HAL
open science

Generic framework for remote laboratory integration

Hcene Benmohamed, Arnaud Lelevé, Patrick Prévot

To cite this version:

Hcene Benmohamed, Arnaud Lelevé, Patrick Prévot. Generic framework for remote laboratory integration. 6th IEEE International Conference on Information Technology Based Higher Education and Training (ITHET 2005), Jul 2005, Saint Domingue, Dominican Republic. pp.T2B/11–T2B/16, 10.1109/ITHET.2005.1560229 . hal-00187622

HAL Id: hal-00187622

https://hal.science/hal-00187622v1

Submitted on 1 Jun 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Generic framework for remote laboratory integration

Hcene Benmohamed [1], Arnaud Lelevé [2], Patrick Prévot [3]
ICTT Laboratory - INSA de LYON
Bât. Léonard de Vinci, 20 avenue Jean Capelle, France

Abstract - **Similar to classical *in-situ* laboratories, remote laboratories are necessary in e-learning environments, especially in scientific and technical disciplines. This paper outlines our current research on this particular way of learning. Our research objectives consist in proposing a generic framework (independent of educational content) to allow, in one hand, tutors to integrate (both remote and virtual) laboratories in their LMS (Learning Management System), and in the other hand, to enable creation, distribution and exchange of pedagogical scenarios for practical works. So, authors are able to deal their scenarios through LCMS (Learning Content Management System) and to reuse scenarios from other authors (as for any classical e-learning content). Furthermore, scenarios are no more written for a specific apparatus, but for a class of them (inverted pendulum, optical bench, …). Two main topics are developed. First one consists of modelling laboratory and scenario structures regarding IMS-LD specification to separate content from containers. Second one consists of describing system components and functionalities using ontologies (OWL standard in our case). A prototype for automation discipline is presented.**

Index Terms – Distance education, pedagogical scenarios, e-learning standards, virtual laboratories, remote laboratories.

INTRODUCTION

Last decade usage feedback analysis of e-learning environments allowed a sensible improvement in their conception, which widened their use in many training contents. Sciences and Technologies of Information and Communication (STIC) still bring new tools basically contributing to a continuous enrichment of educational practices. Yet, e-learning systems were, at first, essentially based on conceptual education activities (online courses, virtual classrooms, e-projects, role-playing, ...). It required to elaborate a research topic on electronic laboratories (e-labs) to gradually open these environments to real practical activities [1]. This need for practical experimentation on real (or possibly virtual) apparatuses answers a recognized pedagogical need, more particularly in scientific and technical disciplines.

Electronic laboratories are either Remote Laboratories (R-Labs), which offer remote access to real laboratory equipment and instruments [2]-[3], or Virtual Laboratories (V-Labs), which are based on simulations of real systems or phenomena [4]. As for any emergent concept, early R-Lab evolution started by an experimental stage. First solutions proposed simple teleoperated systems enabling to make limited experiments (level regulation in process control [5], oscilloscope experiment in electronic [6], …). As no pedagogical assumption was laid, they could be used for different audiences and different pedagogical objectives but no learning support was provided to help learners in their experiment and relieve tutor. Second E-Lab generation added a learning software interface to answer the previous lack. Unfortunately, these interfaces restrained apparatus usage to static pedagogical activities which were often programmed once for all in some programming language. They so had a (painful for users) drawback: they were only editable by computer scientists, not directly by trainers [7]. Moreover, an efficient distant learning environment assumes communication and collaboration tools between learners and trainers. Building from scratch an entire environment combining such tools with teleoporation functions is an Hercule labor.

Starting from this panorama, we launched, two years ago, a study on a generic architecture independent of pedagogical content and apparatuses [8]. The main idea is to propose an architecture for the whole chain of edition, from E-Lab scenario authoring to remote experimentation by learners, supervised by tutors. Scenarios are written for a compliant class of apparatuses and these ones can run different scenarios as long as their respective requirements match. We focused on elearning standards to enable integration of such scenarios with standard tools such as Learning Management Systems (LMS) and Learning Content Management System (LCMS). Note that LMS are also called Course Management System (CMS) or Virtual Learning Environments (VLE). Common points between R-Labs and V-Labs conducted us to propose a solution able to host both E-Lab types as in COLAB platform [9].

This paper presents this specific architecture. We start by clarifying our objectives, the constraints that we were assigned and a typical E-Lab scenario lifecycle. We then present the whole chain of edition before specifying its functioning.

[1] hcene.benmohamed@insa-lyon.fr
[2] arnaud.leleve@insa-lyon.fr
[3] patrick.prevot@insa-lyon.fr

CONTEXT OF THIS STUDY

I. Objectives

We are studying a generic architecture for E-Labs allowing edition, diffusion through LCMS containers and exchange of E-Lab scenarios between pedagogical platforms. So, an author can distribute his scenarios to other authors working with compliant similar apparatuses (identical in term of features or functionalities but possibly differently manufactured). From an other point of view, apparatuses can be used by several audiences with different pedagogical goals and scenarios.

This architecture should enable integration of E-Labs in existing e-learning environments (LMS) and allow learners to access experimentation sessions from their usual e-learning e-portal or through links scattered in online courses, virtual classrooms … The main idea is to reuse existing effective tools (registration and evaluation mechanisms, pedagogical data indexing, audio, video and textual inter-communication facilities, …) offered by classical LMS.

II. Constraints

Integration with classical LMS means conforming to current e-learning standards. It is not possible to comply to every standard so a choice is expanded in following chapter. The second constraint is to use existing free open-source tools (authoring tools, LMS, LCMS) which also comply to the chosen standard.

III. Typical lifecycle

A typical use of E-Labs follows three main stages:
1. installation of a new apparatus on a local platform,
2. authoring of a generic scenario,
3. scenario run on a specific apparatus.

III.1. Installing a new apparatus

When the local E-Lab platform administrator wants to physically install a new apparatus, he has to declare this apparatus within the ELMS platform (Electronic Laboratory Management System, defined in §4 - architecture section) by providing this apparatus class (inverted pendulum, optical bench …) among a list obtained from a public server (cf. architecture section). An apparatus class file details common points for a series of apparatuses, equipped with similar components (industrial controller, electromagnetic sluice gate, motor, …) and offering precise functionalities (motor speed parametrizing, PLC programming, video displaying, …). When no class complies to a specific apparatus, one has to create a new class from scratch or to extend an existing one by creating a class inheriting properties from an existing one (this new class just has to refer to its parent and to declare additional symbolic equipments and functionalities). This new class then becomes public and can be reused by other administrators for similar apparatus installation or by authors to create compatible scenarios (see new section). Once the adequate class is selected, the administrator declares connections between the functionalities defined in the class and real ones (for example, the "display" functionality for the "camera" component is linked to the apparatus webcam URL).

III.2. Scenario editing

Regardless of any apparatus, authors create or modify generic pedagogical scenarios. Each scenario is first given a link (an URL) to a corresponding apparatus class from the same public server as in §III.1. A scenario is written in the same way as for conceptual training: for instance, consecutive steps with exercises and a progressive evaluation. Every functionality required at different stage is either classical (tutor call, cooperation tools between learners, …) or specific to the relative apparatus class. So, only symbolic manipulation functionalities provided by apparatus class file are proposed to authors and coded in scenarios. Such scenarios are, at this stage, generic: they are just linked to an apparatus class, but they are not usable as is with a specific apparatus.

III.3. Scenario using

Tutors needing an E-Lab scenario for a given audience with precise educational objectives look for it on a classical LCMS by providing as a keyword the URL of a matching apparatus class. They fetch this scenario and integrate it into their platform: the ELMS automatically adapts this generic scenario to a provided apparatus (previously declared as in III.1). This scenario is then modified to be used with and only with this apparatus as its own functionality links direct towards a specific ELMS Web server which takes care to transmit information to the real components of the apparatus. So, this generic scenario is transformed in a specific one and is ready to be run by the local LMS within an on-line global training. On the D-day, the LMS then runs the scenario like any classical content and communicates with the ELMS through the pre-inserted URLs.

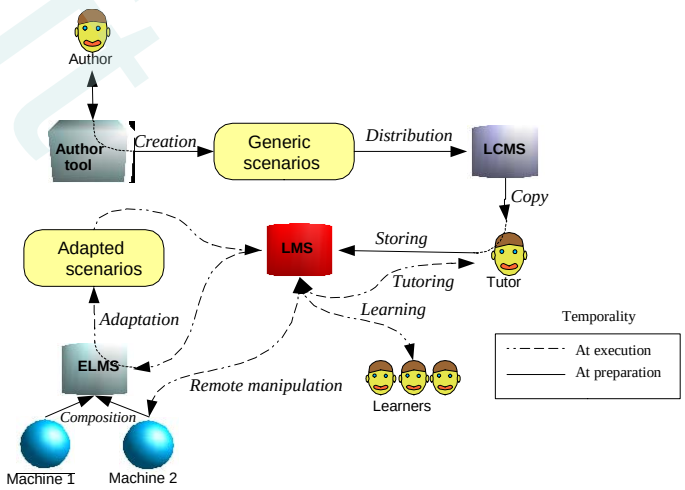

FIGURE 1
TYPICAL LIFECYCLE FOR E-LAB SCENARIOS

CHAIN OF EDITION

I. Content structuring

E-learning specifications, recommendations and standards are proposed by institutions and organizations such as IMS, ADL, AICC, ARIADNE …. Out of most recent specification, IMS-LD (Instructional Management System - Learning Design), proposed in February 2003 by IMS consortium, particularly held our attention (cf. [1]). It specifies educational contents and takes into account the whole process of training (contents, activities and roles), unlike other standards which are more specifically directed towards content structuring. We appreciated the appearance of the notion of "activity" which, in our opinion, allows coding activities of various actors during E-Labs session. Besides, IMS proposes a use case for E-Labs.

II. Generic scenarios and templates

To use pedagogical scenarios on different apparatuses (with same functionalities), it is necessary to create class of typical apparatus which answer authors' identified educational objectives. We use the term *"template"* by analogy with object programming techniques where a template is a class with generic methods to be cast to a data type according to programmers' needs.

To set up an automated process, we appealed to formal description tools, i.e., to a specific vocabulary, embodied by an ontology [10]. We were inspired by semantic Web techniques and we adopted a recent standard: OWL (Ontology Web Language) normalized by the W3C. This approach is the same as [11] who propose using semantic Web to describe the virtual laboratory contents with meta-data LOM RDF / XML (defined by the LTSA working group).

We therefore built an ontology of the standard components found in E-Labs and their typical functionalities. Figure 2 presents some components of this ontology; arrows represent inheritance links, in the meaning of classes in object programming.

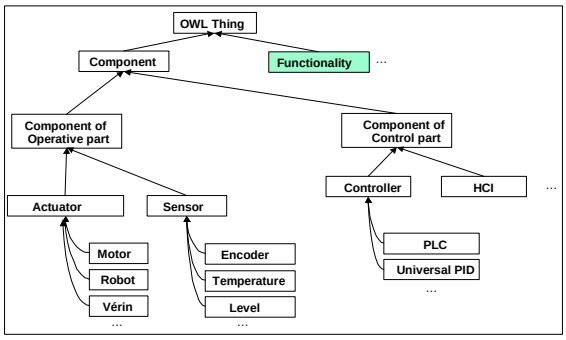

FIGURE 2
EXTRACT OF THE ONTOLOGY MOTHER: COMPONENTS

Figure 3 illustrates associations between components and functionalities; arrows in dotted line correspond to an association *"component X supplies functionality Y"*. This *"parent ontology"* must be universal to serve as a reference dictionary for our whole architecture. Without it, templates could be defined with heterogeneous vocabulary, which would make difficult a choice out of them and would enable false duplicates. At present, our parent ontology is far from being exhaustive. It must be extended by community, according to its needs, with the help of online editing tools.

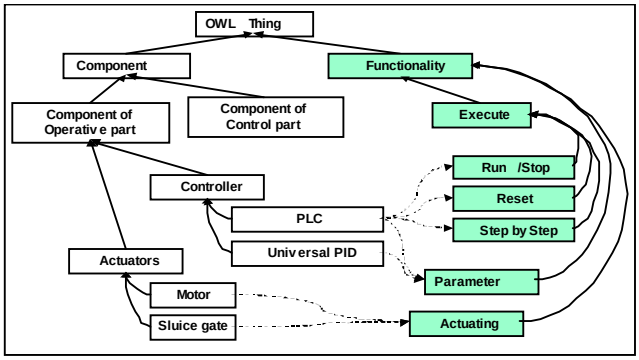

FIGURE 3
EXTRACT OF THE PARENT ONTOLOGY:
COMPONENTS-FUNCTIONALITIES ASSOCIATION

Templates provide equipments and related functionalities for a class of apparatuses. But they currently do not provide information concerning the organization of these components nor the aim of the represented apparatus. For example, an inverted pendulum could be associated with two motors, two incremental coders and a digital controller, as could be an electric vehicle. Nevertheless the same ontology represents two globally different apparatuses, not just by their organization. So, a main improvement of template structure will consist in enabling a more precise and generic modeling.

Figure 4 presents an example of apparatus that can be associated with E-Labs dealing with temperature control. In this case, the apparatus is an industrial oven. Heating power percentage can be manually adjusted and set in an open or closed loop. Learner can create temperature disturbances by opening the oven door.

III. Functionalities coding in scenario steps

IMS-LD standard proposes to code the environment items of an activity as *"learning objects"*. *Environments* are logical groups of learning objects and are referenced in *activities*, which, in their turn, are associated with *roles*. *Learning objects* can be documents or links towards tools and consist of a resource associated with optional arguments and metadata. A resource is simply a file provided in the IMS-LD package containing each scenario or a URL. Anyway, it is unique in an entire package and can be referenced from several environments. It is then easy for authors to propose different functionalities according to roles and their current activity.

It seemed, for us, natural to use these *learning objects* to provide E-Lab functionalities put at learner's disposal for interaction with apparatus (one functionality per object).

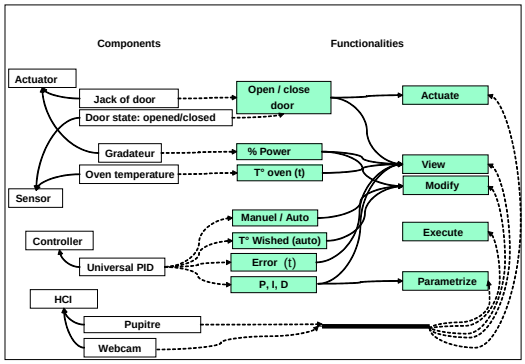

FIGURE 4
EXAMPLE OF A TEMPLATE:
TEMPERATURE REGULATION OF AN INDUSTRIAL OVEN.

As every pedagogical object in a scenario has to refer to the same apparatus template, we opted to declare the URL of the referent template as unique resource for each of them. A pedagogical object differs then from the other one by the parameters associated to its resource call. These parameters, supplied in the form of a string, then have to contain the name of the expected component, a method (accompanied with its own parameters). For example: WebCam display and oven switch off use common URL[4] (pointing towards oven template) given in IMS-LD resources and respective parameters *"webcam.display()"* and "*dimmer.Power.Set(0.0)*".

VI. Adaptation process: from generic to specific

As generic scenarios feature manipulation resources which point towards their respective template, it is thus necessary to modify these URLs before executing a scenario via a LMS so that they can be used with a particular apparatus. This is a task dedicated to the ELMS which already knows the links between the physical apparatus and its template. It just has to replace the scenario template URL by the URL of a Web server able to supply expected services corresponding to the requests performed by the LMS for learning objects (start an applet for the control of a motor, redirect learner web brower towards a webcam video feed, send parameters to a PID controller, …).

We opted for declaring the links between apparatus and template under the form of a new ontology, appropriate for each apparatus of tutor specific platform, and containing only the instances of the classes supplied by the template.

At runtime, when the ELMS web server receives requests for a specific apparatus, it is able, when several apparatuses are declared to use a same template, to automatically redirect learners to a free apparatus. The learner-apparatus association can be programmed to be valid during a whole session or to be reexamined at every call. Second option permits to apply a scheduling algorithm such as described in [12] in order to optimize apparatus usage.

ARCHITECTURE

According to needs, constraints and solutions presented up to here, we conceived an architecture including five fundamental parts, as shown in the figure 5. Two are generic (also used for other educational contents): an authoring tool for the scenario and an IMS-LD compliant LMS. Three others are specific to E-Labs: a specific authoring tool to link a scenario to a template, an ontology server (which is unique, compared to other parts which repeat according to authoring environments and E-Lab platforms), and an ELMS.

I. The LMS

At present, as far as we know, the only available open source IMS-LD compliant environment is CopperCore. It is implemented with Java language and conceived to be used and integrated into other environments. Because Coppercore is just an IMS-LD runtime environment and not an LMS platform, we melt it down into *"Moodle"* open source LMS, providing necessary features (actor intercommunication tools, course management and indexing, ...). However, direct use of a fully IMS-LD compliant LMS could be envisaged.

II .Management of ontologies

OntoServ is a server for different ontologies to be used by the ELMS and the authoring tools related to E-Labs. In one hand, it features a web server publishing template OWL files and, on the other hand, it features an analyzer tool. It provides reference descriptions of components and functionalities for a given class of apparatus in a central way. In a near future, it will be used by E-Lab authoring tool to provide authors such information. It is yet used by the ELMS to associate generic functionalities to specific apparatus components. Based on the *"Jena OWL API"*, this server is able to answer a certain number of requests:

- What are functionalities offered by an apparatus class?
- What are functionalities of a component in an apparatus class?
- …

Ontologies managed by OntoServ are at present manually edited by a person with the help of *"Protegé"* ontology editor. This last one also internally uses Jena OWL API. Later, we plan to provide tools to edit templates on-line.

[4] http://www.ictt.insa-lyon.fr/Elaboratories/Generic/Templates/Oven/Oven.owl

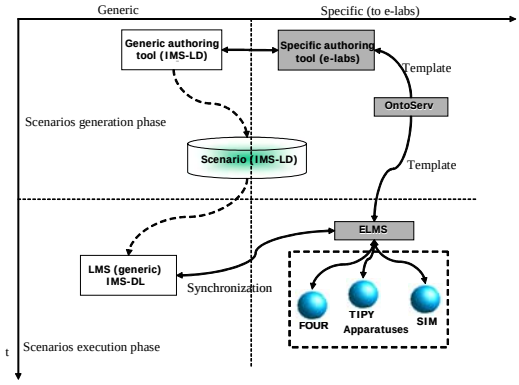

FIGURE 5
GLOBAL ARCHITECTURE PROPOSED

III. Authoring Tool

We use the open source editor "Reload" as generic authoring tool; IMS-LD compliant (it only supports the *"level A"* right now). It is also possible to use XML edition tools such as *"XMLSPY"*, but as they do not intrinsically integrate IMS-LD model, they remain very little ergonomic and entail risks of miswriting.

It is necessary to have a specific authoring tool for E-Labs for the integration of the functionalities in the pedagogical scenarios by means of the collective template server. The resort to two distinct softwares is possible but makes the edition not much ergonomic whereas the use of a two-in-one global software authorizes a simultaneous edition of the scenario and the associated functionalities. Source code of editors such as Reload being open, we can envisage to extend such an open source editor: we then take advantage of professional developments but we must take into account that developing and maintaining such an extension is a task which requires a consequent work.

VI .The ELMS

The ELMS makes the link between the LMS and real apparatuses. Its tasks are installation of apparatuses (with functionality registration), compatibility test of specific scenarios with local apparatuses, scenario conversion (when possible) and, at runtime, apparatus use management for learners and tutors (access rights, choice of a free apparatus and effectively activated or inhibited functionalities according to scenarios).

EXAMPLE: AUTOMATION DISCIPLINE

I. Apparatus description

Next figure illustrates this example apparatus. It is an electric oven featuring a dimmer to modulate heat power and an industrial digital controller. This controller is connected to our ELMS in order to parametrize it (open/close loop, P, I, D) and to get back real time variables (power, temperature, error). The educational interest is, notably, to familiarize students with the control of heating systems.

FIGURE 6
AN INDUSTRIEL OVEN

II. Scenarios and templates

This apparatus is associated with an oven template URL. Some of its functionalities are depicted in figure 4. A scenario can be used for any similar oven and consists of three activities (discovery, modeling and regulation) dedicated to learners and an activity of supervision for the tutor. For the moment, we are limited to A level of the IMS-LD, but, in a near future, we will extend it to B and C levels to be able to create enhanced realistic and dynamic scenarios.

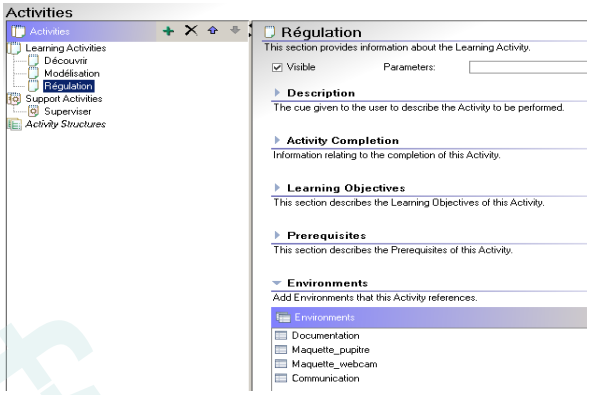

FIGURE 7
RELOAD AS IMS-LD AUTHORING TOOL

III. Runtime environment

Previous generic scenario is firstly automatically adapted to our specific oven by a tutor using the ELMS features. As our local apparatus provides every functionality used in this scenario, this one is fully usable. Then obtained specific scenario is run by the LMS as any classical IMS-LD content. Discovery activity is associated with an environment which is a collection of learning objects (documents, camera, Control User Interface …). When learners or tutor click on webcam resource link, their browser requests ELMS server for video display functionality (argument: *apparatus= oven&component=Webcam&fonctionality=display*) which, in turn, redirects the call to the oven webcam.

CONCLUSION

E-learning environments do not, right now, integrate features allowing to run E-Lab sessions for learners and tutors with the same opportunities of edition, use and reuse of pedagogical content as for other classic contents (online courses, virtual classrooms, …). We thus propose an architecture answering this need and reusing features, tools and modern standards of e-learning platforms. So, by founding the whole architecture upon an ontology arborescence specifying apparatus classes with common components and classical functionalities, scenario authors can henceforth edit their generic scenarios in a IMS-LD format and connect them to a template representing a real or virtual apparatus class. They make them so compatible with any apparatus associated to the same template, authorizing the re-use of their work. It also permits to run different scenarios on a same apparatus series.

These works open rich perspectives areas of research, among which: optimization of apparatus usage by simultaneous groups of learners, improvement of adequacy tests between scenarios fetched from a LCMS and a specific equipment from an E-Lab platform, scenario functionality indexation on LCMS, … These works are being experimented on our local experimental platform before being tested at a wider scale in our open university (INSA-V).

REFERENCES

[1] Benmohamed H., Lelevé A. et Prévot P., "Remote laboratory: new technology and standard based architecture ", *1st International Conference on Information & Communication Technologies: from Theory to Applications (ICTTA'04)*, Damascus, Syria, April 19-23 2004.

[2] Berntzen R., Strandman J.O., Fjeldly T.A. et Shur M. S., "Advanced solutions for performing real experiments over the Internet ", *International Conference on Engineering Education*, Oslo, Norway, August 6-10 2001.

[3] Azorin J. M., Paya L., Jimenez L. M., Garcia N.M., Sabater J.M, "Remote laboratory for automation education ", *2nd IFAC Workshop on Internet Based Control Education (IBCE'04)*, Grenoble, France, September 5-7 2004.

[4] Zsolt I. V., "Virtual Laboratory Support for Electronics Packaging Education ", *International Conference on Engineering Education*, Oslo, Norway, August 6-10 2001.

[5] Saad M., Saliah-Hassane H., Hassan H., El-Guetioui Z., Cheriet M., "A Synchronous Remote Accessing Control Laboratory on the Internet ", *International Conference on Engineering Education*, Oslo, Norway, August 6-10 2001.

[6] Chen S. H., Chen R., Ramakrishnan V., Hu S. Y., Zhuang Y., Ko C.C., Chen B. M. , "Development of Remote Laboratory Experimentation through Internet", proceddings of *Symposium on robotics and control*, Hong Kong, Julay 1999, pp. 756-760.

[7] Nippert C. R., "Using Web Supplemental Instruction for Chemical Engineerig Laboratories", *American Society for Engineering Education Annual Conferene and Exposition*, 2001.

[8] Lelevé A , Meyer C. et Prévot P., "Télé-TP: premiers pas vers une modélisation", *proceddings of Symposium on Technology of Information and Communication in education for engineering and industry*, Lyon, France, November 13-15 2002, p. 203-211.

[9] Jong T. D. : "Learning complex domains and complex tasks, the promise of simulation based training", proceddings of *conference on Computer Aided Learning in Engineering Education (CALIE'04)*, Grenoble, France, February 16-18, 2004, p. 17-23.

[10] Guarino N., "Formal Ontology and Information Systems". *1st International Conference of Formal Ontology in Information Systems*., Trento, Italy , June 6-8, 1998. IOS Press.

[11] Tamáš J., Šulc B., « New possibilities of e-learning systems based on the semantic web principles », 2*nd IFAC Workshop on Internet Based Control Education (IBCE'04)*, Grenoble, France, September 5-7, 2004.

[12] Lelevé A., Benmohamed H., Prévot P., « Sharing a System between Simultaneous Learners in Remote Laboratories », 2nd *IFAC Workshop on Internet Based Control Education (IBCE'04)*, Grenoble, France, September 5-7, 2004.

Draft