

# Lab4CE: a Remote Laboratory for Computer Education

Julien Broisin<sup>1</sup> · Rémi Venant<sup>1</sup> · Philippe Vidal<sup>1</sup>

Published online: 10 November 2015

© International Artificial Intelligence in Education Society 2015

**Abstract** Remote practical activities have been demonstrated to be efficient when learners come to acquire inquiry skills. In computer science education, virtualization technologies are gaining popularity as this technological advance enables instructors to implement realistic practical learning activities, and learners to engage in authentic and problem-based learning. However, virtualization solutions have not been designed especially for education and do not address any pedagogical concern. Since several large-scale studies showed that instructional supports during practical activities are almost as important as technical features, this article investigates the following research question: how the scaffolding around the lab increases students' engagement in remote practical learning of computer science? To answer this question, we introduce the Lab4CE environment, a remote laboratory for computer education which adopts a distributed, modular and flexible architecture to integrate a set of scaffolding tools and services intended for instructors and learners. An exploratory study conducted with 139 undergraduate students enrolled in the first year of a computer science degree suggests a positive effect of the framework on learners' engagement when they come to practice system administration, and reveals a significant positive correlation between students' activity within the system and students' learning achievement.

**Keywords** Online learning environment · Remote laboratory · Computer science

## Introduction

Distribution at a large scale of online learning resources and activities has been in the focus of research in the past few years, but a lower attention has been given to activities

---

✉ Julien Broisin  
julien.broisin@irit.fr

Rémi Venant  
remi.venant@irit.fr

Philippe Vidal  
philippe.vidal@irit.fr

<sup>1</sup> University of Toulouse III, 118 route de Narbonne, 31062 Toulouse, France

that require practice into a laboratory. Practical activities, referred to as “any learning and teaching activity that engages learners in manipulating and analyzing real and physical objects” in this document, are efficient when learners come to acquire inquiry skills (de Jong et al. 2013). In STEM (Science, Technology, Engineering, and Mathematics) fields, inquiry-based learning is a pedagogical method relying on constructivist and socio-constructivist theories of learning, that allows students to learn about science by engaging them in investigation (Bell et al. 2010). In such contexts, learners build their own interpretations of scientific concepts and acquire knowledge about how to do science through realistic works.

Compared to traditional practical activities, those mediated by a remote laboratory (lab) bring a number of advantages (Lowe et al. 2013): they can be used by a large pool of students spread across multiple institutions (i.e., learners of a secondary school consume a laboratory located in another institution (Orduña et al. 2012)); a wide range of equipment (e.g., civil engineering beam (Lowe et al. 2009a), modular vector network analyzer (Leproux et al. 2013), spectrometer (DeLong et al. 2010)) is accessible to students at any time and any location; a large amount of data can be gathered and analyzed; students gain longer interaction time with the apparatus and get higher chances to develop deeper understanding; results obtained by previous students can be reused as a starting hypothesis for the subsequent ones, while all data and conclusions can be available to all students. In the remainder of this paper, remote practical activities refer to traditional practical activities, as defined in the previous paragraph, extended and modified to be accessible online by anyone, anytime, from any device connected to the Internet.

In the context of computer science education, virtualization tools and technologies are gaining popularity over classical ones (Kriz 2014; Bonner et al. 2013) as they significantly facilitate the conception of realistic, complex, controllable and repeatable computer networking experiments. Even if these technologies satisfy most technical expectations, providing learners and tutors with remote access to these environments is not sufficient to reach learning effectiveness: as pointed out by Corter et al. (Corter et al. 2011, p. 2056), “the scaffolding around the lab may be at least as important as the lab itself”. Also, when using virtualization tools, users are not aware of actions carried out by others over the virtual resources; as a result, it is very difficult, even nearly impossible, for distant tutors to guide and support learners when they encounter problems or blocking situations.

This paper thus tackles the following research question: how the scaffolding around the lab increases students’ engagement in remote practical learning of computer science? Even if at least three types of engagement (behavioral, emotional, and cognitive) have been identified by Fredericks et al. (2004), we refer in this article to on-task behavior; more especially, by *engagement* we refer to students involved in (remote) practical activities. To answer this question, we (1) introduce a remote laboratory environment standing on existing virtualization tools so as to benefit from their advanced computational features, and (2) integrate original scaffolding tools and services into this system to improve the user experience and to increase students’ engagement in the context of remote lab activities. The remainder of the paper is organized as follows. The next section exposes how virtualization technologies strengthen companies’ information systems, and highlights their weaknesses when used within an educational context; this section also specifies a set of scaffolding

capacities required to effectively support remote practical activities in computer science, and studies the position of others STEM remote lab projects and initiatives in relation to these requirements. We then introduce Lab4CE, our remote laboratory for computer education, and detail a set of tools and applications integrated into this environment to operationalize the scaffolding features. An exploratory study conducted within our teaching institution follows, together with a discussion about how additional supports could be integrated into the system. The last section gives conclusions and future works.

## Background

This section introduces the foundations of our work: the virtualization tools available for use in computer education, and the mature and ongoing remote laboratories that enable remote practical activities.

### The Requirements for an Educational-Oriented Cloud

Virtualization tools have gained popularity during the past few years, as the wide variety of commercial, free and open source initiatives currently available on the market demonstrates it. Among others, virtualization techniques bring a series of advantages within companies when compared to traditional approaches: the number of physical servers hosting the various IT services is reduced, as a single physical server can be used to deploy several virtual machines hosting a given service; as a consequence, the costs induced by the storage of these equipments in dedicated air-conditioned spaces, but also those induced by their continuous administration and maintenance, are reduced, while the energy consumption is decreased; the production settings can be simulated easily, so that disaster recovery plans can be easily implemented, new products and services can be developed and tested in real conditions without impacting the company's productivity, and new employees can be trained in real production conditions.

When it comes to education, and especially to computer science education, the primary advantage of virtualization is the rise of the degree of freedom for the learner. In practical learning, a full access to the computers is required to experiment concepts such as computer security or system and network administration (Wang et al. 2010), with the risk of a hazardous manipulation from the learner ending to a security breach or a machine out of order. In a real environment, granting such an access is then avoided, while in a virtual environment that barrier vanishes, since a faulty machine dedicated to a specific learner will not never prevent others to access their own lab and can be replaced instantly. Last but not least, virtualization technologies provide substantial advantages for remote labs: their management, like resources allocation, gains flexibility since these technologies allow distribution and live migration of virtual machines on an IT stock (Sahoo et al. 2010); these technologies also prevent side effects between labs that share the same physical resources by isolating virtual components from each other (Kroeker 2009). However, even if virtualization might offer significant improvements, it has never been thought firstly for educational purposes. Virtualization tools remain intended for computing experts and professionals, requiring

important knowledge in computer science. In order to provide teachers and learners with an education-oriented cloud, we need to give them the intelligible tools that translate pedagogical wishes into technical orders.

Therefore, virtualization tools and technologies must be enriched to support effectively learning actors (both instructors and learners) during the practical learning process. Our approach to bring an answer to the research question asked in the introduction consists in studying virtualization environments and remote practical learning as interdependent rather than separate processes. A survey was conducted in different Australian states with 143 students to compare the perceived learning outcomes of remote and hands-on labs (Kostulski and Murray 2011). Students pointed out two main points: “help and support, if required”, and “engagement in the experiment”. This result is “very much inline with the opinions of a large number of academics who had also identified engagement as an area where remote labs need to evolve further” (Kostulski and Murray 2011, pp. 209). Thus, the remote lab environment should give learning actors access to a *common view of the experiments*, but also the possibility to continuously *sharing the control over the remote experiments*. The system should also offer *synchronous communication tools*, as well as *social awareness tools*, to bring students the feeling of being connected with and supported by their peers and instructors (Lowe et al. 2009b). To support tutoring facilities, the remote lab environment should include *learning analytics tools* allowing tutors to monitor both the learners activities and the detailed status of the remote virtual resources, so as to easily and quickly identify students facing with blocking situations and assist them with exactly the support they need (Bell et al. 2010). Instructors should also have the opportunity to design experiments through a user-friendly *authoring tool* to encourage and facilitate the configuration and re-engineering of online experiments.

Before introducing our proposal to build a system that makes possible the use of virtualization technologies in an educational context while aiming at increasing learners’ engagement in remote practical activities, we first analyse how the existing remote labs projects position themselves in relation to the above requirements.

### The Existing Remote Laboratories

A large number of initiatives and projects emerged from the past decade to investigate how traditional practical activities could be offered to distant learners. The mature projects that are considered as significant in the field of virtual and/or remote laboratories include: the iLab<sup>1</sup> framework that has been initiated in the 2000’s by MIT and supported by Microsoft ©, the Library of Labs<sup>2</sup> (LiLa) co-funded by the Community Program eContentplus from 2009 to 2011, the joint Australian project LabShare,<sup>3</sup> the WebLab-Deusto<sup>4</sup> developed at the University of Deusto and based on the iLab architecture, the Go-Lab<sup>5</sup> (Global Online Science Labs for Inquiry Learning at

<sup>1</sup> <http://ilabcentral.org>

<sup>2</sup> <http://www.lila-project.org/>

<sup>3</sup> <http://www.labshare.edu.au/>

<sup>4</sup> <http://weblab.deusto.es/>

<sup>5</sup> <http://www.go-lab-project.eu/>

School) project funded by the European Commission under the FP7 program from 2012 to 2016, and the GOLC consortium's Lab2Go<sup>6</sup> portal.

We studied these various projects to evaluate their fit against the pedagogical capabilities we identified in the previous section; results of this investigation appear in Table 1. Remote experiments made available through iLab are handled by an individual experiment's virtual instruments preventing any collaborative or tutoring support (Harward et al. 2008). LiLa (Richter et al. 2011) adopts the SCORM standard to make pedagogical resources and remote experiments available to students. If this standard promotes sharing and reusing of experiments, its tasks-oriented approach does not suit the need for synchronous apparatus control and limits the tracking of activities to "completed/not completed". LiLa should address cooperation between students through Open Wonderland<sup>7</sup>, a toolkit to build collaborative virtual world; however, a presentation available on the project web site does not refer to such a system. The collaboration services offered by Labshare to users are limited: learners and teachers can simultaneously control a remote equipment through a virtual network computing (VNC) toolkit, but the use of this kind of tool prevents an effective tracking of users (Lowe et al. 2009a); in addition, no awareness tools are provided. WebLab-Deusto offers a set of remote experiments as SCORM packages (Sancristobal et al. 2010), thus the above limitations apply. Collaboration and awareness tools are limited to those included into the learning system of the institution and make very difficult the tutoring process. Lab2Go focuses on technical issues of remote labs and experiments without taking into account relevant learning and collaborative features. Finally, even if the primary goal of Go-Lab is to federate and share a wide pool of distributed laboratories, it includes some learning facilities as well: teachers can use the portal to build learning scenarios as inquiry learning spaces (ILS) containing online labs, instructions, learning resources or apps (Govaerts et al. 2013; Gillet et al. 2013). An ILS can be shared to several students, but students cannot collaboratively control the remote resources and benefit from a common view of the experiment. Moreover, the monitoring of users activities lacks a detailed description of learners' actions, due to privacy issues that are of most importance when dealing with pupils from secondary education. Finally, none of these environments take into account experiment instructional design.

To sum up, these projects do not focus on pedagogical concerns. Instead they have engaged significant efforts to tackle a common issue: the sharing of remote labs composed of a set of various apparatus and devices offering online experiments in different STEM learning areas to a wide range of students (i.e., high school, undergraduate, bachelors), at an international (i.e., iLab all over the world, LiLa, Lab2Go and Go-Lab in Europe) or national scales (i.e., WeLab-Deusto in Spain or LabShare in Australia). Some of them addressed challenges such as scalability, security, reliability and user management (Harward et al. 2008; Richter et al. 2011), while others enhanced the description of labs and experiments through the use of semantic web technologies to make their retrieval by teachers and learners as efficient as possible (Zutin et al. 2010). These researches provide very interesting solutions to complex computer science problems such as architectural design (Harward et al. 2008; Lowe et al. 2009b), reservation and queuing algorithms (Lowe 2013), load balancing

---

<sup>6</sup> <http://www.lab2go.net/>

<sup>7</sup> <http://openwonderland.org/>

**Table 1** Synthesis of ongoing remote lab projects

	iLab	Lila	Labshare	WebLab	Lab2Go	Go-Lab
Requirements						
Common view of the experiment	-	~	~	-	-	-
Shared control of the experiment	-	-	~	-	?	-
Synchronous communication tools	-	?	~	~	?	~
Awareness tools	-	-	-	~	-	~
Learning analytics tools & dashboards	~	~	-	~	-	~
Authoring tool	-	-	-	-	-	-

Legend: requirement supported (+), partially supported (~), not supported (-), unknown (?)

(Sancristobal et al. 2010) or standardisation as smart devices (Salzmann et al. 2015) and learning analytics (Orduña et al. 2014), but a lower attention has been given to learning outcomes even if Singer et al. (2005) recommend to design remote practical activities with clear learning outcomes in mind so they achieve their intended learning goals.

## LAB4CE: The Big Picture

As its world-wide adoption demonstrates it (Hardison et al. 2008; Niederstaetter et al. 2010; Zutin and Auer 2011), iLab specifies a robust and scalable architecture to manage remote labs. Hence, to reuse existing virtualization tools and technologies, but also to suggest a modular framework with integration capabilities, we designed a remote laboratory for computer education based on a three-layered architecture inspired from the iLab shared architecture (Harward et al. 2004) and illustrated in Fig. 1: on top of the virtual equipments on which practical sessions take place, the middleware layer offers a set of services that can be invoked by users through various rich pedagogical interfaces.

The objective of the laboratory layer is to support learning of computer science at scale; the term “scale” denotes both the capability of managing a massive amount of virtual machines using existing virtualization tools, and the capacity of supporting any topic of computer science, as virtual machines and physical computers are characterized by the same logical properties. This layer is thus responsible for the management of the virtual machines and networks offered to end-users, and of the accreditations assigned to these users on the virtual resources. Lab4CE implements OpenStack<sup>8</sup> within this layer, an open source cloud computing platform to build private and public clouds. Considering our objectives, OpenStack is one of the Infrastructure-as-a-Service (IaaS) solutions that best suits our needs: (i) it gives software-defined network capabilities which are of most importance in our context (Zhang et al. 2013); (ii) it supports a fine-grained and customizable virtual organization to manage different kind of actors, resources and relations between them (Wei et al. 2014); (iii) it exposes most of its services through a REST API, and thus facilitates its integration within our environment (Bist et al. 2013); (iv) it supports a wide range of virtualization technologies for

<sup>8</sup> <https://www.openstack.org/>

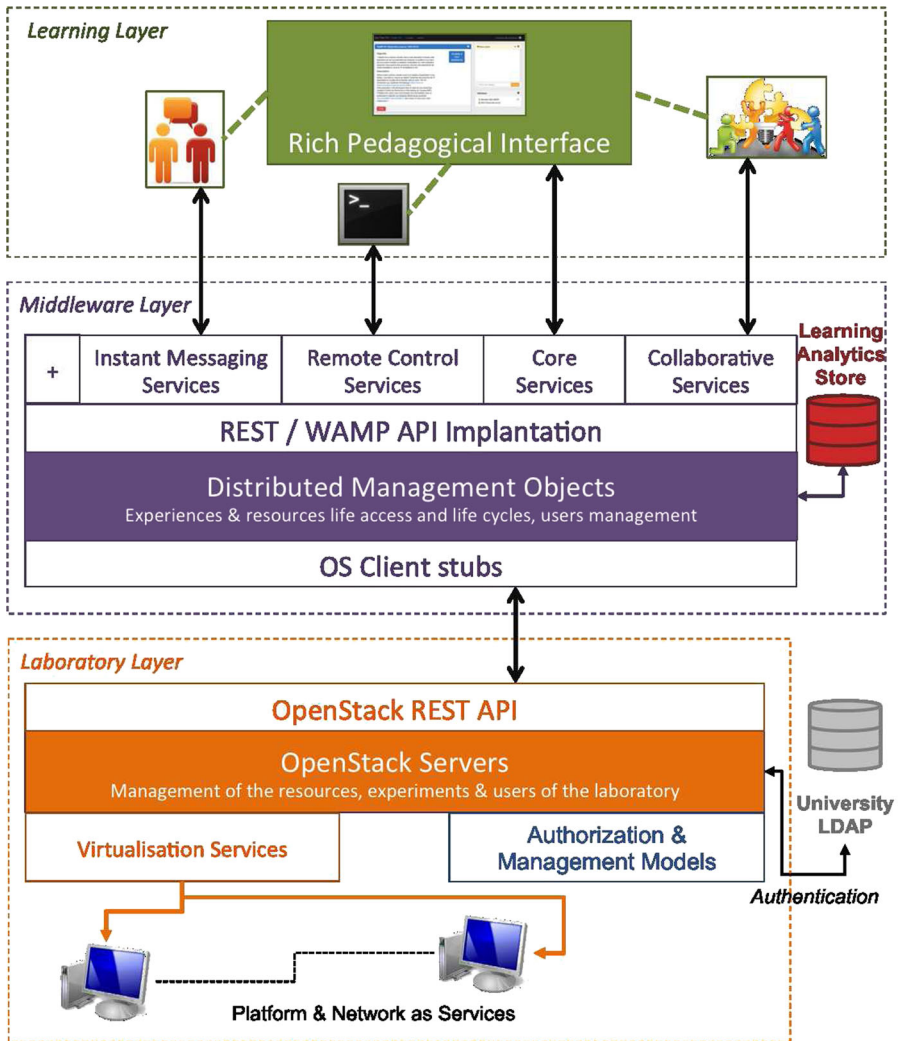


Fig. 1 The architecture of the Lab4CE environment

computers (e.g., KVM, Xen) and networks (e.g., LinuxBridge, Open vSwitch, Cisco Nexus) (Barkat et al. 2014); (v) it is supported by an active and fast-growing community (Wen et al. 2012); and (vi) it adopts a flexible modular architecture that makes it able to delegate certain of its services, such as the authentication mechanism, to an OpenID (Khan et al. 2011) or LDAP server; this configuration has been chosen for Lab4CE to integrate the university information system.

Our approach to operationalize remote practical activities distinguishes two distinct concepts: the *experiment* and the *practical session*. An experiment is a specification defined by teachers to specify the virtual resources (together with the possible inter-connections between them) and the work that must be done by learners to reach a given pedagogical objective such as being able to configure a local network at the Internet Protocol (IP) level. A practical session refers to an instantiation of an experiment and is



owned by a given learner; it provides him/her with the virtual resources required to conduct the matching experiment. Thus, in addition to the native OpenStack features, the laboratory layer includes a module responsible for (1) the integration of these specific concepts into the software, and (2) the management of authorizations. This module ensures the mapping between our concepts and native OpenStack artifacts (i.e., an experiment is represented within OpenStack by a *group*, and a practical session fits an OpenStack *project*), and defines authorizations as policy rules according to a role-based access control approach; these rules are detailed further in the paper.

The middleware layer represents the core of the Lab4CE environment. Indeed, this layer acts as a broker between the learning and the laboratory layers and adopts a service-oriented architecture to offer a seamless communication between end-users and virtual resources. The upper-level of the middleware exposes to the pedagogical interfaces a set of core and learning services whose orchestration is ensured by a set of distributed objects. The core functionalities stand on RESTful services to ease the design of high-level interfaces that facilitate the remote administration and control of both experiments and practical sessions; the lower-level of the middleware achieves the matching treatments by invoking, through dedicated client stubs, the low-level services supplied by OpenStack so as to concretely carry out actions on the virtual resources hosted by the laboratory layer. The learning features rely on the Web Application Messaging Protocol to integrate collaboration settings and synchronous communication capacities.

Moreover, the middleware layer embeds a learning analytics store to record all users' activities, including both actions carried out on the OpenStack virtual resources, and interactions with other users. This data store is based on Elasticsearch<sup>9</sup>, an open source software featuring real-time search and analytics capabilities, as well as a sophisticated REST API that facilitates the development of rich analytics tools and dashboards; a tool reusing the data saved into this store is detailed further in the paper.

Finally, the learning layer represents the rich pedagogical interfaces dedicated to end-users. This layer interacts with the core and learning services delivered by the middleware to instrument the set of facilities (i.e., common view of and shared control over a practical session, communication and awareness tools, as well as learning design and tutoring artifacts) required to effectively support users during the practical learning process, and aims at offering the best user experience as possible. The learning layer currently hosts two main pedagogical interfaces. On one hand, an authoring tool is intended for instructors and facilitates the conception, configuration and publication of remote experiments within the laboratory layer. On the other hand, a rich learning interface dedicated to learners and tutors supports interactions with the remote virtual resources, and provides them with various scaffolding tools and services. As they represent the most significant added values of the Lab4CE environment, the graphical applications integrated into the learning layer constitute the main focus of the following sections.

<sup>9</sup> <https://www.elastic.co/products/elasticsearch>



## The Authoring Tool

As discussed earlier, current remote laboratory environments are poorly featured with authoring facilities. The educational platform of the ongoing Go-Lab project, Graasp.eu<sup>10</sup>, comprises a macro learning design feature (i.e., various learning resources and activities can be integrated and organized within an inquiry learning space), but the applications made available to access the remote virtual or physical apparatus cannot be configured to meet some specific pedagogical objectives. This lack of micro learning design capacity may be explained by the fact that practical activities integrated into STEM learning scenario often consist in applying different values to a (set of) parameter(s) of one or several apparatus in order to investigate the results and to find out or prove a well-known physical law; another explanation for this trend may be that acquiring a specific skill requires a dedicated apparatus.

At the opposite, computer science education includes a wide variety of disciplines, ranging from programming languages to databases through architectural concepts and networking computing, which can all be tested through a single apparatus (i.e., a computer). Therefore, in the context of remote computer science experiments, it should be possible to configure and customise the remote laboratory (i.e., the virtual resources) according to the specific discipline to be taught; indeed, a given programming language cannot be experienced if it is not installed and properly configured on the remote lab.

## Design and Illustration

Lab4CE supplies a micro authoring tool illustrated in Figs. 2 and 3. To design an experiment, instructional designers have first to fill the form of Fig. 2 in order to specify the learning metadata (i.e., the name, description, pedagogical objective and period of availability) of the experiment. Then, the application allows instructors and teachers to design an experiment at two granularity levels so that both simple (i.e., experiments composed of a single computer) and complex (i.e., experiments comprising a set of computers and networking equipments) experiments can be considered. The main panel on the left-hand side of Fig. 3 allows to define the experiment's topology by dragging and dropping the equipments listed in the *Components* tab of the right-hand side panel; three types of components are currently available: Computer, Router and Switch. Each component is characterised by a default software configuration which can be adapted according to some specific pedagogical needs using the *Settings* tab illustrated in Fig. 3; for example, all computers host a default Ubuntu Linux distribution only, but any additional software listed in the *Software catalog* box can be installed and automatically configured as well.

In the scenario illustrated in Figs. 2 and 3, the instructor designed an experiment whose objective is to help learners become familiar with the basic commands dedicated to the Internet Protocol configuration on a Linux-based computer. To support this objective, the teacher set up a network topology composed of four computers, two switches and one router, and linked some of the equipments between them. Each component has a default configuration, except *Station 3* which hosts two additional

---

<sup>10</sup> <http://graasp.eu/>

Create a new remote experiment

### New Remote Experiment

**Name**  
The basics of Internet Protocol (IP)

**Objective**  
The objective of this experiment is to get familiar with the basic commands dedicated to the Internet Protocol configuration on a linux-based computer.

**Description**  
In the network topology below, make all four stations able to talk to each other. Stations 1 & 2 must belong to the same subnet whereas Stations 3 & 4 must belong to another subnet.

**Author**  
Julien Broisin

**Start date**  
June 1, 2015

**End date**  
June 30, 2015

Fig. 2 The authoring tool: learning metadata of an experiment

software (i.e., *apache* and *dnsmasq*). The teacher also described the task (i.e., make all four stations able to talk to each other) assigned to learners together with the associated

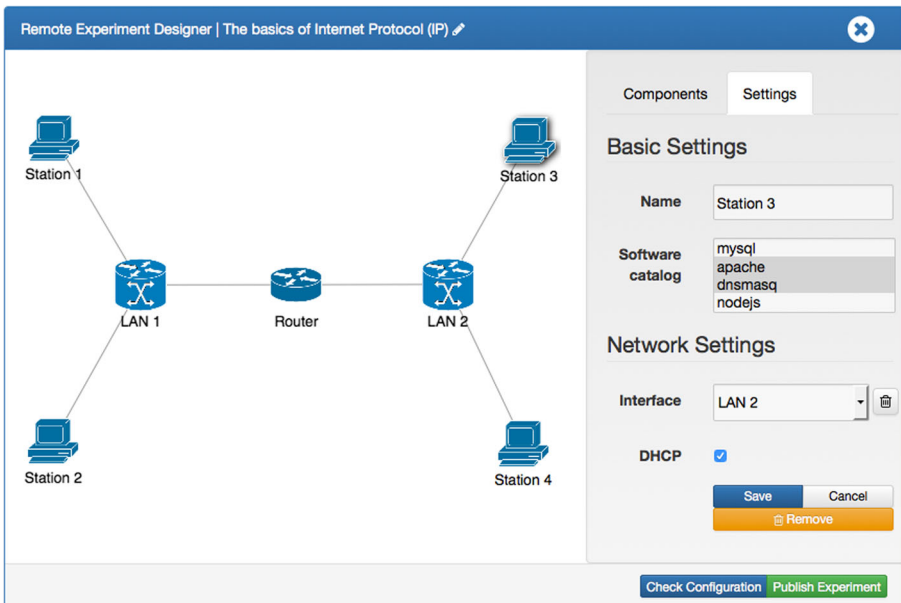


Fig. 3 The authoring tool: computational configuration of an experiment

constraints (i.e., Stations 1 & 2 must belong to the same subnet whereas Stations 3 & 4 must belong to another subnet). In this scenario, the experiment is available to learners from May 20, 2015 to June 30, 2015.

## Implementation

Once instructional designers have configured each equipment of the experiment's topology, they are able to publish the experiment into the laboratory layer (i.e., into the OpenStack's project database). The *Publish Experiment* button invokes the experiment management service of the middleware layer which ensures the mapping between our specific configuration file structure, and the format adopted by OpenStack; this process guarantees independence between the authoring tool and the virtualization software of the laboratory layer. In addition, an image describing the experiment is automatically generated during the publication process and reused later into the rich learning interface to provide learners with a clear view of the experiment's topology (see the next section).

Let us remind that an experiment defines some pedagogical objectives and depicts the apparatus made available to learners to reach these objectives. Therefore, the authoring tool does not trigger the deployment of any virtual resource. Instead, the virtual resources are deployed *on demand* when learners access to their own practical session using the rich application described in the following section.

## The Scaffolding Tools

This section details the Rich Learning Interface (RLI) which represents the space where interactions between users and the remote laboratory occur. Systems such as the Co-Lab collaborative learning environment (Van Joolingen et al. 2005) combine collaboration with inquiry learning, since this combination may be a means to improve students' processes of inquiry (Saab et al. 2012; Pinkwart et al. 2010): as most scientific research is a team activity, learners generally engage in tasks in which they do experiments in groups, and through which they are expected to develop collaborative skills. In addition to remote control capabilities, the web application presented here thus also includes communication, collaboration and awareness artifacts aiming to leverage the user learning experience during a practical activity.

## The Remote Control Capabilities

The rich learning interface supplies functionalities related to the control of a remote experiment on one hand, and to the control of the virtual resources composing a practical session on the other hand. Yet, the capabilities offered to users vary depending on their role into the system, and on the policy rules we defined. We currently distinguish two types of users: *teacher* (or tutor) and *learner*. However, as Lab4CE's features include collaborative work between learners (i.e., they can work together on a given practical session), we also distinguish the *owner* of a practical session and the *guest(s)* that has(ve) been invited to participate to the collaborative session; the mechanism to invite one or more users to a practical session is presented later in the

paper. Table 2 exposes the policy rules that have been specified to manage authorizations on experiments, practical sessions and virtual resources according to these roles. Since the previous section described the experiment authoring tool, the functionalities offered to teachers that are described below deal with practical sessions and resources.

### *Control Over Experiments - Design and Illustration*

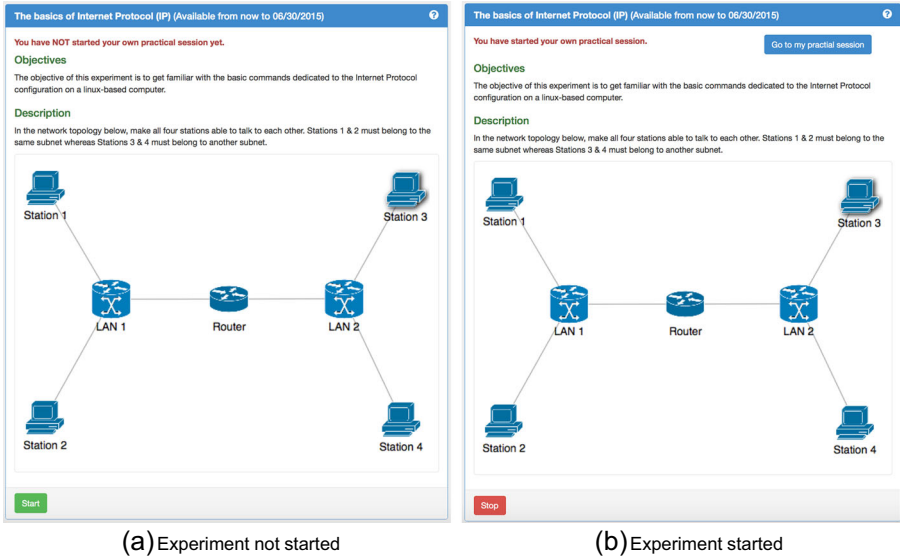
Once logged into the Lab4CE's RLI, learners have access to the screen of Fig. 4a to *consult the metadata* of the experiment and to view the image illustrating the experiment's topology that has been generated by the authoring tool. Starting from this screen, they are able to *start* their practical session (i.e., their own instance of the experiment) by clicking the green button of Fig. 4a. When their practical session is ready, learners can *access* it through the blue button displayed at the top right corner of Fig. 4b, or *stop* their practical activity by clicking the red button of Fig. 4b. Teachers are allowed to *access* the practical session of all learners, and they can also benefit from their own practical session for verification tests and demonstration perspectives. Finally, guests are only allowed to *access* the learner's practical session, they are not authorized to start and stop it.

### *Control Over Practical Sessions - Design and Illustration*

The web application illustrated in Fig. 5 is exposed to users when they access a practical session. The main panel comprises a tab-based navigation bar to manage the virtual resources' life cycle: owners of a practical session are able to *start*, *stop*, put in *sleep* mode and *resume* each resource of their own practical session, whereas teachers are provided with these capacities for the resources of all practical sessions. Ma and Nickerson (2006) demonstrated that Terminal window brings technical and professional competences and skills to learners; it also increases their motivation, as they feel as if they were working on real systems. Hence, another feature gives all users (i.e., teachers, the owner of the practical session as well as guests) the capacity to *control* a virtual resource through a web Terminal so as to run command lines and programs and to achieve the objectives of the practical activity. The implementation of the control interface as a Terminal restricts the panel of computer science topics that can be practiced to those that do not require a graphical interface, but takes into consideration a significant part of them including system and network administration, programming, database management, etc.

**Table 2** The role-based policy rules to interact with experiments, practical sessions and resources

	Teacher/tutor	Learner		
Experiment	Design Publish Update Delete	Consult metadata		
			Owner	Guest
Practical Session	Start/Stop Access	Start/Stop Access		Access
Resource	Start/Stop Sleep/Resume Control	Start/Stop Sleep/Resume Control		Control



**Fig. 4** The start screen of the rich learning interface

### Implementation

The RLI has been developed using the two popular AngularJS<sup>11</sup> and Bootstrap<sup>12</sup> frameworks, whereas the web Terminal is based on Shellinabox<sup>13</sup>, a web-based tool developed using Ajax technologies to reproduce the look and feel of native Shell windows (see Fig. 5); we adopted Shellinabox as it is efficient when it comes to improve web interaction in computing courses (Morell and Jiang 2015).

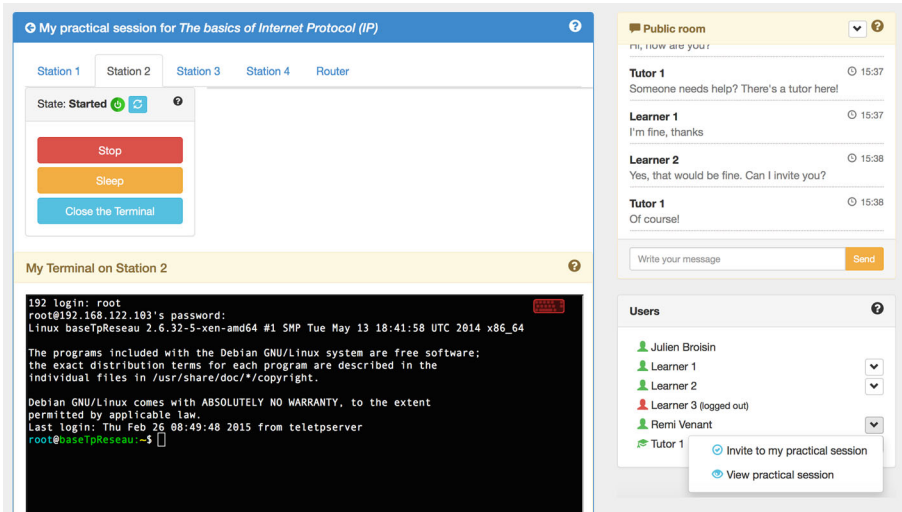
The middleware layer ensures the treatments associated with the various actions made available to users through the interface. The activation of the *Start* button of Fig. 4a triggers the invocation of a service hosted by the middleware layer and responsible for the deployment, within the OpenStack cloud, of the learner's set of virtual resources described into the experiment configuration file stored into the OpenStack information system. At the opposite, when users activate the *Stop* button of Fig. 4b, this service will destroy within the OpenStack laboratory all the learner's virtual resources associated with this experiment. All the actions related to the control of a virtual resource are concretely carried out on the appropriate virtual node of the OpenStack cloud through another service of the middleware layer, and the resulting outputs produced by the remote resource are then forwarded to and displayed by the learning interface. Finally, the set of policies has been implemented within the laboratory layer by adapting OpenStack to our specific authorization rules.

The core features presented here allow individual practical activities to take place. To extend this capacity and to support team work between learners, but also between learners and tutors, we integrated into the web application two distinct components

<sup>11</sup> <https://angularjs.org/>

<sup>12</sup> <http://getbootstrap.com/>

<sup>13</sup> <https://code.google.com/p/shellinabox/>



**Fig. 5** The rich learning interface

dedicated to collaboration: a synchronous communication system and an artifact-awareness tool for shared session.

### The Instant Messaging System

To allow collaboration between users located in different geographic places, synchronous communication tools are required so that users are able to talk to each other, ask/provide help or exchange facts and ideas (Bochicchio and Longo 2009). Different techniques can be used to provide such communication capabilities, including instant messaging, audio-conference, video-conference and 3D-chat (Röhrig and Jochheim 2001). However, the survey conducted by Lowe et al. (2009a) showed that 40 % of students who regularly used a remote lab identified the instant messaging, or online chat, as the preferred method of communication.

#### *Design and Illustration*

The instant messaging service we designed appears at the top right corner of Fig. 5 and gives users an opportunity of sharing questions, ideas and findings about the practical activity. Our instant messaging system also distinguishes the experiment and the practical session concepts by supporting two types of rooms. On one hand, one public room is associated to the current experiment; within that room, any user involved in this experiment is able to post messages. On the other hand, one private chat room is associated to each practical session of this experiment; in other words, one private chat room is provided to one learner. Within a private room, the owner of the matching practical session, the guest(s) and the tutor(s) only have the required credentials to post a message. Two types of discussions can thus take place: the public room is appropriate to general discussions about theoretical knowledge required to achieve the objectives of the experiment, whereas the private rooms are relevant to bring precise help to students

regarding specific issues that must be solved, or actions that must be carried out, to properly control the resources of the remote lab.

Figure 5 shows two learners and one tutor exchanging posts through the public room of the chat system. The tutor first proposes students to support them. One learner (i.e., *Learner 1*) thinks he/she does not need help at this moment. Another learner (i.e., *Learner 2*) needs assistance and accepts the help proposed by the tutor. Then, *Learner 2* will be able to exchange text messages with the tutor through the private chat room dedicated to his/her practical session.

### *Implementation*

To ensure real-time text transmission over the Internet, the instant messaging system has been developed using JavaEE technologies, and stands on the WebSocket specification and protocol to ensure client-server communication between the users' browser and the Enterprise JavaBeans hosted by the middleware layer. These server-side software components also ensure access to the various chat rooms of the experiments according to the policy rules described before.

### **An Artifact-Awareness Tool for Shared Practical Sessions**

In the context of a collaborative practical activity, tutors and learners must be aware of what others are doing on the apparatus involved in the experiment so that they can act accordingly. If synchronous communication tools (such as the instant messaging system presented before) are often used to let people inform others about the actions they carried out, these systems are not designed to deliver any feedback about what happens on the remote apparatus when an action is executed. As no standards are dedicated to such objectives, remote video surveillance and monitoring tools based on audio/video feeds and/or dedicated sensors are sometimes used to get this feedback (Kostulski and Murray 2011; Lowe et al. 2013; Nickerson et al. 2007). These tools provide a live view of the status of the remote laboratory, but it is very difficult, even nearly impossible, to correlate this status with the actions that have been carried out by the users involved in the collaborative work. In addition, in the specific context of computer education where no physical changes occur on the computer, audio and video feeds become worthless.

Another approach, namely the artifact awareness, brings an alternative to support awareness during collaborative experiments (Tee et al. 2009); these authors define artifact awareness as “one person's up-to-the-moment knowledge of the artifacts and tools that other distributed people are using as they perform their individual, ongoing work” (Tee et al. 2009, p. 678). In the context of Lab4CE, a person engaged in a practical session should be aware of (i) who is working on the same experiment, and (ii) what other people are doing, especially in case of a collaborative work.

### *Design and Illustration*

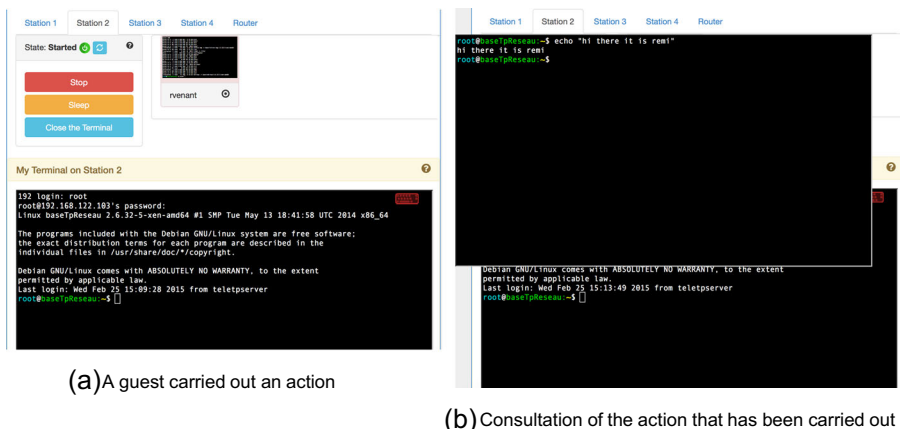
The user block illustrated at the bottom right corner of Fig. 5 suggests a minimal operationalisation of the social presence theory (Lowenthal and Dunlap 2014), defined as “the degree to which a person is perceived as a 'real person' in mediated



communication” (Gunawardena and Zittle 1997, p.4), and known to increase the level of understanding when two distant people have to talk with each other (Barrow 2010). This component lists the learners and tutors involved in the experiment the authenticated user is working on and displays, for each of them, both their role and their connection status. Swan and Shih (2005) investigated that “instructor social presence had a significantly greater impact on perceived learning from online discussions when compared with the impact of student social presence” (Pollard et al. 2014); within our interface, a learner is depicted through a conventional user icon whereas a tutor is represented with an education hat (see Fig. 5). In addition, the user connection status is displayed using a two-coloured icon; a green icon represents a connected user, and a red icon denotes a disconnected user. These visualization artifacts enable quick interpretation and help learners to easily identify peers being working on the same experiment, as well as tutors currently connected to the system.

The user component is also the starting point to initiate a collaborative work. Indeed, through the menu associated to each user, learners are able to invite one or several connected peers and tutors to their practical session; in Fig. 5, the authenticated user *Julien Broisin* is going to invite *Rémi Venant* to join his practical session. The other option *View practical session* offers read access to the practical session of the matching user (but only if the request is accepted by the user).

When several users work together on the same virtual resource, the partners’ Terminal windows appear as thumbnails within the main panel of the application. In the scenario of Fig. 6a where the users *Julien* and *Rémi* are working together on the same resource *Station 2*, *Rémi*’s Terminal shows up within *Julien*’s interface; *Julien*’s Terminal also appears as a thumbnail within *Rémi*’s interface. This artifact allows users to get aware of who is working on what. Besides, when one of the users involved in the collaborative session carries out an action on a given resource, the matching thumbnail is surrounded by a light red-coloured line (see *Rémi*’s thumbnail in Fig. 6a); this visual awareness artifact notifies the other users that an action has been performed by someone else on the remote resource. Finally, by placing their mouse over a Terminal thumbnail, users are able to consult what actions are being carried out by others, and what outputs are returned back by the remote virtual resource (see Fig. 6b).



**Fig. 6** The artifact-awareness tool for shared practical sessions

## *Implementation*

The awareness feature offering transparency about what others are doing on a resource is not a screenshot representing the user's Terminal at a given moment; instead, the tool displays as a live feed what's happening into the user's Terminal. The middleware layer acts as a proxy between users and virtual resources, it is thus able to capture the interactions between those entities. The matching data are then broadcasted, in real-time, to users according to their role within the practical session associated to the virtual resource involved in a given interaction; only the owner and the guests (including tutors) of the practical session will receive the data.

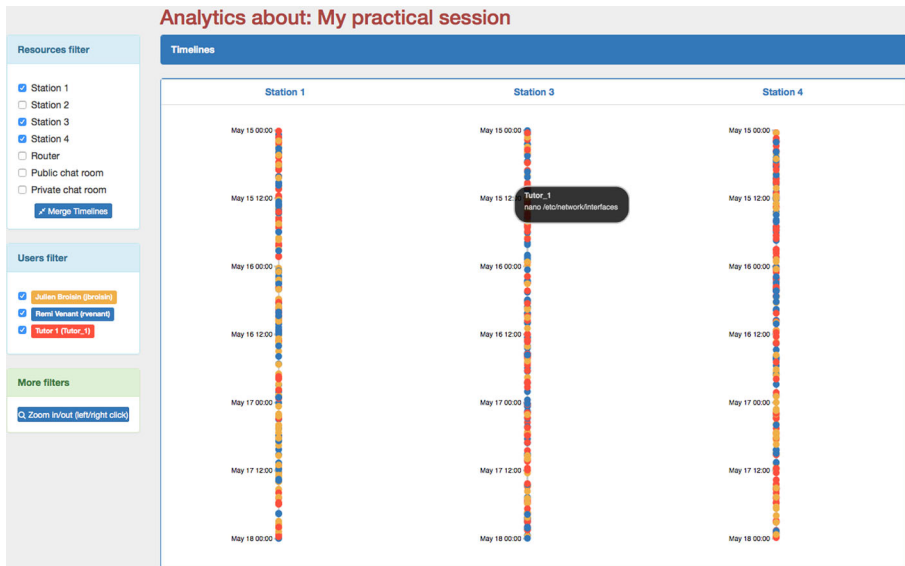
We designed in this section several awareness artifacts supporting collaborative practical work. These proposals make users aware, in real-time, of what is happening on the virtual resources, and facilitate the correlation between events occurring on the remote lab and activities of the users involved in the collaborative process. Compared to other solutions based on virtual network computing (Leproux et al. 2013; Xu et al. 2012), virtual reality (Peña-Ríos et al. 2012) or synchronisation of the lab status between all users (Jara et al. 2009), our tool does not require the installation of an additional software on the users' host, allows to record in detail actions carried out on the virtual resources (see next section), and combines awareness of activities performed simultaneously by several users with ease of use.

## **The Learning Analytics Tool**

In remote learning settings, tutors need to understand students' actions so as to efficiently adjust their tutoring strategy. Some studies addressed this problem by delivering tools that enable tutors to visualize multiple indicators about students' activities, including students' performances (Mazza and Dimitrova 2007), curriculum and productions (Lekira et al. 2012), or learning styles (Bousbia et al. 2009). All these data reduce the time spent by tutors to analyse and react to students' actions and productions.

## *Design and Illustration*

Within Lab4CE, the middleware layer comprises a learning analytics store in which users' activities are recorded, including both commands executed through a Terminal window and messages posted into the instant messaging system. On the basis of these information, we designed a learning analytics tool so as to make these data meaningful to tutors and to learners as well. This tool is illustrated in Fig. 7 and allows users to visualize various information about the experiments they are involved in. Once a given practical session has been selected, users can visualize, for each virtual resource and through a timeline, all activities carried out by the users involved in this session; the details of a command (i.e., the date, the string, and the optional output) can be visualized by putting the mouse over the matching node of the timeline. As shown in Fig. 7, a timeline node is coloured so as to enable quick identification of the matching user; in case of collaborative work, this visual artifact allows to easily identify the most/less active users. Other filtering features allow to visualize activities carried out on a specific resource and/or by a given user only, or to zoom on a period of time by placing the dedicated tool (see bottom left corner of Fig. 7) at



**Fig. 7** The learning analytics tool

the right place on the target timeline. Two other timelines can also be displayed to visualize the messages posted through the synchronous communication system; one of them allows to consult the posts of the public room associated with the experiment, whereas the other reflects exchanges that took place in the private room associated with the selected practical session.

The learning analytics tool supports tutors in various tasks such as monitoring of students' activities, (a)synchronous guidance and assistance, evaluation of learners' performances, or identification of learners who face challenges. As tutors also benefit from their own practical session, they are provided with the opportunity to record a near-perfect session into the store that can be reused by the tool so as to demonstrate to students the best solution for a given experiment. In addition, learners are able to find out whether a specific problem has been solved. In the scenario of the previous section where the users *Julien* and *Rémi* worked together on the same practical session to properly configure the virtual resource *Station 2*, they can visualize, through the timeline of *Station 2*, the actions carried out by their partner; then, according to the actions' outputs returned back by the remote resource and available within the tool, they can deduce if a given command was successful or not.

### Implementation

The learning analytics store has been implemented by a relational database where a chat post is characterised by its author, a timestamp and the content of the message; a command, within the database, is described by the virtual resource on which it has been carried out, the user, a timestamp, the string representing the command or program, and the matching output produced by the virtual equipment. All interactions between users (i.e., chat posts) and between users and virtual resources (i.e., commands) are recorded

into the store. Here again, the authorizations granted to users to review a practical session rely on role-based policy rules: tutors can browse the history of all the practical sessions of an experiment, whereas learners can review their own practical session, those in which they were invited, and the tutors' sessions if any such sessions exist; they can review peers' sessions after their approval.

## The Remote Laboratory Management Dashboard

Another analytics tool is made available to tutors: the management-oriented dashboard enables the monitoring, from a computational point of view, of the learners' virtual resources. For a given experiment, tutors can select the practical session of a specific learner and visualize, for each virtual equipment, various information (see Fig. 8): the status of the resource, the date when it has been created (i.e., the date when the practical session has been started), the "hardware" configuration (including the number of virtual CPUs, the amount of random-access memory and the size of the hard disk), the image name of the operating system, or the list of the network interfaces together with their logical configuration. Tutors can use these data to initiate a chat session with learners if something seems wrong on one or several resources under their responsibility.

The dashboard currently available is based on Horizon (Kumar et al. 2014), the canonical implementation of OpenStack's Dashboard. Even if this ready-to-use tool gave us the opportunity to quickly and easily build a management dashboard, it has been especially designed for cloud monitoring and does not provide the fine-grained information (e.g., the software embedded in a virtual resource, the network routing tables, etc.) required to build learning- and tutoring-oriented analytics tools. Therefore, some work is in progress to implement our own dashboard on the basis of standard supervision protocols and approaches such as the IETF's Simple Network Management Protocol (Case et al. 1990) or the DMTF's Common Information Model (DMTF 2012)

The screenshot shows a web interface for managing virtual resources. At the top, there is a navigation bar with the user name 'Julien Broisin : The basics of Internet Protocol (IP)' and a 'Sign Out' button. Below this, a dropdown menu lists 'Practical sessions:' with two entries: 'Julien Broisin : The basics of Internet Protocol (IP)' (checked) and 'Remi Venant : The basics of Internet Protocol (IP)'. A 'Filter selections' button is also present.

Below the sessions list, there is a section titled 'Select a period of time to query its usage:'. It includes 'From:' and 'To:' input fields with dates '2015-05-10' and '2015-05-22' respectively, and a 'Submit' button. A note states 'The date should be in YYYY-mm-dd format.' Below this, summary statistics are shown: 'Active Instances: 5 Active RAM: 2GB This Period's VCPU-Hours: 23.88 This Period's GB-Hours: 23.88'.

The main part of the dashboard is a table titled 'Usage' with a 'Download CSV Summary' button. The table has the following columns: Instance Name, Status, Created, VCPUs, Disk, RAM, IP Address, and Image Name. It contains five rows of data for 'Station 1' through 'Station 4' and a 'Router' instance. All instances are in a 'Running' state and were created on '05/20/2015'. Each station has 1 VCPU, 1GB Disk, and 512MB RAM. The Router also has 1 VCPU, 1GB Disk, and 512MB RAM. The IP addresses and image names vary slightly between instances.

At the bottom of the table, it says 'Displaying 5 items'.

Instance Name	Status	Created	VCPUs	Disk	RAM	IP Address	Image Name
Station 1	Running	05/20/2015	1	1GB	512MB	eth1: 192.168.1.2	cirros-0.3.3-x86_64
Station 2	Running	05/20/2015	1	1GB	512MB	eth1: 192.168.1.3	cirros-0.3.3-x86_64
Station 3	Running	05/20/2015	1	1GB	512MB	eth1: 192.168.2.2	cirros-0.3.3-x86_64
Station 4	Running	05/20/2015	1	1GB	512MB	eth1: 192.168.2.3	cirros-0.3.3-x86_64
Router	Running	05/20/2015	1	1GB	512MB	eth1: 192.168.1.1, eth2: 192.168.2.1	cirros-0.3.3-x86_64

Fig. 8 The management dashboard

that bring very detailed information, and to adopt appropriate visualization technics allowing tutors to quickly identify misconfigured equipments and to make the right instructional decisions.

## Exploratory Study

### Goals Specification

The Lab4CE environment has been experimented to investigate whether (i) the system has a positive impact on students' engagement in practical learning of the Linux operating system, and (ii) a correlation can be established between students' activity on the system and their learning achievement.

### Context and Design

The study involved 139 students enrolled in a course entitled "Introduction to computer systems" and included in the first semester of a conventional face-to-face computer science curriculum dispensed at the Computer Science Institute of Technology (CSIT) of the University of Toulouse, France. The majority of participants were men (123 men and 16 women), which reflects the distribution of CSIT students, with a mean age of 19.64. The pedagogical objectives of this course consist in learning some basic commands of the Linux operating system: learners must be able to (1) create, modify, delete, and move files and folders, (2) understand and manage the concept of process, and (3) write Shell scripts that facilitate the administration and automation of tasks on this operating system. To reach these objectives, one specific practical activity is proposed to students per week, and each student has to upload a report about the given activity on Sunday on the institution's learning management system (i.e., a Moodle server); late reports' submissions are also allowed. In addition, as students are trained on Linux during this course only (Windows© is used to run the practical activities of all other disciplines), an initial practical activity aims at teaching them how to install the Linux operating system on their own computer to make them able to practice Linux more often than just during this course: learners are taught how to install the software VirtualBox<sup>14</sup> on a Windows©-based computer and how to deploy a virtual machine running Linux within this software.

The exploratory study tackled only the first two points above, i.e., the management of folders, files and processes. Four practical activities were related to these concepts and sequenced as follows: the first session introduced folders management (with the commands *cd*, *ls*, *pwd*, *mkdir*, *rmdir*, and *cp-R*), the second session dealt with the management of files (with the commands *cat*, *touch*, *less*, *cp*, *mv*, *rm*, *wc* and *nl*), the third session looked at objects' rights and permissions (with the commands *chmod*, *umask* and *getfacl* for advanced permissions settings), and the last session was dedicated to the management of processes (with the commands *ps*, *pstree*, *bg*, *fg*, and *jobs*), redirections and piping.

<sup>14</sup> <https://www.virtualbox.org/>

To provide students with the opportunity to work on these topics, we designed a simple experiment composed of a single Linux computer so that each student could access his/her own virtual machine from anywhere, at any time, using any device connected to the Internet. We also presented the Lab4CE's rich learning interface to learners during the first 10 min of the session dedicated to folders management (i.e., the first of the four sessions during which students could use Lab4CE): a 5-minute talk focused on the main Lab4CE's objectives and features, and a 5-minute presentation supported by a videoprojector exposed to learners how to use the graphical user interfaces. During this session, students were also asked if they deployed a virtual Linux-based machine on their own computer. The URL offering access to the system has been integrated into the matching Moodle space. Let us note that *we did not force students to use Lab4CE, they used the system only if they wanted to*.

To measure the correlation between students' activity on the system and their learning achievement, we analysed their performance at the final academic achievement test which took place 2 weeks after the last session. This test consisted of a 45 min multiple-choice quiz collaboratively created by all the teachers involved in the course. *Students' performance* was calculated by extracting the score they received at the 23 questions related to the topics learnt during the four sessions of the exploratory study, the maximum score for each question being 100.

### First Goal - Results and Analysis

Statistics about the usage of the framework appear on Table 3. Seventy one students created their own virtual machine, and each of them opened almost 7 sessions that lasted about 40 minutes, for a mean count of commands per virtual machine higher than 770. Interesting data are the days of the week where students used to connect to the system. They mainly worked during week-end, just before the report should be uploaded. The students also used the system on Monday, for late submissions, even if they were physically present within the institution and could work on "real" computers. That suggests a positive effect of Lab4CE on learners' engagement (as referred to in the introduction of this paper) when they come to experiment system administration: more than 50 % of the students used a virtual machine made available through the Lab4CE environment, whereas only 25 % of them (i.e., 31 students) installed a virtual Linux-based machine on their own computer.

From the data of Table 3, we also analysed the collaborative work between students. Only 75 messages were posted on the public chat room of the experiment, most of them being without pedagogical interest. Only two students worked collaboratively on the same virtual machine and exchanged 7 posts within the matching private chat room. These results can be explained by the fact that (1) students had to upload their own report for a given activity, (2) students are not used to work collaboratively when they are not physically together, and (3) the practical activities proposed in this course were not advanced enough to require the help of peers.

### Second Goal - Results and Analysis

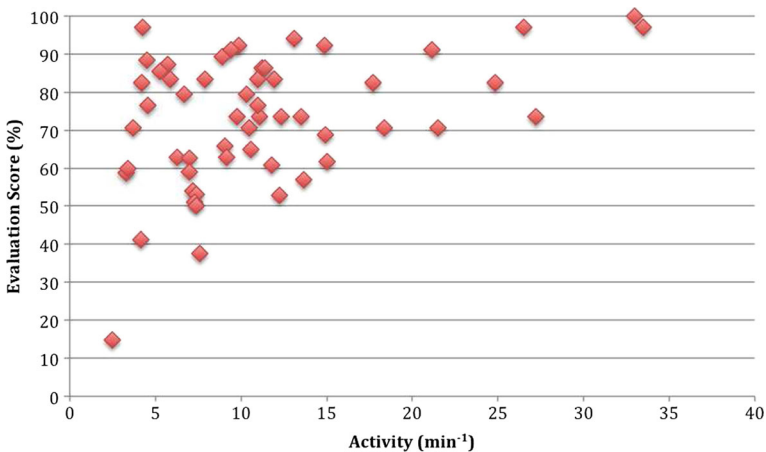
If almost all of the 139 students participated in the mandatory academic achievement test, the analysis exposed in Fig. 9 only integrates students who have taken the test

**Table 3** Statistics of the exploratory study

Measured data	Number of items
Number of virtual machines	71
Number of sessions (with 1 h as a separator)	477
Mean count of sessions per VM	6.7
Mean duration per session (minutes)	39
Mean count of commands per VM	772
Days of the week where students are active	Saturday (26 %), Sunday (16 %), Monday (33 %)
Number of chat messages in the public room	75
Mean count of collaborations per experiment	0.0141
Mean count of messages in a private room	7

(some of the students dropped out of the curriculum or were not present when the test was carried out) and who have used the Lab4CE system at least once. Also, if the average score of the students to a question was lower than the first quartile or greater than the third quartile, the matching question was not integrated into our analysis. Once these filtering rules have been applied, 55 students and 17 questions were taken into account in our analysis.

The *activity* of a student  $i$  has been defined as  $A_i = \frac{I_i}{S_i}$ , where  $I_i$  is the number of inputs (i.e., the number of commands and programs carried out on a virtual machine) of student  $i$ , and  $S_i$  represents the time the student  $i$  has spent into the system. The scatter plot of Fig. 9 shows a significant positive Pearson correlation (i.e., Pearson's  $r$ ) between students' activity and students' achievement:  $r=0.41$ ,  $p=.002$ . It highlights a tendency for students that achieved more than 70 % of right answers at the final academic achievement test to produce more in the system. It also reveals that the activity level in the system could be a good predictor of students' achievement; this indicator could thus be used to detect

**Fig. 9** Correlation between students' activity and performance



difficulties/easiness of students well before the achievement test, and then to personalize the learning scenario accordingly.

## Discussion

The Lab4CE environment currently supports human tutoring through the synchronous communication tool, but also through the opportunity offered to users to share a practical session; help can thus be offered by peers, and/or requested if a tutor happens to be online. However, in the current implementation, one limitation of the system is the automatic and intelligent tutoring to bring to students when they have difficulties doing the tasks they are required in a particular session.

To overcome this shortcoming, one approach consists in reusing the data recorded into the learning analytics store. Until now, these data have been exploited by the dashboard exposed in Fig. 7 only, where users are able to see what happened during a particular practical session. However, more advanced functionalities can be easily developed on the basis of these data. One intelligent tutoring capability includes the setting up of a notification system in order to alert online tutors that one or more learners need immediate help; notifications will appear on the tutors' learning interface when several consecutive unsuccessful commands are executed by a given user (in this case, the *exit status* returned by any well-behaved Unix command will be used to detect the wrong ones), but also when difficulties are expressed by users through the chat system (here, the real-time full-text search and analytics capabilities of Elasticsearch will be reused). Also, useful hints and procedural help such as the manual pages of the Linux commands can be automatically recommended to learners and/or displayed within their interface when the system detects difficulties. Thanks to the layered architecture we adopted, the above functionalities will be integrated as intelligent tutoring services into the Lab4CE's middleware layer.

## Conclusions and Future Works

We introduced in this paper the Lab4CE environment, a remote laboratory for computer education improving existing virtualization tools and technologies by supporting instructors and learners during practical activities. Lab4CE brings significant educational assets through a set of scaffolding tools and interfaces aiming at offering the best user experience as possible: (1) the authoring tool provides an intuitive interface to build realistic experiments while hiding the complexity of the underlying technologies, thus encouraging adoption of the framework by instructors; (2) collaboration tools and awareness artifacts intended for learners promote their engagement in remote practical activities, as the quality of peer collaboration is one of the mediating factors that explain the effectiveness of remote labs (Corter et al. 2011); (3) learning analytics tools and dashboards based on interactions between users and between users and virtual resources, but also on management information gathered from the remote lab, enable tutors to better understand learners' activities and to make appropriate tutoring decisions.

The Lab4CE environment also contributes to the improvement of hands-on lab sessions by making them augmented spaces for productive interactions between students, and between students and tutors.

The exploratory study suggests an impact of students' activity in the system on students' performance at the academic achievement test. This first experimentation, carried out as soon as a Lab4CE prototype has been available, represents a first step towards the clear identification of the scaffolding tools that engage learners in remote practical activities. To reach this goal, other experimentations comprising various experimental conditions will be conducted next year at a larger scale in two distinct courses including almost 200 students: a first-year course about system administration, and a second-year course about network administration. The large amount of data collected during these experimentations should bring food to learning analytics techniques to lead to a better comprehension and knowledge of learning processes, and should offer the opportunity to design and develop adaptive and personalized functions dedicated to online practical works.

Finally, our proposal stands on a three tiers architecture that can be easily upgraded to N tiers architecture to support learning of computer science at scale. As the lab layer relies on a cloud environment which natively supports scalability and availability through the federation of multiple clouds, the middleware layer only needs to be supplemented by new features. A load balancing software should be integrated at the top of this layer, so that requests from the learning tools could be distributed to the 'best' node. This straightforward improvement will naturally lead our efforts towards the integration of computer science practical sessions into massive open online courses, and to encourage collaboration between students and peer tutoring through the integration of pedagogical strategies such as "gamification" that consists in integrating game design elements in non-game learning environments to increase students' engagement and motivation (Deterding 2012; Dominguez et al. 2013).

**Acknowledgments** The authors wish to thank their colleague Dr. Gaëlle Molinari for her insights and expertise that greatly assisted the research, and for her comments on this manuscript.

## References

- Barkat, A., Santos, dos, A. D., & Ho, T. T. N. (2014). Open Stack and Cloud Stack: Open Source Solutions for Building Public and Private Clouds. In *Proceedings of The 16th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, (pp. 429–436). Timisoara: IEEE.
- Barrow, T. (2010). Social Presence Theory. Retrieved from <http://blog.timebarrow.com/2010/03/social-presence-theory/>
- Bell, T., Urhahne, D., Schanze, S., & Ploetzner, R. (2010). Collaborative inquiry learning: models, tools, and challenges. *International Journal of Science Education*, 32(3), 349–377.
- Bist, M., Wariya, M., & Agarwal, A. (2013). Comparing delta, open stack and Xen Cloud Platforms: A survey on open source IaaS. In *Proceedings of The 3rd International Advance Computing Conference (IACC)*, (pp. 96–100). Ghaziabad: IEEE.
- Bochicchio, M., & Longo, A. (2009). Hands-on remote labs: collaborative web laboratories as a case study for IT engineering classes. *IEEE Transactions on Learning Technologies*, 2(4), 320–330.
- Bonner, S., Pulley, C., Kureshi, I., Holmes, V., Brennan, J., & James, Y. (2013). Using OpenStack to improve student experience in an H.E. environment. In *Proceedings of The Science and Information Conference (SAI)*, (pp. 888–893). London: IEEE.

- Bousbia, N., Labat, J. M., Rebai, R. & Balla, A. (2009). Indicators for deducting the learner's learning style: case of the navigation typology indicator. In *Proceedings of The 9th IEEE International Conference on Advanced Learning Technologies (ICALT)*, (pp. 385–389). Riga: IEEE.
- Case, J., Fedor, M., Schoffstall, M., & Davin, J. (1990). A Simple Network Management Protocol (SNMP) [RFC 1157]. Retrieved from <http://tools.ietf.org/html/rfc1157>.
- Corter, J. E., Esche, S. K., Chassapis, C., Ma, J., & Nickerson, J. V. (2011). Process and learning outcomes from remotely-operated, simulated, and hands-on student laboratories. *Computers & Education*, 57(3), 2054–2067.
- de Jong, T., Linn, M. C., & Zacharia, Z. C. (2013). Physical and virtual laboratories in science and engineering education. *Science*, 340(6130), 305–308.
- DeLong, K., Harward, V. J., Bailey, P., Hardison, J., Kohse, G., & Ostrosky, Y. (2010). Three online neutron beam experiments based on the iLab Shared Architecture. In *Proceedings of The 2010 I.E. Global Engineering Education Conference (EDUCON)*, (pp. 145–150). Madrid: IEEE.
- Deterding, S. (2012). Gamification: designing for motivation. *Interactions*, 19, 14–17.
- DMTF Distributed Management Task Force. (2012). Virtual Networking Management White Paper [DSP2025]. Retrieved from [http://www.dmtf.org/sites/default/files/standards/documents/DSP2025\\_1.0\\_0\\_ja.pdf](http://www.dmtf.org/sites/default/files/standards/documents/DSP2025_1.0_0_ja.pdf).
- Dominguez, A., Saenz-de-Navarrete, J., de-Marcos, L., Fernandez-Sanz, L., Pagés, C., & Martinez-Herraz, J. J. (2013). Gamifying learning experiences: practical implications and outcomes. *Computers & Education*, 63, 380–392.
- Fredericks, J. A., Blumenfeld, P. C., & Paris, A. H. (2004). School engagement: potential of the concept, state of the evidence. *Review of Educational Research*, 74, 59–109.
- Gillet, D., de Jong, T., Sotirou, S., & Salzmann, C. (2013). Personalised Learning Spaces and Federated Online Labs for STEM Education at School: Supporting Teacher Communities and Inquiry Learning. In *Proceedings of The 2013 I.E. Global Engineering Education Conference (EDUCON)*, (pp. 769–773). Berlin: IEEE.
- Govaerts, S., Cao, Y., Vozniuk, A., Holzer, A., Zutin, D. G., SanCristobal Ruiz, E., Bollen, L., Manske, S., Faltin, N., Salzmann, C., Tsourlidaki, E., & Gillet, D. (2013). Towards an Online Lab Portal for Inquiry-based STEM Learning at School. In *Proceedings of The 12th International Conference on Web-based Learning (ICWL)*, (pp. 244–253). Kenting: LNCS.
- Gunawardena, N., & Zittle, F. (1997). Social presence as a predictor of satisfaction within a computer mediated conferencing environment. *American Journal of Distance Education*, 11(3), 8–26.
- Hardison, J. L., DeLong, K., Bailey, P. H., & Harward, V. J. (2008). Deploying interactive remote labs using the iLab shared architecture. In *Proceedings of Frontiers in Education Conference (FIE)*, (pp. S2A-1-S2A-6). Cambridge: IEEE.
- Harward, V. J., del Alamo, J., Choudhary, V., DeLong, K., Hardison, J., Lerman, S., Northridge, J., Talavera, D., Varadharajan, C., Wang, S., Yehia, K., & Zych, D. (2004). iLab: A scalable architecture for sharing online experiments. In *Proceedings of The International Conference on Engineering Education (ICEE)*, (9 pages). Gainesville: iNEER.
- Harward, V. J., del Alamo, J. A., Lerman, S. R., Bailey, P. H., Carpenter, J., DeLong, K., Felknor, C., Hardison, J., Harrison, B., Jabbour, I., Long, P. D., Mao, T., Naamani, L., Northridge, J., Schulz, M., Talavera, D., Varadharajan, C., Wang, S., Yehia, K., Zbib, R., & Zych, D. (2008). The iLab shared architecture: a web services infrastructure to build communities of internet accessible laboratories. *Proceedings of the IEEE*, 96(6), 931–950.
- Jara, C. A., Candelas, F. A., Torres, F., Dormido, S., Esquembre, F., & Reinoso, O. (2009). Real-time collaboration of virtual laboratories through the internet. *Computers & Education*, 52(1), 126–140.
- Khan, R. H., Ylitalo, J., & Ahmed, A. S. (2011). OpenID Authentication As A Service in OpenStack. In *Proceedings of The 7th International Conference on Information Assurance and Security (IAS)*, (pp. 372–377). Malacca: IEEE.
- Kostulski, T., & Murray, S. (2011). Student feedback from the first national sharing trial of remote labs in Australia. In *Proceedings of The 8th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, (pp. 203–211). Brasov: International Association of Online Engineering.
- Kriz, P. (2014). Survey on open source platform-as-a-service solutions for education. In *Proceedings of The 18th International Database Engineering & Applications Symposium (IDEAS)*, (pp. 176–184). Porto: ACM.
- Kroeker, J. L. (2009). The evolution of virtualization. *Communications of the ACM*, 52(3), 18–20.
- Kumar, R., Gupta, N., Charu, S., Jain, K., & Jangir, S. K. (2014). Open source solution for cloud computing platform using OpenStack. *International Journal of Computer Science and Mobile Computing*, 3(5), 89–98.

- Lekira, A., Després, C., Jacoboni, P., & Py, D. (2012). An indicator-based approach to promote the effectiveness of teachers' interventions. In *Proceedings of The 11th International Conference on Intelligent Tutoring Systems (ITS)*, (pp. 517–523). Chania: LNCS.
- Leproux, P., Barataud, D., Bailly, S., & Nieto, R. (2013). LABENVI (laboratoire d'enseignement virtuel). Présentation et analyse des nouveaux usages pour la conduite de travaux pratiques à distance. *Interfaces numériques*, 2(3), 453–467.
- Lowe, D. (2013). Integrating reservations and queuing in remote laboratory scheduling. *IEEE Transactions on Learning Technologies*, 6(1), 73–83.
- Lowe, D., Berry, C., Murray, S., & Lindsay, E. (2009). Adapting a Remote Laboratory Architecture to Support Collaboration and Supervision. In *Proceedings of The 6th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, (pp. 103–108). Bridgeport: International Association of Online Engineering.
- Lowe, D., Murray, S., Lindsay, E., & Liu, D. (2009b). Evolving remote laboratory architectures to leverage emerging internet technologies. *IEEE Transactions on Learning Technologies*, 2(4), 289–294.
- Lowe, D., Newcombe, P., & Stumpers, B. (2013). Evaluation of the use of remote laboratories for secondary school science education. *Research in Science Education*, 43(3), 1197–1219.
- Lowenthal, P. R., & Dunlap, J. C. (2014). Problems measuring social presence in a community of inquiry. *E-Learning and Digital Media*, 11(1), 19–30.
- Ma, J., & Nickerson, J. V. (2006). Hands-On, Simulated, and Remote Laboratories: A Comparative Literature Review. *ACM Computing Surveys*, 38(3), Article 7.
- Mazza, R., & Dimitrova, V. (2007). Coursevis: a graphical student monitoring tool for facilitating instructors in web-based distance courses. *International Journal in Human-Computer Studies*, 65(2), 125–139.
- Morell, L., & Jiang, C. (2015). Using ShellInABox to improve web interaction in computing courses. *Journal of Computing Sciences in Colleges*, 30(5), 61–66.
- Nickerson, J. V., Corter, J. E., Esche, S. K., & Chassapis, C. (2007). A model for evaluating the effectiveness of remote engineering laboratories and simulations in education. *Computers & Education*, 49(3), 708–725.
- Niederstaetter, M., Klinger, T., & Zutin, D. G. (2010). An image processing online laboratory within the iLab shared architecture. *International Journal of Online Engineering (iJOE)*, 6(2), 37–40.
- Orduña, P., Rodriguez-Gil, L., López-de-Ipiña, D., & Garcia-Zubia, J. (2012). Sharing the remote laboratories among different institutions: A practical case. In *Proceedings of The 9th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, (pp. 1–4). Bilbao: IEEE.
- Orduña, P., Almeida, A., López-De-Ipiña, D., & Garcia-Zubia, J. (2014). Learning Analytics on federated remote laboratories: tips and techniques. In *Proceedings of 2014 I.E. Global Engineering Education Conference (EDUCON)*, (pp. 299–305). Istanbul: IEEE.
- Peña-Ríos, A., Callaghan, V., Gardner, M., & Alhaddad, M. J. (2012). Remote mixed reality collaborative laboratory activities: Learning activities within the InterReality Portal. In *Proceedings of The International Conferences on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, (pp. 362–366). Macau: IEEE.
- Pinkwart, N., Harrer, A., & Kuhn, M. (2010). Process support for collaborative inquiry learning. *Research and Practice in Technology Enhanced Learning*, 5(3), 185–203.
- Pollard, H., Minor, M., & Swanson, A. (2014). Instructor Social Presence within the Community of Inquiry Framework and its Impact on Classroom Community and the Learning Environment. *Online Journal of Distance Learning Administration*, 17(2).
- Richter, T., Boehringer, D., & Jeschke, S. (2011). LiLa: a European project on networked experiments. *Automation Communication and Cybernetics in Science and Engineering*, 2009(2010), 307–317.
- Röhrig, C., & Jochheim, A. (2001). Group-Based learning using a remote laboratory. In *Proceedings of The 2001 American Control Conference (ACC)*, (pp. 1153–1154). Arlington: IEEE.
- Saab, N., van Joolingen, W., & van Hout-Wolters, B. (2012). Support of the collaborative inquiry learning process: influence of support on task and team regulation. *Metacognition and learning*, 7(1), 7–23.
- Sahoo, J., Mohapatra, S., & Lath, R. (2010). Virtualization: A survey on concepts, taxonomy and associated security issues. In *Proceedings of The 2nd International Conference on Computer and Network Technology (ICCNT)*, (pp. 222–226). Bangkok: IEEE.
- Salzmann, C., Govaerts, S., Halimi, W., & Gillet, D. (2015). The Smart Device specification for remote labs. In *Proceedings of The 12th International Conference on Remote Engineering and Virtual Instrumentation (REV)*, (pp. 199–208). Bangkok: IEEE.
- Sancristobal, E., Castro, M., Harward, J., Baley, P., DeLong, K., & Hardison, J. (2010). Integration view of Web Labs and Learning Management Systems. In *Proceedings of 2010 I.E. Global Engineering Education Conference (EDUCON)*, (pp. 1409–1417). Madrid: IEEE.

- Singer, R. S., Margaret, H. L., & Schweingruber, H. A. (2005). *America's Lab Report: Investigations in High School Science*. Committee on High School Science Laboratories: Role and Vision, National Research Council.
- Swan, K., & Shih, L. F. (2005). On the nature and development of social presence in online course discussions. *Journal of Asynchronous Learning Networks*, 9(3), 115–136.
- Tee, K., Greenberg, S., & Gutwin, C. (2009). Artefact awareness through screen sharing for distributed groups. *International Journal of Human-Computer Studies*, 67(9), 677–702.
- Van Joolingen, W. R., de Jong, T., Lazonder, A. W., Savelsbergh, E., & Manlove, S. (2005). Co-Lab: research and development of an on-line learning environment for collaborative scientific discovery learning. *Computers in Human Behavior*, 21, 671–688.
- Wang, X., Hembroff, G. C., & Yedica, R. (2010). Using VMware VCenter Lab Manager in Undergraduate Education for System Administration and Network Security. In *Proceedings of The 2010 ACM Conference on Information Technology Education (SIGITE)*, (pp. 43–52). Midland: ACM.
- Wei, H., Rodriguez, J. S., & Tapiador, A. (2014). Enhance OpenStack Access Control via Policy Enforcement Based on XACML. In *Proceedings of The 16th International Conference on Enterprise Information Systems (ICEIS)*, (pp. 283–289). Lisbon: SCITEPRESS.
- Wen, X., Gu, G., Li, Q., Gao, Y., & Zhang, X. (2012). Comparison of open-source cloud management platforms: OpenStack and OpenNebula. In *Proceedings of The 9th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD)*, (pp. 2457–2461). Sichuan: IEEE.
- Xu, L., Huang, D., & Tsai, W. T. (2012). V-lab: a cloud-based virtual laboratory platform for hands-on networking courses. In *Proceedings of The 17th ACM annual Conference on Innovation and Technology in Computer Science Education (ITICSE)*, (pp. 256–261). Haifa: ACM.
- Zhang, Z., Wu, C., & Cheung, D. W. L. (2013). A survey on cloud interoperability: taxonomies, standards, and practice. *SIGMETRICS Performance Evaluation Review*, 40(4), 13–22.
- Zutin, D. G., & Auer, M. E. (2011). Work in progress - Integrating educational online lab platforms around the iLab Shared Architecture. In *Proceedings of Frontiers in Education Conference (FIE)*, (pp. F1G-1-F1G-3). Rapid City: IEEE.
- Zutin, D. G., Auer, M. E., Maier, C., & Niederstatter, M. (2010). Lab2go - A repository to locate educational online laboratories. In *Proceedings of 2010 I.E. Global Engineering Education Conference (EDUCON)*, (pp. 1741–1746). Madrid: IEEE.