# Survey of Remote Laboratories Using Service Oriented Architectures

doi:10.3991/ijoe.v5i1.781

D. Ponta, A. M. Scapolla and P. Buschiazzo
University of Genoa, Genoa, Italy

***Abstract*—Remote access to real laboratories can enhance traditional educational paths with practical experiences. However, most of the existing remote laboratories cannot communicate with each other and they are not yet completely integrated with common educational platforms such as Learning Management Systems. These problems could be tackled by offering to end users remote experiments as distributed services using web service technology. The paper examines the architectures of remote laboratories developed by three institutions: DIBE ISILab (Internet Shared Instrumentation Laboratory), HPI DCL (Distributed Control Laboratory) and MIT iLab. Their front-end services are compared and discussed. The paper details how end-user applications interact with these remote labs and reports on the results of a preliminary test of interoperability between remote labs, providing a hint on how different remote experiments can be shared between different institutions.**

***Index Terms*—remote laboratories, distance learning, service oriented architectures, web services.**

## I. INTRODUCTION

The evolution of information and communication technologies has had a profound impact on educational and professional laboratories. On one hand, with the development of software simulation and design techniques, traditional hardware laboratories have been often replaced by software environments where components and systems are simulated. The diffusion of Internet has allowed, on the other hand, the set-up of remotely controlled real laboratories.

Today, Internet-controlled remote laboratories are available in a quite large and growing number of educational institutions and are establishing themselves in the mainstream of educational tools and practices. While there are still several limitations to their use, they offer, though, a good potential for enhancing lectures with practical activities, without the complex logistics and costs of traditional labs. And, of course, they are ideally suited for distance learning activities.

Learning Management Systems (LMS), nowadays widely adopted in education, are sets of software tools, generally web-based, designed to manage users, roles, courses and facilities. The use of remote laboratories within a LMS can integrate experiments with learning contents. Moreover, LMSs provide other useful features, such as discussion forums, questionnaires, report preparation and delivery tools, user statistics and feedback. Also, remote labs have the potential to be matched with, and used as resources by emerging educational tools that are innovating contemporary teaching and learning methodologies, emphasizing knowledge construction through social interaction and sharing of common working spaces [1]. At the moment, despite the success and diffusion of remote laboratories, most of the existing implementations, especially the ones relying on technologies offered by vendors, are still complex and closed systems. They have their own proprietary user and security policies, and demand strict requirements on the client side. The issue of scalability, i.e. the increase of the number of devices and experiments available remotely, is seldom addressed satisfactorily. It is, therefore, difficult to see the laboratory as a component of the educational path. In our view and practice, we expect that teachers do not use experiments as stand-alone events but integrate theoretical contents with practice and take advantage of the access to composite experimental set-up from diverse providers without any constraints. Students should therefore be able to access this composite experimental set-up like any other learning content.

From this point of view, existing remote laboratories, built as monolithic systems, do not offer sufficient flexibility. In our opinion, they should evolve toward new architectural designs, support the exchange of control between interoperating systems and deal with remote experiments as distributed services.

The paper explores the use of service oriented techniques applied to remote laboratories and reports on the experience gained while working on the European project Vet-Trend.

## II. SERVICE ORIENTED LABORATORIES INFRASTRUCTURE

Service oriented architectures for remote laboratories have been studied by the scientific community for a few years [2, 3]. In particular, the authors have investigated and proposed service oriented middleware to access heterogeneous remote laboratory equipment [4, 5], to control networks of small devices [6], and to extend the GRID [7] paradigm to the management of cooperative instruments and services.

The partnership in the project VET-TREND – “Valorisation of an Experiment-based Training System through a Transnational Educational Network Development” (LEONARDO DA VINCI, Community Vocational Training Action Programme, RO / 06 / B / F / NT175014) has provided encouragement and support to this research activity. VET-TREND main objective is the creation of a transnational network in the domain of virtual and remote experiments for e-learning, with the aim of expanding and valorising the existing installations, and integrating and optimizing the available products and

the new ones created in the framework of the project itself. The final target is the adoption of distance learning practices in applicative domains, where experiment based training is of relevant importance.

The project partners had a consolidated knowledge, from the experience of previous projects, in the development and test of virtual and remote laboratories, and they cooperated to build a transnational network that benefits existing skills, while taking other significant initiatives.

A network of laboratories sharing remote instrumentation and practices needs the definition of common ways of publishing the resources of each node, of running them and of establishing security policies.

The project included a phase of analysis and comparison of the solutions already adopted by the partners. The service based implementations of the University of Postdam and the University of Genoa looked as the most advanced solutions on the basis of the their functionalities. Also, the analysis took into consideration, as a reference, a laboratory outside the consortium, the iLab from MIT [8]. In the following paragraphs, short descriptions of the three labs highlight their general features.

Distributed Control Lab (DCL) is hosted at the Hasso-Plattner-Institute for IT-Systems-Engineering of the University of Potsdam, by the "Operating Systems & Middleware Group". The laboratory infrastructure is used in bachelor courses of computer science to teach the usage of programming languages such as C and C# and advanced topics in the area of embedded real-time systems in master-level courses. In particular, within the master program, embedded operating systems and special programming languages for real-time applications are investigated. In the lab, students can submit source code, which is then compiled and run to control a variety of physical devices, such as a Foucault's pendulum, mobile robots and the "higher striker" experiment, a competition among students for the implementation of the most efficient real-time program [9].

ISILab (Internet Shared Instrumentation Laboratory) [10], developed at the University of Genoa, is currently used to deliver online access to experiments on electronics for the benefit of a few engineering courses. It allows practicing with electronic instruments and measurement methods, executing real experiments of scalable complexity on analogical and digital circuits. The experiments deal with basic electronic measurements, such as delays in digital circuits or the gain and the distortion of amplifiers, and use devices such as waveform generators and oscilloscopes.

MIT researchers have developed the iLab architecture to provide online access to laboratories focused on several engineering disciplines. The iLab architecture is independent from the target application domain and uses reliable, generic services to give remote access to the lab equipments, through a web browser. The iLab provides several different experimental environments, such as the Heat-Transfer, Polymer and the Microelectronics llabs. The Heat Transfer experiments are related to thermal processes. The Polymer Lab enables users to access polymer crystallization experiments, using optical microscopy. Finally, the Microelectronics lab is focused on devices' characterization: users can measure the current/voltage characteristics of diodes, transistors and other components.

## III. SURVEY OF LABORATORIES' ARCHITECTURES

In order to examine in depth how the service oriented approach has been adopted by the different labs, the authors prepared a template document to be completed by the three institutions. This document aimed at collecting information about the lab servers' architectures and the interactions, mediated by web service interfaces, established between each remote laboratory and its client applications. The index of this document is reported in Table I.

TABLE I.
INDEX OF THE DOCUMENT USED FOR THE SURVEY

| | |
|---|---|
| **1. General Information** | |
| | *Laboratory Name* |
| | Description |
| | Education Domain |
| | Location |
| | URL/ Home page |
| **2. List of available experiments** | |
| | Type of the experiment (remote experiment / remote simulation) |
| | Experiment "Reservation" |
| | Experiment type (interactive or batch) |
| **3. General technical information** | |
| | Operating system |
| | Software developing environment |
| | Application server |
| | Programming Languages used |
| | Web service technologies / framework used |
| **4. Web service interface description** | |
| | Description of the front-end web service interfaces |
| | Details on each operation structure |
| | Data structures used by each web service |
| | How to use the WS interfaces (sequence diagram, source code examples) |
| **5. Client application description** | |
| **6. Guidelines to create a new client application** | |

The document has been completed by the VET-TREND partners for their own laboratories and compiled for iLab on the base of the public documentation available via WEB [11].

The answers to the section 3 of the survey have shown a strong degree of similarity among the technical solutions employed by the different developers. Microsoft Windows and .NET framework are the common choice for the three labs. The web service paradigm, instead, has been applied in different ways. This was not unexpected, as the remote labs are complex, distributed systems and the usage of standard technologies, protocols and languages leaves space for diverse implementations.

Just to give an example, ISILab uses web services for all the system components: to expose the functionalities of the elements of the measurement chain (instruments and other devices), to activate sequences of these services, to manage parallel work sessions, to check the availability of the instruments and to book them for the time required by the measurement process [4].

The architecture of MIT iLab presents a software infrastructure that adds, as a further layer, service based functionalities to preexisting remote laboratories. Moreover, iLab doesn't use a fine grained set of web services, but it concentrates all the operations in a main

service that checks users' authorization, contacts a particular lab server and waits to retrieve the results, once the experiment is completed.

We must keep in mind that our target is to analyze how the service oriented approach can facilitate the integration of lab experiments into educational processes and the communication and interoperability of lab servers with other tools and systems, such as the LMSs quoted before.

As a consequence, the fact that there are different lab server implementations does not represent an obstacle. What we are interested in, i.e. the focus of this survey, is the analysis of the services/interfaces directly controlled by the client applications. In other words, we want to highlight the set of services that must be available to the end user applications in order to run the remote experiments offered by the labs.

## IV. ISILAB, DCL AND ILAB FRONT-END SERVICES

The present section summarizes the results from section 4 of the survey documents for the ISILab, DCL and iLab laboratories. Their front-end service interfaces are detailed in Fig. 1, Fig. 2 and Fig. 3, respectively.

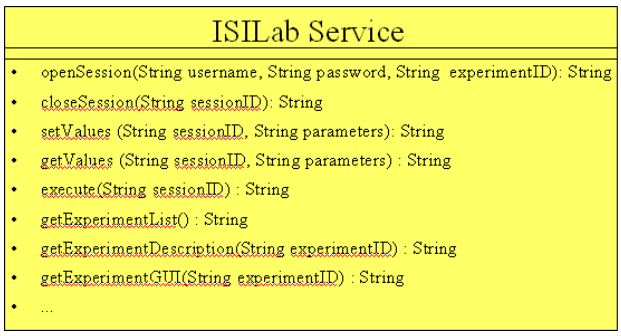


Figure 1. ISILab Service interface

ISILab can be controlled through a single web service (see Fig. 1). ISILab supports several different experiments, and client applications can use the `getExperimentList()` operation to retrieve the list of the available experiments. In ISILab, the interactive and concurrent access to the laboratory is provided by an internal session management mechanism. In order to interact with the instrumentation, client applications must choose one experiment and create a new measurement session, invoking the `openSession()` operation, which returns a session ID as a string. The measurement session contains a set of configuration parameters, and client applications can update or retrieve their values using the `setValues()` and `getValues()` operations.. The latter use, as input and output parameters, only XML documents, serialized as strings. This choice allows ISILab to use different data structures, according to the specific experiment used. After a client application has set the configuration parameters, the `execute()` operation must be invoked and the server sends the configuration parameters to the instruments and retrieves the measurement results.

The ISILab service also exposes the `getExperimentDescription()` and `getExperimentGUI()` operations. Given a specific experiment ID, `getExperimentDescription()` returns metadata encoded as XML serialized string, which contains general information on the experiment and link to images such as the circuit schematics. The `getExperimentGUI()` operation returns an XML representation of the remote instruments in the ISILab workbench. The client application can use this XML document to build interactive user interfaces that represents the remote instruments.

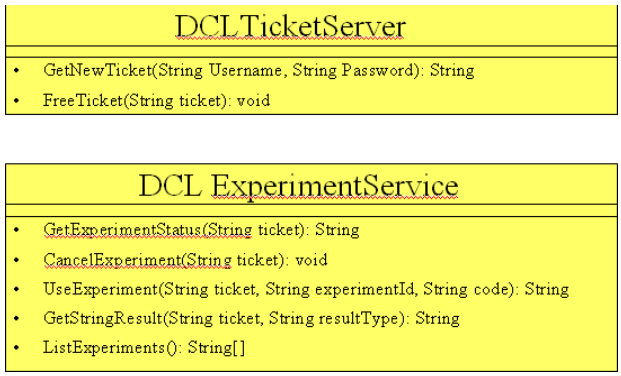


Figure 2. DCL Service interfaces

The DCL laboratory interface is dictated mainly by its application field. DCL allows students to control real time embedded devices by submitting C/C# source code and scheduling the execution of batch jobs. Each job's code is compiled and sent to the target embedded systems. The user can verify the execution status of the experiment and retrieve the results.

DCL uses a dedicated service for user authentication: the DCL TicketServer. Client applications call the `GetNewTicket()` operation to obtain a valid ticket and access the remote experiments provided by the DCL ExperimentService,the web service used by client applications to control the remote laboratory. With the `ListExperiments()` operation the client can list the available experiments and choose one of them. The `UseExperiment()` operation allows a client to submit batch jobs to the laboratory infrastructure. This operation accepts as plain strings parameters the ticket, the experiment ID and the source code. Once submitted, the job execution can be checked with the `GetExperimentStatus()`, which polls for the execution status. When the experiment successfully ends, the client application retrieves the result (encoded as a string) invoking `GetExperimentResult()`.

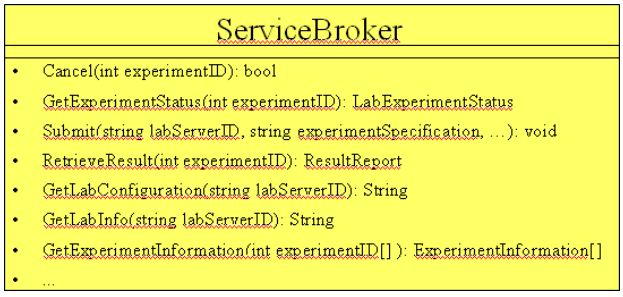


Figure 3. ServiceBroker is the front-end web service in iLab

The front-end web service of iLab is the ServiceBroker, very similar to the DCL interface. As DCL does, it uses the `Submit()`, `GetExperimentStatus()` and `RetrieveResult()` operations to execute batch jobs.

The session management is built through web cookies exchange: the measurement session is created by the web portal that contains the Service Broker after the user logs in. The Service Broker supports a wide range of operations dedicated to experiment and remote laboratory description: some examples are the `GetLabInfo()`, `GetLabConfiguration()` and `GetExperimentInformation()` operations.

## V. FRONT-END SERVICES COMPARISON

The following section points out differences and similarities among the three labs. ISILab interface is optimized for an interactive and concurrent access to the remote laboratory, while both DCL and iLab use a batch architecture approach. The most important feature, common to the three labs, is the modeling of experiments as services: all of them collect in a web service interface all the functionalities exposed by the lab itself. Moreover, they need to structure measurements as work sessions and they show a similar behavior in implementing the session mechanism: clients, once authenticated, create new measurement sessions that are identified by users and the operations they want to perform. The measurement session keeps track of its status and stores data sent or received from the instruments. All the remote laboratories offer operations specifically dedicated to session management in their front-end service interface:

- Open/close a measurement session
- Retrieve information on measurement status
- Send/retrieve data from the instruments

The last feature must be further detailed. In fact, even if each remote laboratory implements its own mechanism for data exchange with the instrumentation, all the lab middlewares accept data as opaque objects, typically plain strings or serialized XML. So the same interfaces may be applied to different target applications.

It should be noted that web service technology is associated with stateless mechanisms, a concept directly taken from the HTTP protocol. As plain web service technology doesn't provide a session management mechanism, the measurement session is manually implemented by each remote laboratory developer. Both in ISILab and DCL infrastructures the measurement session is based on the Web Service Resource Framework (WSRF) [12]. The WSRF specification adds the concept of Resource to plain web services: while a web service remains stateless, the state is stored in a separate entity, called Resource. As the state is completely separated from the service interface, a web service that supports WSRF may have several different Resources and each of them can store a different measurement session. WSRF is a stable and well known framework, especially in Grid infrastructures [13] and both ISILab and DCL remote laboratories use it internally.

Web service technologies have been quickly evolving in the last few years and many different specifications were developed (WSRF, WS-Notification, WS-Resource Properties, etc...). Such set of web service specifications is often referred as WS-* and it isn't completely supported by all the developing environments used today. All the laboratories described in this article hide the WS-* specifications internally and provide a WS-I compliant front-end service interface. The WS-I Basic Profile [14] defines a stable subset of web service specifications and provides high interoperability between different Service Oriented architecture. This choice allows developers to use different technologies to implement new client applications.

In addition to the basic measurement management, all the laboratories provide additional operations, as the ones, for example, listed in Table II.

TABLE II.
ADDITIONAL OPERATIONS

| | |
|---|---|
| **ISILab** | |
| | getExperimentList() |
| | Description |
| | getExperimentDescription() |
| | getExperimentGUI() |
| **DCL** | |
| | ListExperiments() |
| **iLab** | |
| | GetLabInfo() |
| | GetExperimentInformation() |

Client applications use these operations to retrieve the available experiments on each remote laboratory, their descriptions and other data (such as the GUI description in ISILab). Data formats depend on the architecture: as an example, while ISILab uses XML as encoding format, DCL and iLab prefer plain text and URL links to existing web pages.

Concerning authentication and authorization policies, the labs show different approaches. ISILab is available for every Web user, without any authentication mechanism. However, if necessary, there is the possibility to add username and password authentication. DCL infrastructure uses a basic authentication system with username and password.

iLab has a strong authentication procedure.. The iLab Service Broker has an internal user's database and manages user access authorization locally. The Service Broker acts as a proxy: when a user tries to access a lab, the Service Broker authenticates itself to the remote laboratory and mediates the communication between the client and the lab.

Thanks to the detailed description of the front-end web service interfaces, the remote experiments provided by the three laboratories can be shared with other institutions. In fact, choosing a WS-I compliant service interface, new client applications can be created using different technologies, such as Java, LabVIEW or Flash environments. In addition, common features among the remote laboratories provide general guidelines to implement new client applications.

## VI. INTEROPERABILITY BETWEEN REMOTE LABS: A PRELIMINARY TEST

Structuring remote laboratory functions as a set of services has the major advantage of allowing the sharing of the physical experimental setup, while leaving the possibility of customizing the client application interface. The same real experiment, for instance, can be exploited in several ways and at different pedagogical levels, with different graphical interfaces for instrumentation control and display of measurement results. Experiments from

different laboratories can be shared by assembling them together in a pedagogical path.

As an example, the current ISILab portal (see Fig. 4) allows users to choose and run an experiment from a list .The user interface contains both a description of the selected circuit and a simple representation of the remotely controlled instruments: an oscilloscope and a wave generator. The client can use simple text fields to supply configurations parameters and can visualize output waveforms through a screen dump of the remotely controlled oscilloscope data.

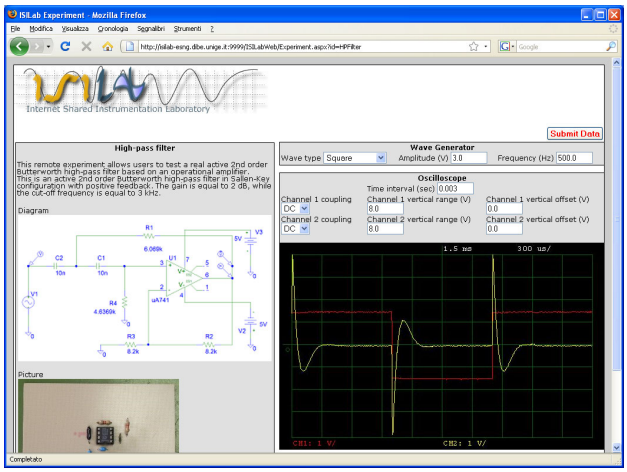


Figure 4. ISILab user interface.

Fig. 5 shows an alternative usage of the lab: a Moodle [15] course, on the background, calls ISILab services to show the front panel of the oscilloscope controlling an experiment inserted in the learning path. The oscilloscope panel is implemented as a java applet and provides a more detailed representation of the instrument, allowing to adjust with more accuracy the waveforms acquired. This is a good example of the integration of the laboratory with the LMS. An experiment can be called when needed, with the most appropriate client interface.

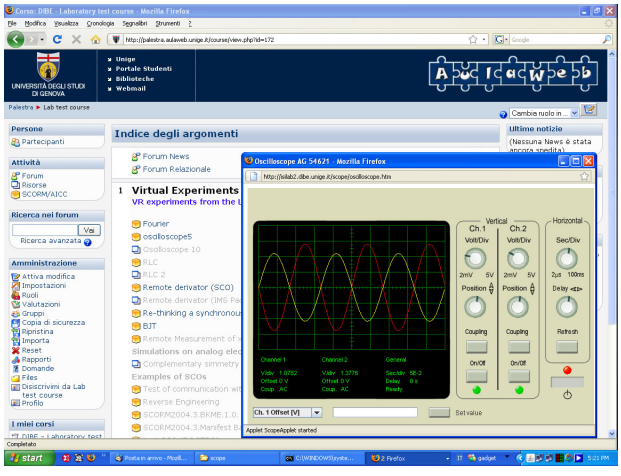


Figure 5. ISILab experiment integrated in a Moodle course.

In the Vet-Trend project, the sharing of experiments from ISILab and DCL has been tested, incorporating them in DIBE and HPI lab portals, without altering the portals' look and feel.

On the basis of the lab services specifications (see section IV) DIBE and HPI created two demo client applications, for implementing a cross access to ISILab and DCL. The HPI demo application uses the ISILab service to control a single remote experiment on ISILab workbench. The DIBE demo application uses DCL services to control the Lego Simulator, a remote experiment which simulates Lego Mindstorms mobile robots. End users can submit C source code to simulate the robot behavior and view the simulation result in a Flash animation, which displays the robot path.

The creation of such client applications could have been a hard job if we had to develop them from scratch. Since both demo applications are .NET web applications, the use of pre-existing source code enhanced the integration process.

The successful integration allows an easy access to each other's laboratory, even if the service interfaces used by the labs are different. This positive result encouraged both DIBE and HPI teams to study a stronger solution for the ISILab and DCL infrastructures integration.

## VII. CONCLUSIONS

This paper describes three service oriented remote laboratories and points out differences and similarities among them. The advantages of structuring the labs as service providers come out clearly from the work. The authors tested and proved the possibility of sharing remote experiments between different institutions. In this process they gained the conviction that the approach deserves to be pursued and expanded to form a large framework that enables students/users to access experiments all over the world via Internet.

To improve lab integration, discovery mechanisms such as UDDI [16] offer a good potential for the future. They allow to dynamically discovering registered remote laboratories, to list the available experiments and to retrieve information on the lab providers.

## REFERENCES

[1] H. Markkanen , M. Holi, L. Benmergui , M. Bauters, C. Richter, "The Knowledge Practices Environment: a Virtual Environment for Collaborative Knowledge Creation and Work around Shared Artefacts", in Proceedings of World Conference on Educational Multimedia, Hypermedia and Telecommunications 2008 (pp. 5035-5040). Chesapeake, VA: AACE, 2008.

[2] D. Dagger, A. O'Connor, S.Lawless, E.Walsh, V.P. Wade, "Service-Oriented E-Learning Platforms", IEEE Internet Computing, Volume 11, Number 3, May/June 2007. (doi:10.1109/MIC.2007.70)

[3] Y. Yan, Y. Liang, X. Du, H.S. Hassane, A. Ghorbani, "Putting labs on-line with Web services", IEEE IT Professional, Volume 8, Issue 2, March-April 2006.

[4] A. Bagnasco, A. Boccardo, A. Buschiazzo, P. Poggi, A. Scapolla, A.M., "A Service-Oriented Educational Laboratory for Electronics", Accepted for future publication in IEEE Transactions on Industrial Electronics: Digital Object Identifier: 10.1109/TIE.2008.2002729.

[5] A. Bagnasco, M. Scapolla, "A Grid of Remote Laboratories for Teaching Electronics", Frontiers in Artificial Intelligence and Applications - Towards the Learning Grid Advances in Human Learning Services, Volume 127,November 2005, 252 pp., ISBN: 1-58603-534-7.

[6] A. Bagnasco, D. Cipolla, D. Occhipinti, A. Preziosi, A.M. Scapolla, "Application of web services to heterogeneous networks of small devices", WSEAS Transactions on Information Science and Application, Issue 5, Volume 3, May 2006, ISSN 1790-0832.

[7] A. Bagnasco, D. Cipolla, A. Poggi, A. M. Scapolla, "Service-oriented architectures for distributed cooperative instrumentation

Grids", in "Grid Enabled Remote Instrumentation", edited by Franco Davoli, Norbert Meyer, Roberto Pugliese, Sandro Zappatore, Springer, in press.

[8] V.J. Harward, J.A. Del Alamo, et al., "The iLab Shared Architecture: A Web Services Infrastructure to Build Communities of Internet Accessible Laboratories", Proceedings of the IEEE, Vol. 96, No. 6, 2008. (doi:10.1109/JPROC.2008.921607)

[9] P. Tröger , A. Rasche, F. Feinbube, R. Wierschke, "SOA Meets Robots - A Service-Based Software Infrastructure for Remote Laboratories", International Journal of Online Engineering, Vol. 4, No. 2, 2008.

[10] A. Bagnasco, M. Chirico, A.M. Scapolla , "A New and Open Model to Share Laboratories in the Internet", IEEE Transaction on Instrumentation and Measurement, June 2005, ISSN 0018-9456, Volume 54, pp 1111-1117.

[11] MIT iCampus: iLabs Architecture, http://icampus.mit.edu/iLabs/architecture/.

[12] Web Services Resource Framework Technical Committee, http://www.oasis-open.org/committees/wsrf/.

[13] I. Foster, K. Czajkowski, D.E. Ferguson, J. Frey, S. Graham, T. Maguire, D. Snelling, S. Tuecke, "Modeling and Managing State in Distributed Systems: The Role of OGSI and WSRF", Proceedings of the IEEE, Volume 93, Issue 3, March 2005, 604 – 612 pp, Digital Object Identifier 10.1109/JPROC.2004.842766.

[14] Web Services Interoperability Organization, http://www.ws-i.org/.

[15] Moodle homepage, http://moodle.org/.

[16] Universal Description Discovery and Integration Spec. Technical Committee, http://www.uddi.org/.

## AUTHORS

**D. Ponta** is with the Biophysical and Electronic Engineering Department, University of Genoa, Genoa, Italy (e-mail: ponta@ unige.it).

**A. M. Scapolla** is with the Biophysical and Electronic Engineering Department, University of Genoa, Genoa, Italy (e-mail: scapolla@ unige.it).

**P. Buschiazzo** is with the Biophysical and Electronic Engineering Department, University of Genoa, Genoa, Italy (e-mail: paolo.buschiazzo@ unige.it).

Manuscript received 22 December 2008. Published as submitted by the authors.