A Distributed Online Laboratory System for Distant Learning

Christophe GRAVIER, Jacques FAYOLLE, Jérémy LARDON, and Gérard NOYEL

DIOM laboratory, Jean Monnet University
{christophe.gravier,jacques.fayolle,jeremy.lardon,gerard.noyel}
@univ-st-etienne.fr,
http://www.univ-st-etienne.fr/diom/

Abstract. In this paper, we discuss the use of online laboratories in distant learning. Learners need, obviously, hands-on approaches in order to receive the knownledge on how to use appliances. Being a complement to annotated lectures in distant learning, online laboratories need to be as authentic as possible to increase learners' understanding. In contrast, the state of the art of online laboratories provided in this paper shows a lack of solutions, especially because of challenging computer science issues. Not only must we provide an authentic learning experience to students, we must also orchestrate collaborative learning in the online laboratory (several users acting in a classroom; accessing a shared and remote device).

Key words: E-Learning, Distance Learning, Distributed Architectures, Semantic Web, Computer Collaborative Collaborative Learning, Process Control

1 Introduction

Distance learning have brought to the Web a number of learning tools, making lectures possible where teachers and learners are in different place and/or at different time. Almost all those solutions are generic, as they can address any kind of lectures. They can be web based ([2]), dedicated rich clients (as annotations on slides or video) or even based on MBone[1] toolset ([1]). Some even allow audio annotations ([3]). On one hand, providing media leads to advanced network protocols ([4]). On the other, making the classroom geographically distributed leads to create virtual environments, where we try to recreate the process of learning in a group ([5]). Those solutions, however, are inadequate when learners need hand-ons approaches in order to obtain the knowledge on how devices are supposed to be manipulated. In the process of learning, widely accepted works in cognitive science demonstrate that learners begin to compare situations to already known examples. Only then are they able to build cognitive

[1] Multicast Bone

rules regarding the learning experiences they had : "this is principal means by which knowledge transitions from a declarative form (encoding of examples) to a procedural form (production rules)" ([6]). We can only agree from our own experiences: see, hear, touch, and even smell help in remembering. Enhancement of information technologies for learning by "doing things" has already been proven ([7]). Nevertheless, most universities provide distance learning through lectures but fewer are able to put their hands-on approaches online. That is the reason why we see online laboratories one of the most exciting issue in distance learning today.

Some propositions already cover the marriage between lectures and laboratories in distance learning ([3]). Our proposition aims at going some steps further. Indeed, we think that hands-on approaches should not be put online independently from one another. The major risk is to reinvent the wheel each time a device goes online. This is truly unsatisfactory, especially when the number of devices is large. We propose here a generic factory for online laboratories. In addition, we believe in collaborative learning as reinforcement to learners interactions ([8]). Regarding computer science, the collaborative service must be common for each devices. This way, we would avoid reinventing the wheel each time it is needed. Hence, we present in this paper, a generic, synchronous, collaborative, and online laboratory. The paper is organized as follows. First, state of the art from researches on distant learning and laboratories are briefly discussed. This is followed by the presentation of our online laboratory and the support of "device independancy" (a solution of online laboratory that is not supposed to know the device it supports). Then, we put the stress on collaboration of learners in online laboratories and how our proposal bring solution in computer science. Finally, section 5 concludes.

2 Laboratories in distance learning

2.1 Regarding classification

We identify distance learning as situations where teachers (or their pedagogical materials) and students are not in the same place and/or at the same time. We already discussed the need of laboratories in distance learning. Nevertheless, it cannot be said that all laboratories endorsed the same properties. Three different kind of laboratory are identified, according previous work ([9]):

- *local* laboratory are universities' lab in which student practice hands-on exercises.
- *virtual* laboratory are simulations of an existing laboratory, trying to be as close as possible to the real device.
- *online* laboratory are local laboratory that are accessed via Internet, at different places but at the same time. It can be sum up as "Same time, different place" ([10]). Those are *real* devices remote controlled over Internet.

We argue local laboratory are the best kind of laboratory one can make, for learners' sake. This is because students are faced to the real device and have face

to face tutoral assistance. Of course, local laboratories suffer from drawbacks: learners and teachers must be in the same place, at the same time, in a room sufficiently wide for the learning experience to occur. Those drawbacks lead computer scientists to propose alternative options, for learners/teachers who cannot make it to the local laboratory.

2.2 Virtual and online laboratories

As for the first question, some of current proposals are pure http based. These are LAMP[2]-based solutions, JSP[3] coupled with MySQL Database ([11]) or other Internet technologies. We strongly believe rich clients are more suitable for such applications. Of course, they suffer from a higher development cost, but they are more precise in reproducing "felt-life" ([12]): things that are moving, making particular noises, buttons resistance ([13]), etc.

Next, fall into question the choice between virtual and online laboratories. Severals research works tend to demonstrate that online laboratory are "the ultimate solution"([14]) and simulation is "the old way of doing things" ([15]). Nevertheless, we have a subtle thought as we think simulations are intended to deliver laboratory facilities to the door of the students, when an online laboratory is not a way to look at. Many drawbacks to simulations have already been related ([14]). We strongly believe in two arguments. Firstly, simulation cannot provide an authentic learning experience as, at its best, can only deliver partial representation of the reality of use of the instrument. Then, the limited functions set usually implemented, and the guided scenario associated to the manipulation, lead to an aseptic scenarii of use for learners. We state that it limits what we can call "learning walkthough": learners' ability to try out things by curiosity. If the simulation does not foresee such tries, the response will not satisfy the learner and the system will miss opportunities of teaching.

2.3 Limitation of existing online laboratories : what a next-generation online laboratory should aim

At first, almost all online laboratories are whether www-based and asynchrone *or* collaborative asynchrone. From our point of view, this is not a choice. One prefers a collaborative *and* synchrone instead of a asynchrone and point to point online laboratory. The difficulty to build such a system (so its uncommonness) is due to complexity. Such a laboratory must provide "group awareness" ([16]) that it is still a strong and relevant topic in CSCW[4]. Moreover, online laboratories must take profit from the use of a middleware connecting teachers, learners and devices, to try to adapt device's functionalities towards learners' abilities. In addition, we propose the integration of "Postpone Distance Learning" in online laboratories. When online laboratories are be able to store and replay

[2] Linux Apache MySQL PHP
[3] Java Server Page
[4] Computer Supported Collaborative Work

actions that were performed, teachers can attach additional pedagogical materials to their lectures. Learners would then replay those hands-on approaches at home, reinforcing Interactive Home Learning ([1]). Finally, we strongly believe in genericity for online laboratories. Many of researchers' and industrial propositions aims at bringing online one device. But it is hardly designed for being reused for another systems (and even more in heterogeneous devices "farming"). Reusing architectures of online laboratories must really be taken into account when building an online laboratory as it prevents the architect from reinventing the wheel each time a device goes online. Those four abilities of online laboratories (group learning awareness, user adaptation, postpone distance learning, genericity) are the strong principles that guide us while thinking and building our solution presented in the two following sections.

3 Providing a generic solution for bringing a device online.

3.1 Learning from the past

We started remotely laboratory researches in 2000 ([17], based on a network analyzer[5] we wanted online. A direct TCP/IP application exists and the entire project (with the Graphical User Interface (henceforth GUI)) is coded using C++ and CORBA[6]. The resulting application allowed the network analyzer to be remote controlled over Internet. This gave us full satisfaction as it is used in distance learning between students of different universities, as an online laboratory. Nevertheless, we dropped this implementation in favor of J2EE[7] (see current implementation in section 3.3) due to firewall issues and lack of interoperability in heterogeneous information systems.
This successfull pedagogical experience make us think about putting another device online: an antenna workbench. Of course, unlike the network analyzer, the antenna workbench conveys mechanical experiences (moving antenna and starting/stopping motors). The resulting GUI, however, are close to one another, because the GUI displays the same kind of widgets, whatever the device is (square, rectangle, round or knob buttons, led, curves, objects moving, menus, etc.). Besides, we become aware we were about to reinvent the wheel each time we want another device online. This tends to prove dedicated integration are short term answers but aren't suppose to be reused for other experiment involving other devices. Moreover, as we exploited this solution in our teaching, we understood how authenticity of the device displayed is important. Because students mostly learn from hands-on approaches *how to use* appliances, not *how they work*. As such, it is very important to be as real as possible, because students tend to be lost when put in front of "the real thing". Consequently we can

[5] A network analyzer allows the measurement of module and phase of reflected and transmitted signals of a device.
[6] Common Object Request Broker Architecture
[7] Java 2 Enterprise Edition

say we learn from past experiences that genericity is a major issue for bringing several devices online *and* authenticity is a major factor for learning experience to succeed.

3.2 How to solve devices' independancy using an ontology

Nowadays, laboratories based their experiments on a heterogeneous set of devices. In the context of remote laboratories, aiming at devices' independancy means supplying interoperability tools, in order to get remote laboratories platform able to discuss with any kind of "remote-able" appliances. Such an aim needs to get a formal representation of what a device is, qualifying the appliance with no more and no less details than necessary. To reach that goal, we need a representation of knowledge that allows to conceptualize a specific domain *and* to specialize (instanciate) that domain ([18]). This way, the representation of the knowledge is shared among all devices, and each device goes with a specialization of that domain of knowledge. This requirement of interoperability perfectly fits the definition of ontologies (and its standard specifications known as OWL[8], by the W3C).

We established the ontology of devices one could find in a laboratory. With such an ontology (see 1), we are able to dress the complete GUI of a device.

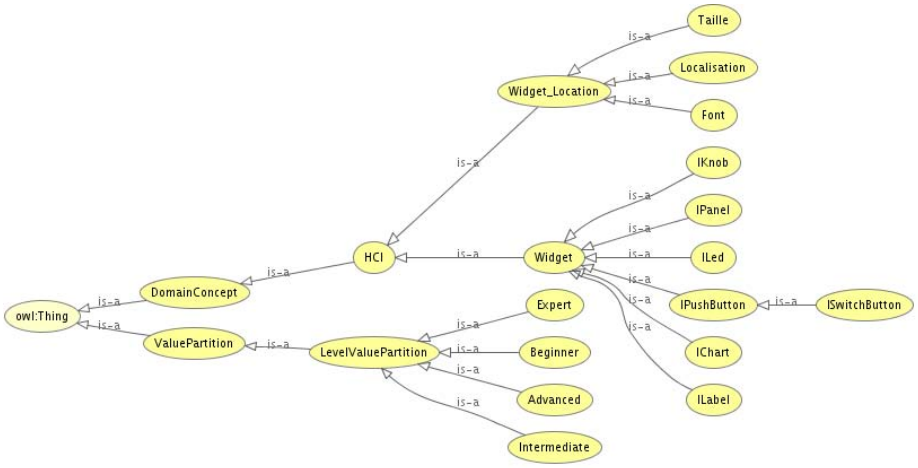

Fig. 1. Laboratory devices' ontology used for bringing our appliances online.

This means the vocabulary is common to all appliances and is instanced once. In the end, we produce a single file in OWL format, one per device. Using ontologies, we are able to describe in OWL format a network analyzer, an antenna workbench and we are about to dress the OWL of an optic fiber stretcher.

[8] Ontology Web Language

3.3 Implementation

Using this OWL file, a stand alone client interpret it to build the representation of the real remote device. It has to be noticed that we try to be as authentic as possible (see section 4.1). The stand alone client uses a MOM[9] above an application server (for authentification, authorization and transactions). We use publish/subscribe paradigm ([19]) to deliver message to all learners in the classroom.
The three possible commands the system can rely are:

- *ASK*, stand for "asking". Such a key word is used when a widget is actioned (a command needs to be performed on the remote appliance). Those messages are unicasted to the instrument.
- *ACK*, stands for "acknowledgment". Because commands performed on a device can last long (if an antenna needs to be moved for example), the device sends back an acknowledgment as a reflex, when receiving any "ASK" message.
- *ANS*, stands for "answering". It is the response from the device corresponding to the stimulus generated when a widget is actioned.

Unlike "ASK" message, which are sends only to the instrument (unicast), the two last sort of messages are "multicasted" to all users in the learning group. In our implementation, the first layer a message enters, is built thanks to JAAS[10]. Our users and roles are stored within an openLDAP server. Besides, logging is performed using PostgreSQL open source database. Finally, the delivery layer is supported by JORAM[11], an ObjectWeb[12] open source message oriented middleware, written in Java. The entire system is hosted in an J2EE application server; the implementation chosen is JOnAS[13], as we were looking for a J2EE certified server.

4 How to maintain collaborative learning experiences.

4.1 Felt-Life and Collaboration awareness embedded within widgets

Our challenge is to avoid reinventing the wheel each time a device goes online. This means avoiding duplicating messaging transportation and authentication or authorization issues as describe earlier (see 4), but also trying to escape making one Java GUI per device online. Not that it is hard work, but GUI "genericity" could save a huge amount of the time required to put an appliance available over Internet. As a consequence, we think about two majors issues for genericity: do not write Java code for the device GUI *and* do not write any Java code for any

[9] Message Oriented Middleware
[10] Java Authentication and Authorization Service
[11] Java Open and Reliable Asynchronous Messaging
[12] http://www.objectweb.org
[13] Java Open Application Server

message received by the device. This means we aim at avoiding writing *any* line of client code to be compiled for any device we put online. The best use case of integration we found was 550 lines of code needed per collaborative application ([20]). When a widget is actioned, let's say a push button, its unique identifier in the device's GUI is associated to the message. This identifier helps, device's side, to get the action that is being requested. The corresponding push button is in a "pushed" state. When the device answer is built, an array holds the list of modifications to make in the GUI. The corresponding method is invoked on the widgets using Java reflection API[14]. In our push button example, upon response of the device, we have to set the push button back to normal state. The device answer sent to all clients will contain a single entry in its body stating that the widget identified by this id must call the "BackToNormal" method, without any argument. Client side, the widget is found via introspection in the swing GUI. Here is intervention of Java 2 reflection API: we search in the Java classpath for the widget's class, a pointer to the method with the signature "setBackToNormal()". When such a pointer is found, it is invoke for widget instance, thus modifying the GUI, without writing any line of code client side. Of course, the use of MOM infers all clients receive the messages in the order they we casted (MOM's casual ordering property).
As for device's GUI, we use an instance of our device ontology (see 3.3). This explains that all the collaborative and non collaborative behavior of widgets are embedded within those widgets, each widget is held responsible to support group awareness. Our custom widgets are composed of three blocs:

- one or severals Swing components (JComponents) for display
- public methods that are available to be called for such a widget (those methods are part of the vocabulary of the ontology used)
- methods enabling felt-life. Those are mainly reflexes in the widgets behavior. They supply media in order to reproducing an authentic learning experience, as real as possible (background sounds, sounds when a widget is actioned (if any), images for the representation of the real appliance, ...).

A caption of our running application illustrates this paragraph (figure 2).

4.2 Regulate access to devices: orchestrate CSCL experiences

[15] All this present work is implemented in our laboratory and being used. Nevertheless, one could argue about collaborative sharing of a single resource and how this "pseudo-concurrent" access (as it is collaboration) is managed. This is the subject of this paragraph. We present here our work that is currently being implemented. It aims at orchestrating CSCL experiences. This is mainly because severals learners are accessing in *real time* and in *a collaborative* way an existing device plugged over Internet. Hence need we a layer being able to orchestrate

[14] http://java.sun.com/docs/books/tutorial/reflect/index.html
[15] computer supported collaborative learning

Fig. 2. High frequency analyzer being used via our application. Caption of our stand alone application, used by learners.

accesses to the device. Our point of view is that we are not in complete concurrent and hostile environment. In fact, computer supported collaborative work is defined for us as several users using a remote resource for a *common* objective. If the objective is not shared among all users, this is not collaborative work any more. That explains why we cannot apply already well known current resource sharing sequencing such as first in first out, round robin or what ever can be used in complete concurrent environment (by analogy of CPU time sharing). This is mainly because the role the user plays in the collaboration is a factor of orchestration. Moreover, orchestrating rules are not supposed to be set forever, so they can be modified in time.

In order to achieve this sequencing, we use a server-side element that is using an ontology to set the strategy of collaboration being used. Our aim is not to support all already known collaborative situations but to supply the possibility to build our own collaborative strategy that will lead the hand-on approach, using learning scenario and context aware elements. This way, one can assemble and describe one collaborative strategy in an ontology (still OWL format) and can load/unload it in our gateway above the device. For example, it can address such following statements : "I give priority access to the learner that is late in the learning scenario", or "I give access to learners that have the lesser number of actions made but the user that plays teacher role can make preemptive accesses". Our purpose is to get a orchestrating unit that can be parametered using an ontology (that describes the collaborative strategy to play in the learning experience). Ideally, everyone should be able to dress his own collaborative strategy ontology to be used with the learning manipulation.

5 Conclusion

In this paper, we provide a state of the art about online laboratories for distant learning. We also argue that bringing appliances online independantly from one another is not a satisfactory response to this challenge. That is the reason why we propose a generic solution, mostly based on ontologies and distributed systems, in order to supply a generic and a shared platform for any device within laboratories. This is called "device independancy". Moreover, we explained how authentic learning experience are produced, using "felt-life", to be as close as possible to the real thing. Furthermore, we claim that collaborative learning is part of authentic learning experience. As a consequence, we illustrate how we supply group awareness within classroom by embedding such a service within predefined widgets. This lack, however, of a sharing policy for an equal sharing of the remote device among students in the classroom. This is why we set up, still using semantic web, a layer (above authentification and authorization and before message delivery) to define and play strategies to share an online appliance. Altogether, what is achieved here, is a single platform for bringing any device online, supporting felt-life and collaborative learning experiences.

Acknowledgments. The development and researching shown in this paper is financed by the General Council of Loire, France.

References

1. A. Eckert and W. Geyer and W. Effelsberg: A Distance Learning System for Higher Education Based on Telecommunications and Multimedia - A Compound Organizational. Proceedings of ED-MEDIA'97 (1997)
2. Wei-hsiu, M., L. Yen-Jen, D. Du and M. P. McCahill, Video-based hypermedia: for education-on-demand IEEE MultiMedia 5 (1998) 72-83
3. S. Lukosch and J. Roth and C. Unger: Marrying on-campus teaching to distance teaching. Proceedings of the 19th world conference on open learning and distance education (1999) 20–24
4. H. Abdel-Wahab and K. Maly and A. Youssef and E. Stoica and M. Overstreet and K. Wild and A. Gupta: The software architecture and interprocess communications of IRI: an Internet-based interactive distance learning system. Proceedings of the 5th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET ICE'96) (1996)
5. G. D. Abowd: Classroom 2000: An experiment with the instrumentation of a living educational environment. IBM Systems Journal, Vol. 38. (1999) 508–530
6. J. R. Anderson and J.M. Fincham and S. Douglass: The role of examples and rules in the acquisition of a cognitive skill. Journal of Experimental Psychology: Learning, Memory, and Cognition, Vol. 23 (1997) 932–945
7. Latchman, H.A. and Salzmann, C. and Gillet, D.: Information Technology Enhanced Learning in Distance and Conventional. Education IEEE Transactions on Education, Vol. 42 (1999) 247–254
8. R. Holliman and E. Scanlon: Investigating cooperation and collaboration in near synchronous computer mediated conferences: Elsevier Science Ltd., Vol. 46 (2006) 322–335

9. J. Tuttas and B. Wagner: Distributed Online Laboratories. Proceedings of International Conference on Engineering Education, Oslo (2001)
10. N. Jr and R. Sharda and J. Lucca: Computer-supported collaborative learning requiring immersive presence : An introduction. Information Systems Frontier, Vol 7 (2005) 5–12
11. N. Faltin and A. Bhne and J. Tuttas and B. Wagner: Distributed Team Learning in an Internet-Assisted Laboratory. Proceedings of International Conference on Engineering Education (2002)
12. J. McCarthy and P. Wright: Putting felt-life at the centre of human-computer interaction. Proceedings of In Reective HCI Workshop (2004)
13. S. Guss: Interface metaphors and web-based learning. Lecture Notes in Computer Science, Vol 2783 (2003) 168–179
14. B.A. Alhalabi and M.K. Hamza and S. Hsu and N. Romance: Virtual labs vs remote labs: between myth and reality. Florida Higher Education Consortium (1998)
15. B.A. Alhalabi and M.K. Hamza and S. Hsu and D.M. Marcovitz: Remote labs: an innovative leap in the world of distance education. Proceedings of Information Systems, Analysis and Synthesis ISAS2000 (2000)
16. M.Roseman and S. Greenberg: Building real-time groupware with groupkit, a groupware toolkit. Transactions on Computer-Human Interaction, Vol 3 (1996) 66–106
17. J. Fayolle and J. Lottin: Eden3 : dispositif d'enseignement distance en école d'ingnieurs, retour d'expérience et positionnement stratégique. Proceedings of TICE 2002 (2002) 335–337
18. Thomas R. Gruber: Toward Principles for the Design of Ontologies Used for Knowledge Sharing. Elsevier Science Ltd., Vol. 43. (1993) 907–928
19. K. Geihs: Middleware Challenges Ahead. IEEE Computer, Vol. 34. (2001) 24–31
20. J. Roth and C. Unger: DreamTeam - A Synchronous CSCW Environment for Distance Education. Proceedings of the ED-MEDIA / ED-TELECOM (1998) 1185–1190