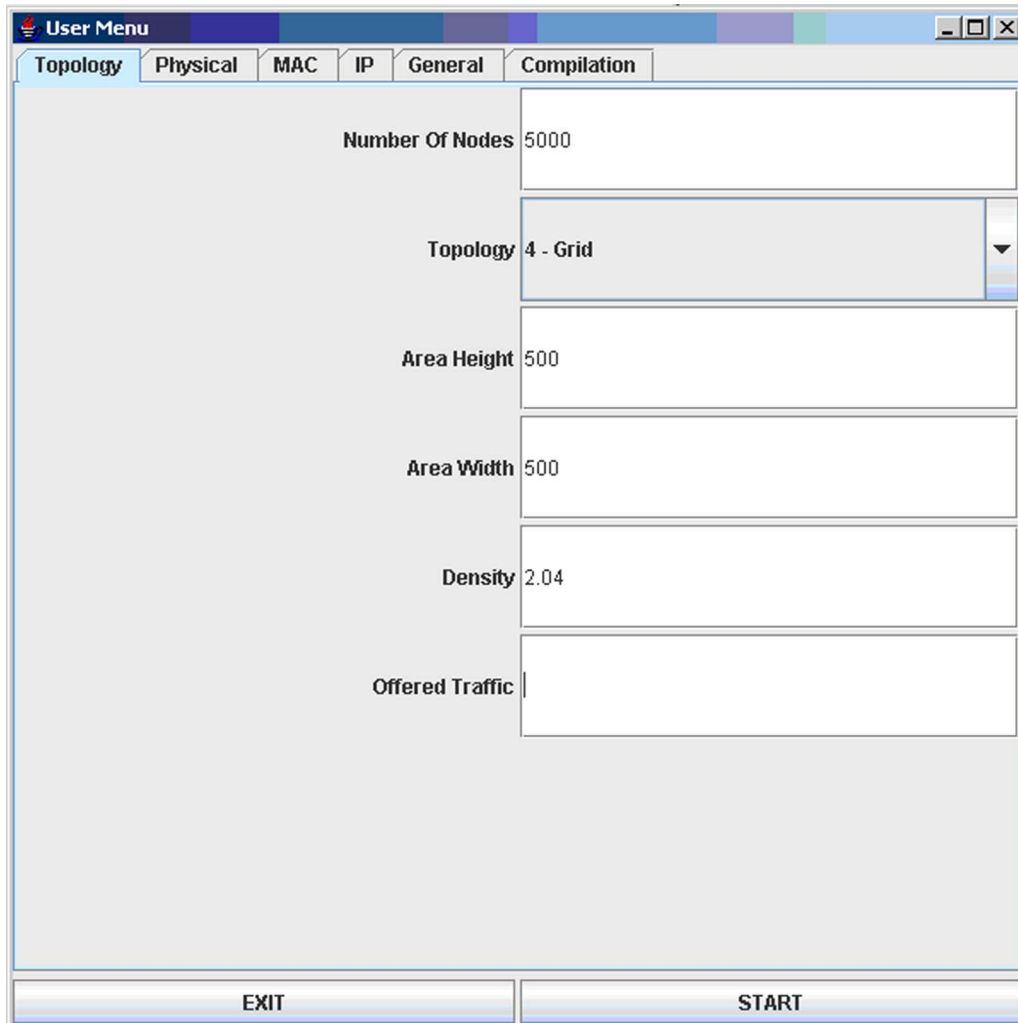
Fig. 2. Original SENSASIM desktop GUI.

C. Additional Application and User Interface Requirements

For a Web-based remote laboratory to become truly successful, communication and collaboration between the experimenters, whether they are students, researchers, or tutors and supervisors, must be supported so that no difference exists whether the participants are located in remote places or are all gathered together in the same physical space. Further, the experiments must allow for extensive interaction with the equipment, and in the particular case of VSEs that are simulations, they must allow for a "man-in-the-middle" approach to simulation runs [11]. The services to design should include voice and instant messaging interaction among the participants of a lab session.

III. SENSASIM ON THE GRID

A. The Original Desktop SENSASIM Application

The Broadband Wireless and Sensor Networks Research Group at AIT had already built the original SENSASIM application [1], an event-based simulator for wireless networks in the Java programming language using an efficient and robust object-oriented implementation methodology. As aforementioned, this simulator can be used for any *ad hoc*

wireless network, but the focus is on the more energy-limited and constrained sensor networks. The simulator supports the whole network protocol stack in varying degrees of granularity. Emphasis is given in the modeling of the parameters, functionality of the various communication layers, and the energy consumption of the nodes. To be able to support realistic scenarios of sensor networks having thousands of nodes, the tool does not take into account the operating system latencies of the nodes, thus providing underestimated results of the end-to-end latencies. This result is considered a minor caveat for most WSN applications in which the energy consumption and the lifetime of the nodes are the most critical metrics considered. The supported physical model, on which special emphasis has been placed, includes the use of directional antennas, enabling the switching of the antennas' lobes at will and the finding of the direction of arrival (DoA) of the signal. In the original implementation, the user is provided with a configuration file to set inputs, an excel output file holding all the statistics of the simulation, and an interactive GUI showing the flow of information across the sensor nodes. The simulation simulates one (or more) initial event that is sensed by a sensor node in the network, which then generates a message that has to be propagated to the "fusion node," or access point of the network.
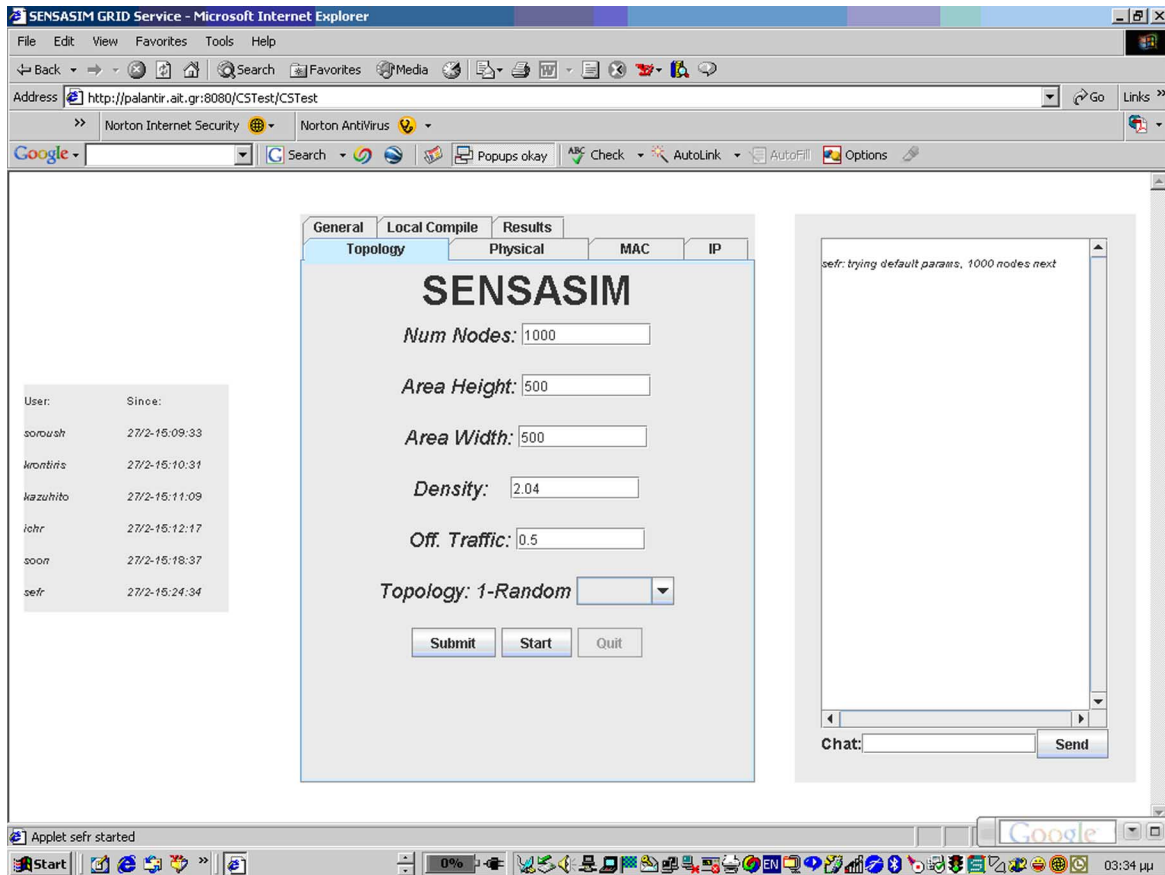
Fig. 3. Virtual lab sensor net simulation GUI.

The transmission of this message through specific paths in the network, and its implications on energy consumption, etc., is the objective of the simulation.

The architecture of the SENSASIM simulator is shown in Fig. 1.

Because of the need to simulate reasonably sized sensor networks with thousands of nodes, the simulation was designed as an event-driven simulation, and all the events, such as sending a packet or waiting for an acknowledgement, were modeled as independent threads which run concurrently in each simulation. Such threads are synchronized so that all events happen in chronological order. The mechanism to ensure this coordination is the scheduler [12].

Fig. 2 shows the GUI for the original desktop-based SENSASIM, supporting user configuration, written using the Java Swing package.

The original application allowed the general configuration of the simulation and the configuration of the topology, and the physical, MAC, and Network layers, but after the simulation is started, no further interaction is possible until the simulation is complete. Further, since it is a desktop application, the GUI is only controllable by the desktop user.

B. Collaborative Simulations Using GRID-Enhanced SENSASIM

The decision was made to port the SENSASIM simulator on a GRID environment that would allow the vision of remote laboratory experiments by geographically separated teams.

As a Grid middleware platform, the Globus Toolkit version 4 (GT4) was chosen [13], [14]. The toolkit is very popular for developing Grid services and implements the Web Service Resource Framework (WSRF) model for Grid infrastructure and architecture. The GT4 toolkit was installed from the open source distribution on three Linux servers running Suse 9.3; on two other Windows machines of the Wireless Networks laboratory in AIT, the WS-Core part of GT4 was also installed. The core SENSASIM engine was written as a WS-Core GRID service in Java using the *java2wsdl* (java-to-Web services definition language) package and the Java Annotations Toolkit, an open source tool made available from the Distributed Systems Group of the University of Marburg, Germany. A new user interface that allows multiple teams to interact in real-time with the GRID-based simulator service from a Web front-end was developed. The original SENSASIM engine's architecture was essentially unmodified and is accurately depicted in Fig. 1.

The particular Web-based GUI the authors settled on is very similar to the one developed in the original desktop application because an already installed user-base of the desktop application is at the AIT campus. The deviations from the original GUI include the following.

- A login screen which allows experimenters, including students and tutor/coordinator, to login to a new simulation instance that will be served by one of the available GRID nodes.
- A component that shows at all times on the participants' Web browsers of the logged-on members of the team.
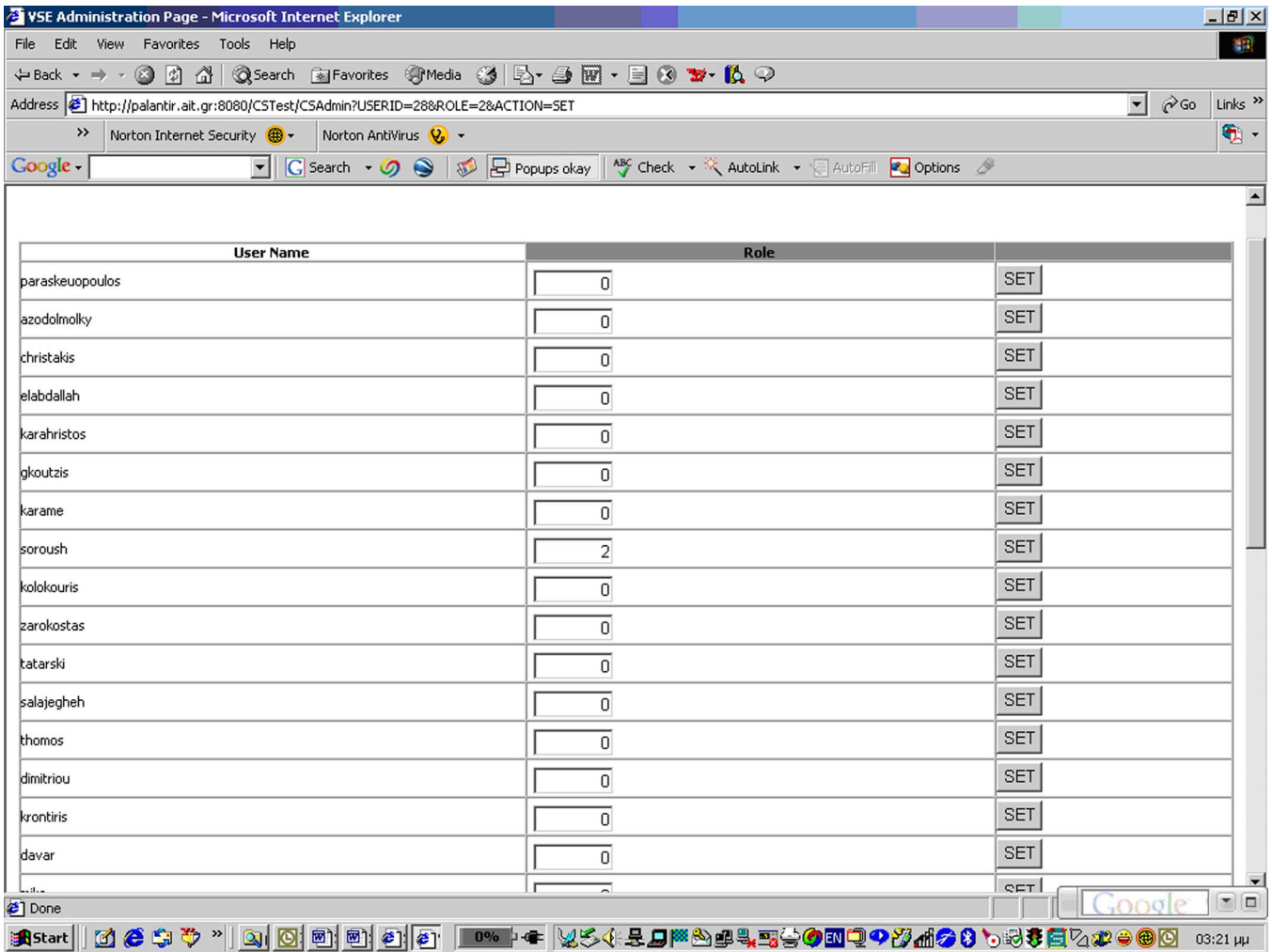
Fig. 4.   VSE participants control screen.

- A built-in chat component to allow instant messaging interaction.
- A quick enabling voice interaction, FlashMeeting (http://cnm.open.ac.uk/projects/flashmeeting) was used for its voice services. FlashMeeting also allows video conferencing, a feature not exercised strongly in the experiments.
- A control component that allows the administrator of the experiment to allow or deny in real-time access/modify/execute privileges for any participant. This control screen is, of course, only available to the administrator.

Fig. 3 shows the new Web front-end to the VSE on Wireless Sensor Network simulations.

On each participant's computer, another browser screen runs the flash-meeting session that provides audio conference support. For reasons that will be discussed in the Assessment section of this paper, the next version of the virtual lab will be more tightly integrated with FlashMeeting and BuddySpace [15].

Fig. 4 shows the control screen of the administrator which can be used to modify at run-time the privileges of any participant in the experiment. For example, a student may be allowed to modify the parameters of a simulation, and start its execution on the grid nodes of the virtual lab.

C. The Architecture of SENSASIM on the GRID

As mentioned, a grid cluster of three Linux nodes running GT4's "globus-start-container" and another cluster of two Windows nodes running the WS-Core component of GT4 were created. Each node is equipped with dual Pentium4 processors running at 3.2 GHz, 1 GB of RAM, and 120-GB 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 a front-end node as shown in Fig. 5 depicting the new system architecture.

There are three different kinds of nodes in this architecture.

1) Client nodes running the client browsers plus Flash-Meeting
2) A server running the Tomcat Servlet/Java Server Pages (JSP) container acting as the front-end server to which all clients connect
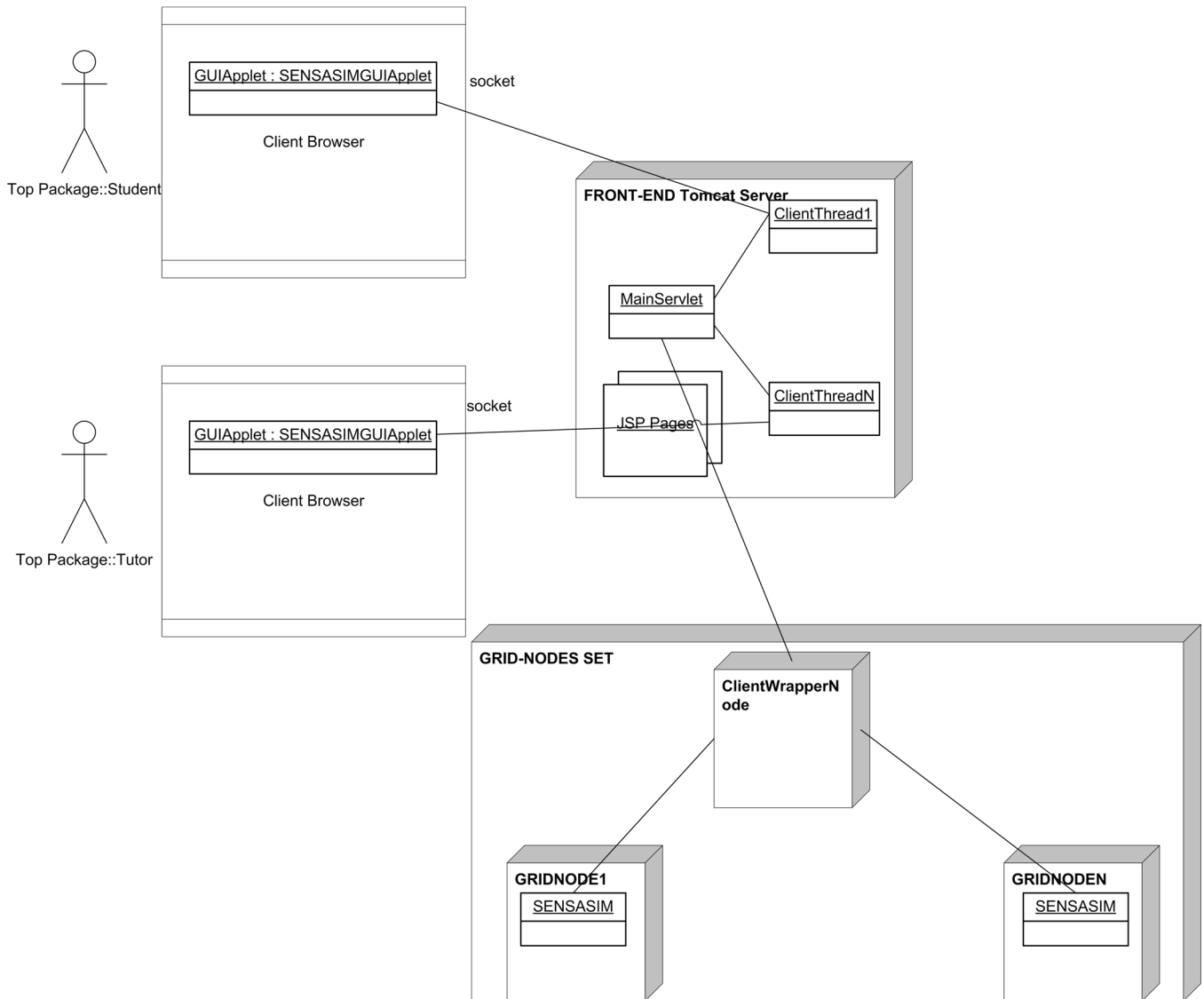
Fig. 5. Grid-based SENSASIM architecture.

3) Grid nodes running the Java WS-Core components of GT4, with the core SENSASIM simulator published as a grid service in Grid Web Services Definition Language (GWSDL) [13], [14].

The client Web browsers run three applets totally: one is the left applet that shows users currently logged in; the middle is a special applet that shows the new SENSASIM GUI; and the right applet which supports instant messaging (Fig. 3). The applets communicate directly with the front-end server, via sockets, and submit/process/render requests and results. Only the coordinator of a simulation team or someone who has been given the appropriate privileges via the control screen has the right to start a new simulation instance. The instance will start on the GRID node with the lightest load.

IV. GRID-BASED SIMULATIONS USER INTERACTION AND EXPERIENCE ASSESSMENT

The virtual lab prototype service can be used in two distinct contexts. The first context is purely educational, i.e.,

forming teams of students that participate in the "Sensor Networks" course, offering well thought-out, predefined virtual experiments with the simulator services. In this setting many simulation parameters are fixed by the coordinating tutors, and the student participants are allowed to modify the rest of the parameters. In the first experiments the focus was on experimenting with the antenna hardware characteristics, and the MAC layer, while keeping the other parameters fixed. In an exploratory first assessment phase, the prototype services were demonstrated to a body of 25 students at AIT, who were then asked to fill out a questionnaire. The students' backgrounds were mixed with students having electrical and computer engineering degrees ($\sim$60%) and the rest having computer science and informatics degrees ($\sim$40%). Overall, the students enjoyed the possibility of a virtual laboratory, and the freedom it provides from geographic constraints.

The communication element turns out to be, as expected, crucial to the overall success of the laboratory session and that of the collaboration process in particular. Integrated communications functionality within the virtual laboratory front-end
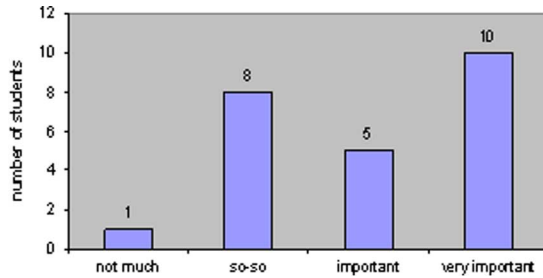
Fig. 6. How important verbal communication is during remote lab session.

Fig. 7. Preferences regarding virtual lab functionality.

is needed. The chart in Fig. 6 shows the student preferences regarding communications functionalities as depicted in the questionnaire.

As aforementioned, the current version of the simulation engine does not allow for on-the-fly modification of the simulation parameters while the simulation is running. Students clearly find this omission to be a shortcoming of the prototype and would like to see a more interactive version (60%). The interactivity students desire is not related to fast response times, but rather to the ability to interact with the experiment and modify the simulation while the simulation is still running. They also want to have real-time graphical interfaces showing the simulation progress. A modification of the current architecture [16] that allows authorized participants to interact with the experiment via a graphical interface is desired, in the form of disabling or enabling sensor nodes via simple point and click actions on a map of the sensor network while the simulation run is underway.

Since the grid aspects of the virtual laboratory related to GT4 are completely transparent to the students, they should not be in a position to evaluate these. However, the quick response times of the simulations were nicely received and kept student interest at high levels. Nevertheless, 60% stated that they would not care how long the simulations last as long as they get the results they want.

Overall, an overwhelming majority of students found remotely accessibly labs to be a great option and tool to complement a course. Fig. 7 shows the answers to a question regarding overall remote-virtual-lab preferences. The choices are shown as follows.

1) Traditional lab experiments with all students sitting in front of one desktop, one of them controlling the experiment.
2) Traditional lab experiments, with all students situated in the same lab, but each student allowed to modify parameters, etc.
3) Ability to connect to a lab session from home or elsewhere with Web front-end.
4) Ability to connect to lab session from anywhere with Web front-end while at the same time providing interaction functionality with voice and text.

Eighty-four percent of the students appreciated the virtual laboratory that performed sensor network simulations, provided the communications facilities allowed them to interact with each other while setting up or running the simulation.

To test the virtual lab's educational value in a truly geographically distributed context, *yet another experiment was set up in collaboration with CMU CyLab Japan, in Kobe, Japan.* Nine
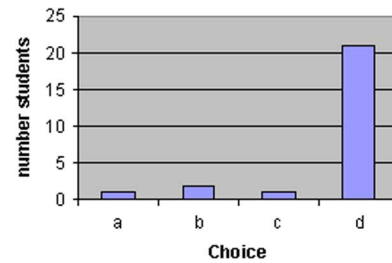
students from CyLab Japan (a whole class) participated in this collaborative virtual lab session, joined with seven of MSIN students in AIT. Tutors from both Japan and Greece were present to assist the run of the experiment. Before the virtual lab session, two tutors, one from each location, gave half-hour lectures on wireless sensor network material relevant to the issues that students would later tackle using the simulator and using the SENSASIM software itself. Immediately after the end of the virtual lab session, all students filled out a questionnaire to assess the impact of the lab. The overall results were positive, despite a few communication problems because of bandwidth limitations that were probably caused by heavy Internet traffic during the time that the experiment took place. The sensor network class students gave the contents of the VSE in a scale of 1 (very interesting) to 5 (not interesting at all) an average rating of 2.0, the middle value being of course 3.0. The communication tools employed overall were rated as 2.28 in a scale of 1 (appropriate) to 5 (not appropriate). The collaboration process was rated 2.71 in a scale of 1 (process worked very well) to 5 (process did not work at all). The lower rating can be attributed to the fact that because of the communication problems and delays, the voice-conference via FlashMeeting was problematic for a measurable part of the duration of the experiment and discouraged students from attempting to interact more during the session. Overall, on a scale of 1 (strongly agree) to 5 (strongly disagree) the students rated the system with a 2.4 on how well its components were integrated, giving room for more improvement. Some students specifically commented that they liked the collaborative features of the experiment. But another commented that without the ability to modify and execute simulation instances, "it is like watching a movie." For this reason, a new turn-taking based approach similar to the one implemented in FlashMeeting and BuddySpace [15] has been investigated, whereby each participant will await his or her turn, and when it comes, is free, subject to administrator approval, to set any values he or she wants for the experiment parameters and to run the experiment on the grid. Finally, the speed with which the simulation runs, because of the underlying grid infrastructure, was well received again. The experiments chosen were meant to demonstrate the effect of various protocols and the use of directional antennas on energy consumption. All experiments were completed in less than 3 min each, most in less than 1 min, giving the students an almost real-time response to the simulation.

In a second context, the virtual laboratory can facilitate graduate students and faculty performing research in wireless sensor networks to form research teams and study the effects of various

strategic decisions on sensor nets topologies, etc. Assessment of the value of this approach within a demanding research context is currently ongoing.

V. CONCLUSION AND FUTURE DIRECTIONS

This paper presents the design and implementation details of a first prototype of a virtual laboratory for a course in "Sensor Networks." The main simulation engine of SENSASIM, a Java-based simulator that has already been used for the study of *ad hoc* multihop wireless networks, is reused and published as a grid service on a local grid of two clusters of servers. A front-end server ensures the delivery of a Web-based GUI to the simulation that is close to the original interface and at the same time allows the smooth collaboration between the team members participating in a simulation run. Simulation runs start after team members have settled on the values of a simulation, and are submitted as a full job to the server that has the lightest load at the moment of submission. (More advanced policies that the Globus Toolkit and underlying systems, such as Condor-G, allow, will be studied in the next release of the software). The team members are currently allowed to communicate via an embedded chatting system and via FlashMeeting via voice. More integrated support for textual and voice communication will be available in the next release. The next version will also be more interactive while the simulation is running—something that requires changes in the simulation engine as well. The SENSASIM engine is also redesigned to take advantage of the parallel computing capabilities grid offers. The simulation distributes nodes in a rectangular area according to a topology policy. While in the main iteration loop, messages are sent between neighboring nodes in the simulated area. By decomposing this simulated area among a number of computing nodes participating in the grid and by minimizing communication overheads when messages are exchanged between the processing nodes and assigning each node to a particular subarea of the decomposition, one can speed-up the computation almost linearly with the number of nodes [17].

Overall, the students' assessment of the use of this virtual lab design was positive, and they believed that running the experiments shed more light into topics that had not surfaced in the lectures. Although the student body was not tested immediately after the experiment, they were able to answer questions corresponding to the topics covered by the VSE during their final exam, showing that the experiment met its objective to instruct students on the effect of communication layer protocols on the power consumption of networking nodes in WSN. The collaboration aspects of these experiments made the overall experience more interesting and fun. All components were running throughout the entire virtual lab session, except the five grid nodes which were utilized less than 10% of the total virtual lab time. Unfortunately, even though the components were running, the voice of the Greek instructor did not reach the students in Japan approximately 30% of the time for this particular experiment, probably because of bandwidth limitations (the students in AIT had no such communication problems). Therefore, an instructor wishing to enhance his/her course with such virtual laboratory experiments should consider that the more integrated

the various modes of communication are in a single system (text, data, voice, video conferencing [18]) the better the overall experience will be. As for building a cluster of GT4 nodes under the Linux or Windows Operating Systems (the latter allowing only the Java WS-Core component of GT4 to run) is a matter of simply following the installation instructions. Similarly publishing a Grid Service made available as a GAR file is trivial. Setting up a lab session currently takes approximately 15 min, including the time it takes to initialize the grid nodes and start the services on the front-end server. Some planning is required, since a FlashMeeting session must be booked in advance (usually a day earlier). Overall maintenance of the system currently costs very little (requiring a few minutes of attention by a single system administrator) however, if this grid extends beyond the institutional borders of AIT it would likely cost more both in terms of time needed to set up and log experimental sessions and in terms of the personnel needed. The GRID aspects of the work described in this paper will pay off more in a research-oriented environment, where very large scale simulations will be required to study large-scale phenomena (e.g., deployment of hundreds of thousands of sensor nodes in an area to monitor a particular physical phenomenon). In such cases also, the collaborative aspects of the work described here can be very useful.

As a last comment, a faster collaboration and communication environment for such simulation runs could be built by using Remote Desktop functionality on a number of Linux Servers (e.g., using the well-known Virtual Network Computer (VNC) Server from AT&T research) and combined with standard instant messaging tools such as the ones currently employed. Then students would be allowed to log-in in one of those servers dedicated to a specific team and view the remote desktop running the original SENSASIM application. This approach is neither very scalable, nor very valuable for a true sharing of available resources or "computing on demand" when a simulation need arises (remote desktop vendors always indicate that such tools are not designed for application sharing). However, this approach may be a good option in an environment where the student body is small, the simulation requirements are not too heavy, and the student participants in the experiment all have enough bandwidth for their connections (VNC transfers as bitmaps the whole desktop surface, and so requires broadband connectivity for a comfortable interaction experience).

REFERENCES

[1] G. Rayess, I. Tsirilakis, C. Kakogiannis, A. Kalis, A. G. Kanatas, and Ph. Constantinou, "Java-based simulator for wireless multi-hop networks using directional antennas 2006," in *Europ. Wireless Conf.*, Athens, Greece, Apr. 2–5, 2006.
[2] *ELeGI Project*, [Online]. Available: http://www.elegi.org
[3] J. Joseph and C. Fellenstein, *Grid Computing*. Upper Saddle River, NJ: Prentice-Hall, 2004.
[4] I. Foster and C. Kesselman, Eds., *The GRID2: Blueprint for a New Computing Infrastructure*. San Francisco, CA: Morgan-Kaufmann, 2003.
[5] S. Vigione, M. Desiderio, and G. Iovane, "ELeGI Deliverable D10: Preliminary version of didactical and knowledge representation models for virtual scientific experiments," ELeGI Rep. D10, Jan. 2005.
[6] ELeGI Consortium, "ELeGI Deliverable D18: Update version of didactical and knowledge representation models for VSEs," ELeGI Rep. D18, Jul. 2005.
[7] F. Zhao and L. Guibas, *Wireless Sensor Networks: An Information Processing Approach*. San Francisco, CA: Morgan Kaufmann, 2004.

[8] T. Dimitriou, I. Krontiris, F. Nikakis, and P. Spirakis, "SPEED: Scalable protocols for efficient event delivery in sensor networks," in *Proc. Networking 2004*, Athens, Greece, pp. 1300–1305.

[9] T. A. Mikropoulos, *Educational Software*. Athens, Greece: Klidarithmos, 2003, (in Greek).

[10] S. Henessy, D. Twigger, R. Driver, T. O'Shea, C. E. O'Milley, M. Byard, S. Draper, R. Hartley, R. Mohamed, and E. Scanlon, "A classroom intervention using a computer-augmented curriculum for mechanics," *Int. J. Scientif. Educ.*, vol. 17, no. 2, pp. 189–206, 1995.

[11] T. Utsumi, private communication on VSEs in the ELeGI Project.

[12] J. A. Miller, R. S. Nair, Z. Zhang, and H. Zhao, "JSIM: A java-based simulation and animation environment," in *Proc. 30th Annu. Simulation Symp.*, Apr. 7–9, 1997, pp. 31–42.

[13] V. Silva, *Grid Computing for Developers*. Rockland, MA: Charles River Media, 2005.

[14] B. Sotomayor and L. Childers, *Globus Toolkit 4: Programming Java Services*. San Francisco, CA: Morgan-Kaufmann, 2005.

[15] M. Eisenstadt, J. Komzak, and M. Dzbor, "Instant messaging+Maps = powerful collaboration tools for distance learning," in *Proc. TelEduc03*, Havana, Cuba, May 2003, pp. 1–11.

[16] K. Zajac, A. Tirado-Ramos, Z. Zhao, P. Sloot, and M. Bubak, "Grid services for HLA-based distributed simulation frameworks," in *Proc. 1st Euro. Across Grids Conf.*, Santiago de Compostela, Spain, Feb. 2003, pp. 147–154, LNCS vol. 2970, Springer 2004.

[17] I. T. Christou and R. R. Meyer, "Decomposition algorithms for communication minimization in parallel computing," in *Non-Linear Assignment Problems Theory and Practice*, P. M. Pardalos and L. Pitsoulis, Eds. Boston, MA: Kluwer Academic, 2000.

[18] D. C. Sicker, A. Kulkani, A. Charali, and M. Fajandar, "A federated model for secure web-based videoconferencing," in *Proc. Int. Conf. Inf. Technol.: Comput. Commun.*, 2003, pp. 396–400.

Sofoklis Efreemidis received the M.Sc. and Ph.D. degrees in computer science (minor mathematics) from Cornell University, Ithaca, NY, in 1992 and 1994, respectively, and a Professional Diploma in electrical engineering in 1986.

His academic research work focused on theoretical aspects of programming languages and systems, program transformations, semantics of programming languages, programming logics and formal verification of program correctness, methodologies for program development, and embedding of attribute grammars into high level functional programming languages. In October 1996, he joined INTRACOM S.A., while from February 1999 to September 2003, he was responsible for the company's representation office in Brussels. He has worked in various European research projects in the areas of software technologies, focusing on modeling techniques, methodologies, middleware technologies, and system architectures. In September 2003, he joined Athens Information Technology, Paiania, Greece, where he currently is an Assistant Professor. He works in the area of software systems and technologies, distributed and dependable systems, and semantic Web services. He is the author of several scientific papers that appeared in journals and conferences.

Dr. Efremidis is a member of the ACM and of the Technical Chamber of Greece.

Thanassis Tiropanis (M'96) received the Professional Diploma in computer engineering and informatics from the University of Patras, Greece, in 1993, and the Ph.D. degree from the University of London, U.K., in 2000.

He is an Assistant Professor and Co-Head of the Software Engineering and Web Applications Group, Athens Information Technology (AIT), Paiania, Greece—a Centre of Excellence for Research and Graduate Education in collaboration with Carnegie Mellon. He is also the representative for AIT in the World Wide Web Consortium (W3C). Prior to joining AIT, he was Project Manager with Atkosof SA. Earlier, he was Research Fellow with University College London, U.K.. Over the last 10 years, he has participated as a researcher and served as technical coordinator or project manager in a number of research projects related to the exploitation of Grid, Web, and Communication Technologies in different application contexts. His current interests include Web services, grid services, semantic grid, e-commerce, and e-Learning. He is a member of the programme committee in a number of international conferences in these areas.

Dr. Tiropanis is member of the British Computer Society (BCS) and the Technical Chamber of Greece.

Ioannis T. Christou (M'02) received the Professional Diploma in electrical engineering from the National Technical University of Athens, Greece, in 1991, and the M.Sc. and Ph.D. degrees in 1993 and 1996, respectively, both in computer sciences from the University of Wisconsin at Madison, Madison, WI.

He has been a Senior Developer with Delta Technology, Inc., an Area Leader with the Data and Knowledge Engineering Department with Intracom S.A. Development Programmes Department, and a Member of Technical Staff with Lucent Technologies Bell Labs. He has also been an adjunct Assistant Professor with the Department of Computer Engineering and Informatics, University of Patras, Greece. He is currently an Assistant Professor with Athens Information Technology, Paiania, Greece. He has published numerous articles in scientific journals and peer-reviewed conferences. His current research interests include Distributed Information Retrieval, Data Mining and Web Mining, and Parallel and Distributed Computing, including Grid and P2P Computing.

Dr. Christou is a member of the ACM and of the Technical Chamber of Greece.

Antonis Kalis (M'00) received the Diploma degree in electrical engineering in 1997 and the Ph.D. degree with honors in June 2002, both from the University of Patras, Greece.

He is an Assistant Professor with Athens Information Technology (AIT), Paiania, Greece, and an adjunct Assistant Professor with Carnegie Mellon University, Pittsburgh, PA. He joined the Lab of Electromagnetics, University of Patras, participating in various R&D projects for the Greek Government and the European Union, as a member of the research staff. In 2000, he worked as a Research Engineer and an Assistant Research Unit Manager the Computer Technology Institute. His research interests are in the areas of radio communications, antenna design, and wireless networks. He has numerous journal and conference publications, a US patent.

Dr. Kalis received the 2000 Chester Sall award of the Consumer Electronics Society in the aforementioned research areas. He is a member of the Technical Chamber of Greece and AFCEA.