A Rig Booking System for On-Line Laboratories

Alberto Gallardo, Thomas Richter, Peter Debicki
RUS Computing Center
University of Stuttgart
Stuttgart, Germany
{gallardo,richter}@rus.uni-stuttgart.de,
ruspete@po2.uni-stuttgart.de

Luis Bellido, Verónica Mateos, Víctor Villagrá
ETSI Telecomunicación
Ciudad Universitaria
E-28040 Madrid, Spain
{lbellido,vmateos,villagra}@dit.upm.es

Abstract—**Recently, many educational institutions have acknowledged the importance of making laboratories available on-line, allowing their students to run experiments from a remote computer. While usage of virtual laboratories scales well, remote experiments, based on scarce and expensive rigs, i.e. physical resources, do not and typically can only be used by one person or cooperating group at a time. It is therefore necessary to administer the access to rigs, where we distinguish between three different roles: content providers, teachers and students.**

This paper reports on a conceptual model and technical design of a rig booking system that provides mechanisms for content providers and teachers to control and grant access to on-line remote laboratories. The design of the booking system is based on a requirements analysis carried out by the EC funded LiLa project in cooperation with international partners from the Global Online Lab Consortium, GoLC.

Keywords-Remote Experiments; Reservation System; design for experiments

I. Introduction

In the context of the project LiLa ("Library of Labs"), many virtual and remote experiments of the contributing partners spread out over Europe will be made available through an internet portal, the so called "LiLa Portal". The portal will also collect feedback from the students to check their knowledge and learning success, and whether the experiments fit their needs. [1]

From a technical point of view, the LiLa portal is a repository of virtual and remote experiments with associated meta-data; the purpose of this metadata is two-fold: While it not only allows to search for experiments fitting the need of a teacher or a student, it also addresses the needs of librarians to integrate references to such experiments into library catalogues. The access to the experiments is provided via simplified SCORM [2] (Sharable Content Object Reference Model) objects; in the content of LiLa, these objects are also denoted as LLOs (LiLa Learning Objects), which can be downloaded as SCOs (Sharable Content Objects) and reused in other SCORM-compliant LMSs (Learning Management Systems) such as moodle or Ilias.

However, as the remotely controlled rigs are a scarce resource, access to such hardware resources must be scheduled by an access control mechanism; the aim of this access control is to accommodate as many students as possible and to help them organize their activities in the portal. This functionality can be provided by the LiLa booking system – at least for those experiments whose provider did not implement a scheduling system of their own. The booking system service does not come with its own user interface, but relies upon the LiLa portal GUI to administrate much of its functionality. The LiLa system, however, is extensible enough to allow content providers to deploy their own system if needed. It should be noted that, upon downloading an experiment from the LiLa portal, the access control by the booking system becomes an integral part of the SCO, and hence remains in effect if an experiment in SCO form is deployed in a learning management system.

The LiLa Booking System is built upon three different portal user roles – content providers, teachers and students – and four key elements: rigs, reservations for teachers, reservations for students and tickets. These roles and concepts will be explained in a later section.

This document is structured as follows. In section II we give an overview on related work in the area of experiments booking systems, following that, the workflow within the LiLa system is presented. In section IV we describe more detailed the conceptual model for the LiLa Booking System. Section V provides an overview of the system architecture. Section VI finally concludes and future work in this area.

II. Related Work

The concept of a booking system for remote laboratories is not new. Other projects have designed and implemented already some solutions that address the contended access to share resources. Some of these solutions are extensions to the functionality of selected learning management systems (LMS) and cannot be reused independent from them – hence, they are not suitable for LiLa purposes. For example, the MARVEL project [3] and its successor EDIPE [4] include a booking system for moodle [5]; WebLab[6] of the Slovak University of Technology in Bratislava is also module for moodle; and ReLEEP, developed by the Qatar University also uses a moodle specific solution extending the "Meeting Room Booking System" (MRBS) [7] to accommodate their needs.

Other solutions build on proprietary software, such as iLabs [8], which is open source itself, but requires a Windows Server machine and uses the Microsoft SQL Server and Visual

Studio.NET; or NETLab [9], developed by the Indian Institute of Technology, Kharagpur, that relies on proprietary software licensed products from Oracle, National Instruments and Agilent. In [10], the authors present a "Web Service-based MetaScheduling Service" to coordinate the allocation of time-slots with local resources in Grid-Computing environments developed in the context of the VIOLA project [11]. Solutions that are based on proprietary software are, however, unsuitable for LiLa purposes as well as they conflict with our license policy.

There is even an alternative approach adopted by the Kumamoto University, Japan, consisting of a time scheduling system compatible with various LMSs that focuses on the generation of courses timetables [12].

All these solutions are not easy to adapt, or cannot be reused at all, in the context of the LiLa project where the separation of the three roles is important, open source is a must, and re-usability and flexibility one of its main goals.

III. Typical Use Cases

Below, we describe the basic scenario for each of the different users of the LiLa booking system to provide an overview of the conceptual model our work is based on.

A. How a student accesses a schedule-controlled rig

Within the LiLa framework, access to experiments is provided to groups of students – typically members of a lecture. The membership to such a student group is identified by a "reservation code" the teacher typically provides during the lecture, e.g. by writing the code on a board. Along with the reservation code, it is in the responsibility of the teacher to hand out the information on the URL under which the experiments can be accessed. For LiLa purposes, this is typically the URL of the learning management system of the university, but the LiLa portal would be an alternative: the URL could point to an experiment hosted in the LiLa Portal or in a SCORM compatible LMS.

For a student interested in using this rig, she must enter this URL in her browser, or navigate the LMS to the experiment. The LiLa portal or the LMS will detect that the experiment requires the LiLa Booking System and will try to automatically authenticate the student.

If authenticated, the LiLa Booking System will further check if the student has allocated a time-slot that comprises the current time to use the rig. If the student has not allocated any time-slot, she will receive a message informing her of the need of making a reservation for the rig previous using it, and will provide her with an interface to request reservations.

Because such reservations should only be made by the audience of a specific lecture, students need to identify themselves to the system. While the student identity is provided by a Shibboleth architecture the LiLa system has access to, the membership to the lecture is, unfortunately, typically not included in the information the Shibboleth system can offer – it is the purpose of the reservation code to prove this membership.

The combination of this student identity with the reservation code is called a "ticket" in the following, and this ticket remains valid for the duration of the course, i.e. typically a semester, and then – once established – can be used for any subsequent reservation for the same lecture.

Once the validity of a ticket has been established, she must choose a suitable time-slot, and if the rig is free for use, the LiLa Booking System will "stamp" the ticket for the student, granting her the right of use the rig during that time-slot. The "stamping" prohibits that the same ticket can be used again to reserve the same experiment twice, and hence to block the system by booking all available time slots.

With one "reservation code", a student gets one ticket; and with one ticket, she can only get one stamped ticket. A student can only request a new stamped ticket cancelling her old one; additionally, a stamped ticked becomes valid for a new reservation again – i.e. the "stamp is removed" – once the time slot reserved by the ticked has been expired. Then, a new reservation can be made, using the same ticket again.

B. How a teacher allocates resources for her students

One of the teachers' tasks is to find experiments that fit their pedagogical intent for their lectures. If a teacher finds an interesting experiment that is subjected to scheduled access control, the teacher must contact the content provider of this rig used by the experiment, and negotiates with the provider when the rig will be available for her students. As product of this negotiation, the teacher will also get a "reservation code" from the content provider of the rig; however, this reservation code now protects the access to the rig reservation and is not identical to the student reservation code.

To make the rig accessible to her class, the teacher then must log in into the LiLa Portal, and access the LiLa Booking System functionality; there, she must enter this reservation code to add the negotiated availability of the rig to her calendar of rigs availability.

These negotiated time slots can then be broken up into several groups, e.g. several different lectures given by the teacher, each of the groups protected by its own "student reservation code". This is the code that will be made available to the students, one code per group.

The teacher can define different groups by creating different reservations for students with corresponding different reservation codes.

It should be noted that the term "teacher" does not necessarily need to address a natural person; a whole institution could share a single "teacher reservation code", i.e. an access right to a rig *plus* the ability to break up the negotiated time slots into student time slots by creating reservation codes.

C. How a content provider makes a rig available for teachers

The LiLa Booking System assumes that the owner – or provider – of a rig is the only one who is in the position of designing experiments for it. Thus, the LiLa Portal will identify any authenticated user that uploads an experiment as a content provider.

If an uploaded experiment identifies a rig it depends on, the LiLa Portal will offer to the content provider the functionality to create reservations for teachers as discussed above. This works similarly to the “reservations for students” that teachers create. That is, they define time availability for a selected rig.

A content provider that has negotiated the availability of a rig with a teacher will create one of these reservations for teachers; along with such a reservation, a “teacher reservation code” is created that then is handed over to the teacher.

IV. LiLa Booking System Description

We mentioned above the three portal user roles and the four concepts upon which the LiLa Booking System sets its foundations. In this section we explain these concepts with more detail.

The user roles are defined similarly to airline's computerized reservation systems [13]: the content providers in LiLa Booking System act as vendors who provide the services; the teachers act as representatives for the vendors; and the students act as customers.

This separation of roles determines the functionality available to the users of the LiLa Booking System. The main goal of this Booking System is to control access to rigs. Rigs are the scare resources to administrate, the expensive physical set-ups, the bulky and noisy machines, the fragile robots... and those whose access must be scheduled are available in LiLa through LLOs.

To limit access to the LLOs, the Booking System relies on the administrative elements “reservations for teachers”, “reservations for students” and “tickets”, depending on whose privileges they restrict.

Content providers are users of the LiLa Portal authenticated by the Shibboleth IdP of a LiLa-friendly institution who have uploaded at least one LLO to the portal and therefore own its administration rights. Having a trusted Shibboleth account is important for auditing reasons.

Content providers should not need to care about fine grained time-slots distribution among students in the same way as the airlines vendors don't care who a travel agency's representative sells a ticket to. Content providers create “reservation for teachers” to delegate the administration of the rigs during defined period of times to teachers.

Teachers are users of the LiLa Portal, also authenticated by a Shibboleth IdP, that “hold” one or more “reservations for teachers” – hold in the sense that they know the reservation code. They have rights to create “reservation for students”, that further refine when rigs will be available for their students.

Teachers have the main task of finding and getting resources for their students. Once a teacher finds an experiment that fits her pedagogical intent, she must negotiate with the content provider the available time slot on the corresponding rig; it should be noted that experiments and rigs are not necessarily identical, and that several experiments might require the same rig; or the same experiment might run on several identical or similar rigs as well.

The teachers, with these time-slots blocked for their exclusive use, can then make the corresponding experiments available for their students by distributing the “student reservation code”.

Students are users of the LiLa Portal or of any other LMS that hosts LLOs, that have one or more “student reservation code”. They can run the experiments supported by the LLOs. Students must be users with a Shibboleth trusted account; anonymous users are not allowed to use LLOs that require booking. A student that has a “reservation for students” can request a “stamped ticket” that grants her access to an RR-LLO during one selected time-slot.

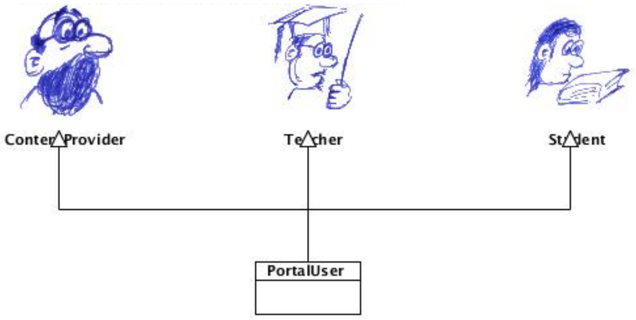

Figure 1. A Portal user can play any of these roles.

Reservations for teachers are administrative elements protected with a secret “reservation code” that content providers create to allocate rigs during selected time-slots for teachers of their choice. If teachers want their students to use selected LLOs, they must negotiate with the content provider which time-slots and under which conditions the rigs should be blocked.

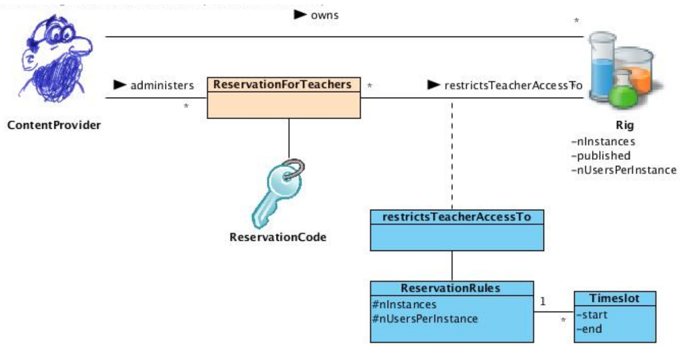

Figure 2. The content provider creates and administers reservations for teachers.

Reservations for students are the administrative elements protected with a secret "reservation code" that the teachers create to allocate rig time-slots for students. If teachers that already have "reservations for teachers" want their students to use the corresponding LLOs, they must further refine the time-slots blocked by the content provider for teachers' exclusive use.

This mechanism allows easy groups administration. For example, if a teacher wants to create two different groups of students of a selected rig, one using it on Monday and the other using it on Tuesday, the teacher would create two "reservations for students" with two different reservation codes; and then distribute one code to one of the groups and the other to the other group.

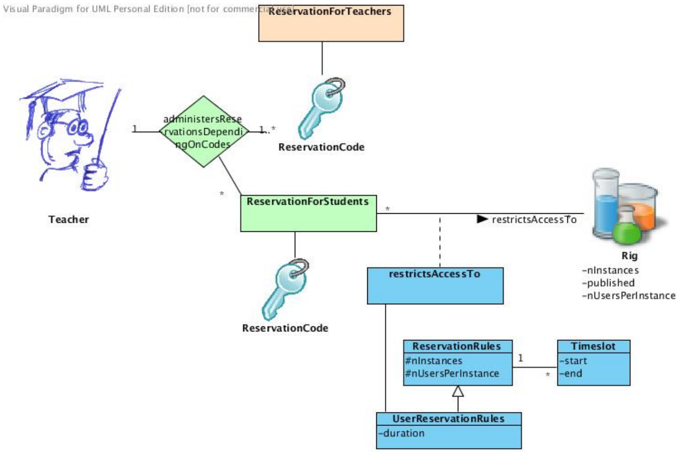

Figure 3. The teacher enters teachers reservation codes to allocate resources for the students. The teacher creates reservation for students to further refine rig availability.

Tickets allow students to make reservations within time slots reserved to their student group, and hence to retrieve stamped tickets from the booking system. To get a ticket, the student must enter a student reservation code. Once holding a valid ticked, which is associated to the student identity, the LiLa Booking System will provide her with the list of time-slots during which the rig is still available for use. When the student selects one time-slot, the LiLa Booking System "delivers" a *stamped ticket* to the student.

Stamped tickets, finally, grant students access to rigs in the time slot selected by the student.

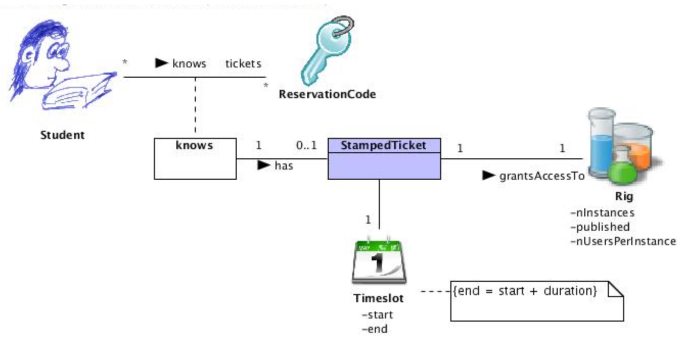

Figure 4. The student must enter a student reservation code to get a ticket, and can get a stamped ticket that grants her access to a rig during a time-slot.

To summarize, to make the rig accessed by an LLO available for use,

1) a Portal user must upload an LLO using the upload functionality of the LiLa Portal, becoming content provider;

2) this content provider must create at least one reservation for teachers using the content providers' functionality of the Booking System, protected with a secret "reservation code";

3) the content provider must provide this "reservation code" to at least one teacher;

4) a teacher must enter this "reservation code" into the Booking System using the teachers' functionality of the Booking System and create at least one reservation for students - using the teachers' functionality of the Booking System, protected with a new, usually different, secret "reservation code";

5) a student must enter this new "reservation code" into the Booking System - using the students' functionality of the Booking System; the system returns a ticket and associates it with the identity of the student

6) the student gets a stamped ticket by selecting an available time slot.

7) Once the student has a stamped ticket, she can redeem it and access the rig during its assigned time slot. After redeeming a stamped ticket, the ticket becomes available for a new reservation again.

The following activity diagram depicts the process of making a rig available and using it. It separates the corresponding steps into different lanes according to the Portal user role.

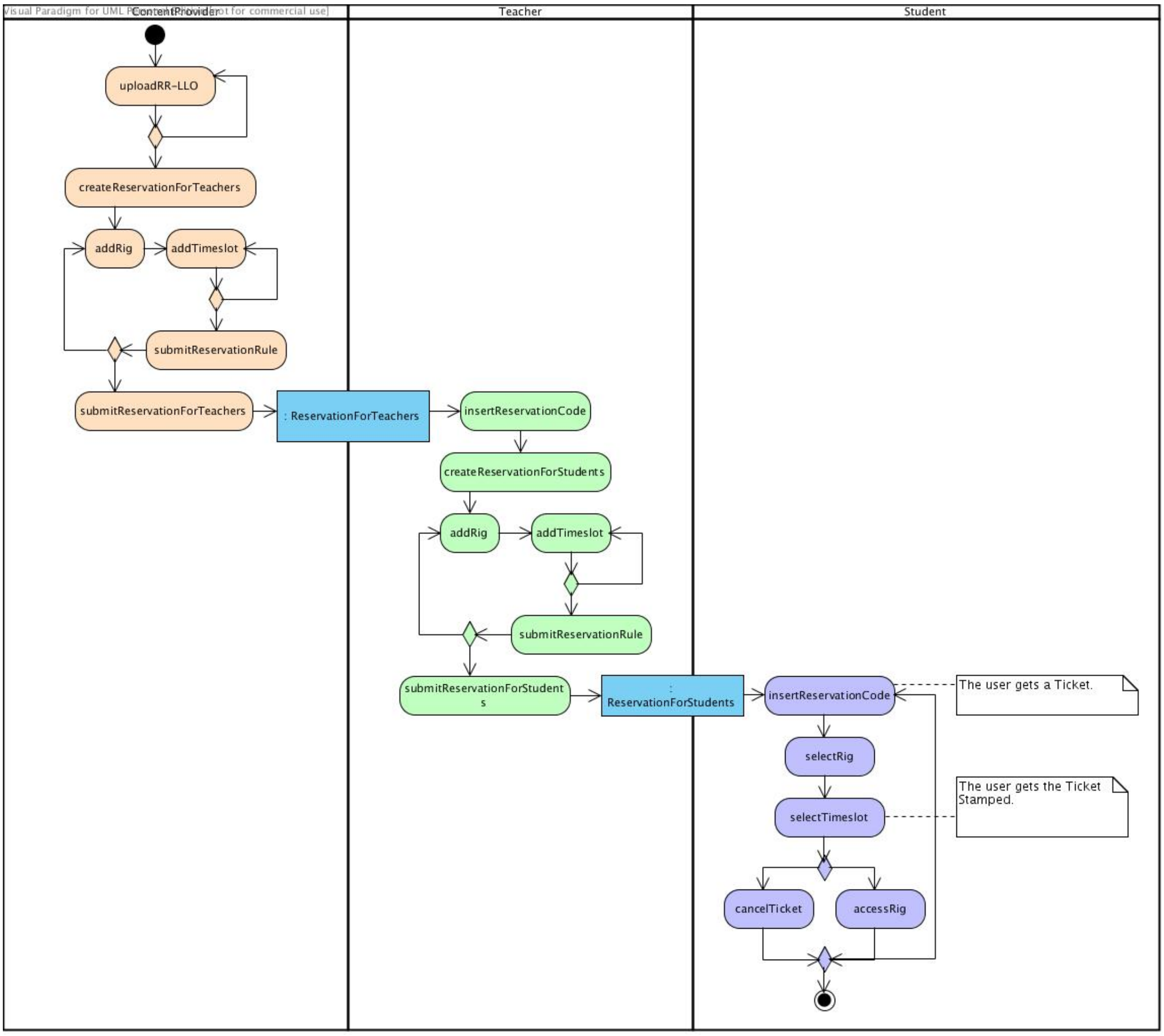

Figure 5. UML activity diagram that summarises the process of making a rig available for booking, and booking and using it.

V. Booking System Architecture

The Booking System will provide two levels of access control to LLOs: it will grant or deny access to LLOs hosted in the LiLa Portal; and it also will communicate with LLOs hosted in other SCORM-compatible LMSs (exported as SCOs) to provide basic access control functionality using AJAX technologies.

For LLOs hosted in the LiLa Portal, the Portal will display an "access not allowed" message when a student tries to run an LLO without a valid stamped ticket. For LLOs hosted in external systems, the Portal will embed JavaScript code into the exported SCOs that will check with the Booking System if the student has a valid ticket. If not, the embedded JavaScript code will block the access to the SCO content and will display an "access not allowed" message.

VI. Conclusions and Future Work

The paper discusses a solution to control access to virtual laboratories and remote experiments using learning objects, in the context of the LiLa project. The solution, based on the identification of three different user roles and the idea of tickets, is flexible enough to extend the booking system functionalities provided for learning objects accessed through the LiLa portal to learning objects accessed through external LMSs. The authors are currently working on implementation of the whole system that will help confirm the viability and expose possible problems of the proposed solution.

Acknowledgments

This work was partially funded by the eContent*Plus* programme of the EC. The authors want to thank the RUS computing center resp. the UPM for funding the work.

REFERENCES

[1] eContent*plus* project "Library of Labs," EU grant ECP-2008-EDU-428037, (2009)

[2] "Home SCORM®," November 2010, http://www.adlnet.gov/Technologies/scorm/default.aspx.

[3] J. Ferreira, A. Cardoso, "A Moodle extension to book online labs," Remote Engineering and Virtual Experimentation Symposium (REV 2005), Brasov, Romania.

[4] S. Uran, D. Hercog, K. Jezernik, "Remote Control Laboratory with Moodle Booking System," Int. J. Online Engineering, vol. 1, n. 2, 2005.

[5] "Moodle.org: open-source community-based tools for learning," November 2010, http://moodle.org/.

[6] Ľ Čirka, M. Kvasnica and M. Fikar, "WebLab Module for the Moodle Learning Management System," Proceedings of the 9th International Conference Virtual University 2008, E-academia Slovaca.

[7] "MRBS: Introduction," November 2010, http://mrbs.sourceforge.net/.

[8] "MIT iCampus: iLabs," November 2010, http://icampus.mit.edu/iLabs/.

[9] A. Maiti, "NETLab: An Online Laboratory Management System," Education Engineering (EDUCON), 2010 IEEE, 24 Juni 2010.

[10] P. Wieder, O. Wäldrich, W. Ziegler, "Advanced techniques for scheduling, reservation, and access management for remote laboratories," Proceedings of the Second IEEE International Conference on e-Science and Grid Computing (e-Science'06), 2006.

[11] "VIOLA," November 2010, http://www.fz-juelich.de/jsc/grid/VIOLA.

[12] H. Nakano et al., "Web-based time schedule system for multiple LMSs on the SSO/Portal environment," IEEE EDUCON Education Engineering 2010, Madrid, April, 2010.

[13] P. Chang, "Computerized Reservation Systems," IEEE International Conference on Systems, Man and Cybernetics, 1992.