Towards semi-automatic generation of training scenarios in industrial automated systems

Saher Arnous, Arnaud Lelevé, Khalid Kouiss, Patrick Prévot

Towards semi automatic generation of training scenarios in Industrial Automated Systems

Saher ARNOUS

LIESP, INSA Lyon,

Université de Lyon
F69621 Villeurbanne
Cedex-France

+33 (4) 72 43 60 47

saher073@gmail.com

Arnaud LELEVE

LIESP, INSA Lyon,

Université de Lyon
69621 Villeurbanne Cedex-
France

+33 (4) 72 43 60 47

arnaud.leleve@insa-lyon.fr

Khalid KOUISS

LIMOS, IFMA, Campus
Scientifique des Cézeaux
BP 265, F63170 Aubière

France

+33 (4) 73 28 81 07

Khalid.Kouiss@ifma.fr

Patrick PREVOT

LIESP, INSA Lyon,

Université de Lyon
F69621 Villeurbanne
Cedex-France

+33 (4) 72 43 82 94

patrick.prevot@insa-lyon.fr

ABSTRACT

In the context of E-Learning, remote hands-on training has become an insisting need as in traditional learning, especially in scientific and technical disciplines. Electronic Laboratories (ELabs) have been growing for the last few years. But till now, exchanging learning scenarios of ELabs is still difficult as existing scenarios (elaborated by means of standard ELearning authoring tools) are restricted to specific apparatuses, which prevents instructors from reusing or exchanging scenarios. Actually, LIESP team started in 2002 a research aiming to provide a framework which helps towards exchanging ELab learning scenarios when they fit to similar apparatuses (same functions, maybe not the same hardware). Meanwhile, LIMOS team focused on a design process to automate PLC code generation to help to design and generate programs for industrial discrete systems.

This paper presents a project of merging these works to help ELab designers to design and integrate apparatuses into ELab frameworks when these apparatuses are discrete systems.

General Terms

Design, Experimentation, Human Factors, Standardization.

Keywords

E-Learning, Hands-on Training, Remote Laboratories, Virtual Laboratories, Online Laboratories, Remote Interactive Laboratories, Distant Laboratories, Learning Management System

1. INTRODUCTION

In a scientific education system, hands-on training goes side by side with theoretical learning to reduce the gap between what students learn in the classroom and what they will face in the real world. As well, virtual universities offer scientific education through the internet to a large number of students around the world. Such educative systems must offer practical training for their students to achieve a reliable level of education especially in scientific and technical disciplines [10] [22]. Actually, Electronic Laboratories (ELabs) became, as a part of a virtual university, an essential need to provide students with practical training.

In fact, classical educational systems present significant drawbacks concerning the hands-on training. Some researches, e.g. [15] [19], cited that the increasing number of students in scientific and technical colleges limits the capability to provide students with adequate practical knowledge. In addition, the obligation of presence in a certain place at a certain time may not be convenient for all the students. Moreover, up to date hardware equipments as subject of training are not always affordable in all the universities, not to forget the devoted time and technical staffs [16] besides maintenance expenses [19]. Finally, but not the last, professional laboratory technical staffs are not always available [24]. Accordingly, distant learning can be understood either within a virtual university or even within a real one.

Thanks to the Information and Communication Technologies (ICTs), distant learning has progressively improved; we can find in almost every university a Learning Management System (LMS) to provide electronic resources for local and distant learning activities. ICTs are also used to provide learners with alternative solutions for hands-on training by means of simulation tools which are widely spread through specialized software (e.g. MatLab and Labview). But the real challenge is to enable users to access real hardware through a web browser for the purpose of hands-on training, since the designers of Remote Laboratories (RLabs) require good knowledge in industrial computing and automated systems besides their skills in didactics in ELearning systems. Not to forget the difficulties faced by managers of RLabs to provide the instructors with flexible methods to propose pedagogical activities on remote materials (an instructor of civil engineering may not have skills in ICTs to exploit a RLab platform). Nowadays, current ELabs are thought to be used beside LMS with learning scenarios [17], [23], [3]. The integration of RLabs in LMS has been slowed down according to the fact that first LMS were usual closed proprietary software systems that are often not customizable [24].

LIESP team began in 2002, according to this realization, a project focusing on a framework involved in an edition chain for ELab learning scenarios. We aimed at proposing effective common tools for the authors and instructors of ELab learning

scenario whatever discipline or apparatus, namely enabling the creation of generic scenario, independent from the materials and adapting it on different platforms. We took into account existing E-Learning frameworks: authoring tools, Learning Content Management System (LCMS) and LMS.

Meanwhile, LIMOS has been working since 2004 on a Component-Based Approach for the design of discrete control to drive conveying systems. A methodology allowing to automatically generate the control programs has been proposed to provide an easy way to obtain source code compatible with the IEC 61131-3 standard. The methodology is based on a MDE (Model Driven Architecture) approach in which models are described using meta-models at each step of the process.

This paper introduces a collaboration between LIMOS and LIESP to help ELab designers to design their apparatuses. This project is founded on the notice that in a distant learning context, many operations have to be automated to permit the teleoperation of appliances which were manually handled. Therefore such automated systems have to be designed with two purposes: enabling their teleoperation and offering a network common interface to enable LMS cooperation in an Elearning context. Therefore, this project aims at providing a semi-automated design process to help designers in this work.

The rest of this paper is organized as follows: in section (2) we make a general observation of the different forms of E-Laboratories. In section 3 we overview our model of "generic" ELabs while in section 4 we overview the approach of system control design, and we introduce our collaboration project in 5. Last section closes this paper by a general conclusion.

2. E-LABORATORIES OVERVIEW

Electronic laboratories (ELabs) are related to distant learning as well as to classical learning, especially in scientific and technical disciplines. Through the literature one can find several kinds of ELabs dedicated to answer specific needs in specific disciplines; for example: electronics, circuits and electricity [2] [4] [16], mechanics [14] [1].

The common point within the two forms of learning is that ELabs are manipulated through computers, so we distinguish between local and distant ELabs. The literature also carries out some studies about the aspects of ELabs in addition to several comparisons between the different contexts and types of ELabs, [2] [9] [21] [1] for instance.

2.1. Local E- Laboratories
Computer was introduced in traditional hands-on laboratories as an attempt to overcome some of the drawbacks mentioned in section 1, by offering software to help students to perform actions on a real system (automate some actions, get, plot and record massive data, etc.) [22][10][27][26]. Simulation tools are offered for training or if there is no real system, also simulators

replace some parts of a real system when those parts are hard to be manipulated by learners.

2.2. Remote Laboratories (Rlabs)
They are real laboratories whose users control and use it remotely. It should contain the same equipments as in local ELabs for experimentations in addition to new ones dedicated to enable their teleoperation through a LAN, a WAN [16] or the internet [25].

2.3. Virtual Laboratories (Vlabs)
They rely on system simulation, [13][18], and are widely adopted where no physical resources are available, e.g. sensor networks [8]. But it does not lead to real useful experience as they don't allow acting/reacting with physical components.

2.4. Hybrid Laboratories (HLabs)
They are RLabs remotely handling real systems beside simulations. Simulators emulate the performance of a real part of a device, [12] [1], when the simulated part does not exist or when this part is hard to be connected to the system to read information about its performance (e.g. measuring the temperature of the magnetic core of an electric engine). Also simulation can be used to give a graphical view of spatial operations.

Through the literature we noticed that the researchers concerning the design of ELabs are concentrated on two major points:

1) The design for the purpose of an efficient and reliable connectivity with the outer side of an RLab via a network or the internet. The concept of Portals is introduced in [11] [20], and proposed for a future improvement in [13].

2)The modelling of the physical structure of laboratory constituents leads to a better realization of the material part of the system, which simplifies the design of software to control and exploit the physical resources of the laboratory. This point is interestingly discussed for automated systems design and different ideas about the modelling of such systems were proposed. The applicability of each proposed model on an automated system was proved by its author, which lead us to believe that the cases (problems) in this domain don't have one ideal solution but can be adapted and handled form different points of view, (e.g. two different models and methodologies of almost two similar convoying systems were presented in [6] [18]).

3. LIESP GENERIC ELAB MODEL OVERVIEW
In this section we illustrate the general aspects of our previous works about ELabs. Hacen Benmohamed has proposed in his PHD work [5] an ELab architecture enabling its use through learning scenarios written within Elearning standards (IMS-LD "Instruction Management System-Learning Design"). He designed a lifecycle opening the way for reusing learning scenarios for similar ELabs systems (same functions but possibly not exactly the same hardware). This work defines how to reuse standard Elearning tools to create (by authoring tool), diffuse

(LMS "Learning Management System") and exchange (LCMS "Learning Content Management System") ELab learning scenarios between laboratory platforms.

Hence, scenario authors can create, manage and distribute their scenarios over a LCMS as classical training contents. As well, authors and instructors can search scenarios compliant with their systems and download them from ELab scenario libraries (LCMS). They then may enhance them to satisfy new needs and then redistribute new versions of these scenarios.

3.1. Scenario and Genericity Towards Apparatuses

The IMS Global Learning Consortium creates standards for the development and adoption of technologies that enable high-quality, accessible, and affordable learning experiences.

Figure 1. IMS-LD conceptual model

The IMS-LD specification supports the use of a wide range of pedagogies in online learning by providing a generic and flexible language, which is designed to enable many different pedagogies to be expressed.

IMS-LD defines the structure of a learning scenario: a scenario written by an author (generally a teacher), delivered by an instructor towards learners. The standard defines resources (documents and web), activities for each user and how they may interact with each others during the corresponding training session. LD has been designed to be flexible so as it does not embed any didactic approach: scenario author is responsible for this didactic approach.

A learning package is composed of specifications about pre-requisites and the global learning objectives, in addition to temporal organization of the activities of different roles (Fig.1). Defining simultaneous plays is possible, each play consists of sequentially running actions. Therefore, the training scenario can be applied on different learning groups of different chronologies.

In turn, each action is composed of activities, each one is associated to one or more roles (learner, tutor… according to design needs). Every activity has a description, learning

objectives, pre-requirements, ending conditions and one or more environments. Each environment is a collection of learning objects (like documentation) and necessary services for learners and instructors. The environments are responsible of distributing the pedagogical resources according to appropriate needs for each activity and each role.

According to IMS-LD standard, tools must be declared in the environments of a scenario as references (URL). When such scenarios are exchanged and used in different institutes, such links have to be manually replaced by local equivalent ones. In the context of ELabs, authors have to declare manipulation tools (teleoperation panel, video, plots, …) to permit users to use the corresponding apparatus. The same problem arises when the scenario is linked to another equivalent apparatus: every URL has to be manually updated (local apparatus address, its HMI address, portal address… this is impractical). To make sure that the scenario is generic, these URLs should describe the referred tool and its expected functions rather than a static address of a given manipulation server to avoid restricting the scenario to a specific one only apparatus in the world.

Thus, to use a generic scenario on equivalent apparatuses it is important to create type models of apparatuses that can satisfy clearly identified pedagogic needs. We call a type model "Template" in analogy with the techniques of object programming, where a template is a class defining generic functions (common ones in every apparatus of a given family) that can be adapted according to programmers needs.

3.2. Formal Description Ontology

Templates management should be semi automatic to help scenario and apparatus designers. Therefore, both of software and humans should be able to manage the semantics of a template and other related representations; so we used formal description tools leading to a specific vocabulary represented under a form of ontology. Web semantics techniques are inspired and the standard OWL, normalized by W3C, is adopted.

Core ontology, called "Root Ontology", is constructed to describe the classical components of a pedagogic apparatuses associated to the functions they provide. This distributed ontology, dealt by a web server, has to be unique to provide reference to the ensemble of the architecture, but it is not exhaustive and has to be associated to extension tools. (Fig.2) shows some components of the root ontology where the arrows represent inheritance links, while dashed arrows represent the association between components and functions of form: "*component X provides function Y*".
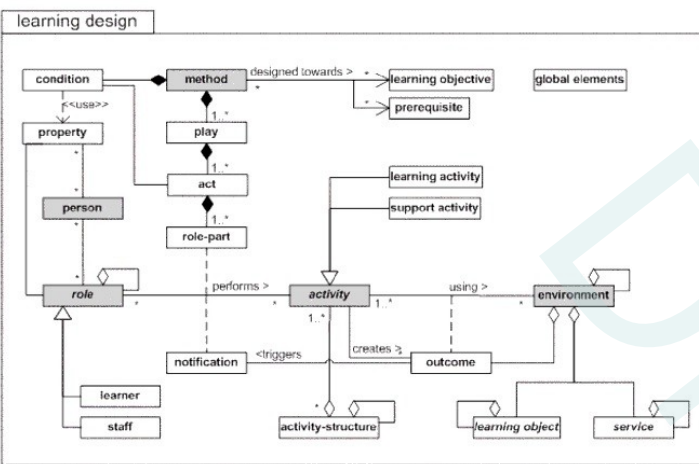
Root ontology can be considered as an abstract description of apparatus parts (classes: Component, Function, Operative Part Component, Actuator, Sensor, ...). This root ontology can be extended by building new Components and Functions following the same inheritance rules as in object programming to provide concrete part definitions (Temperature Sensor, Jack, PID Controller...). Next, child ontologies group some of these components are related to functions to describe a family of apparatuses (inverted pendulum, spectrometer, …).
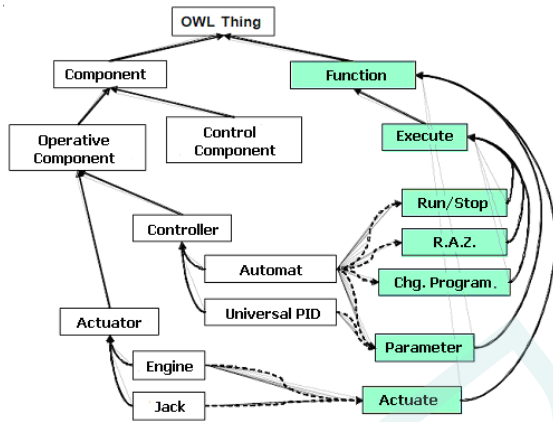

Figure 2. Extract from Root Ontology

These ontologies, "family templates", help to classify each apparatus in ELabs in order to associate to them corresponding learning scenarios. At the end of the chain, there are two other kinds of templates, related to a unique family template:

• apparatus templates which declare effective components and functions present in a given apparatus in a real ELab;

• scenario templates defining required functions of a scenario.

3.3. Ontology Illustration

When we want to add a new family of apparatus in ELabs, a new template for this family must be written. As shown in (fig.2), we defined links between components and functions in a bidirectional way in the following sense: "component-provides-function" and "function-provided-by-component". This definition helps designers to associate functions of their choice to certain parts (components) of the apparatus. These root classes are defined in the root ontology (a single file which is provided online on a public web server), so that it is shared by any child ontology (templates).

After having defined the root ontology file, template ontologies can be created by inheriting from the root ontology any necessary component and function.

3.4. Installing an Apparatus

To install a new apparatus in an ELab, the administrator of the platform declares it in the Electronic Laboratory Management System (ELaMS) managing the platform of the laboratory. He has to assign to this apparatus the template describing the functions really provided by it and the way to access them. At this stage, each class member is tagged with an OWL annotation

which gives the URL to call to run this function. The ontology can also declare components corresponding to parts of the apparatus and defines which component provides which function.

(Fig.3) illustrate the process of installing an industrial oven through three levels of ontologies: starting from the root ontology by selecting a graduator component, then assigning a template of generic oven with the root ontology and associating the functions it provides by the chosen component(s) and finally instantiating the corresponding classes to the parts of the oven and their provided functions.
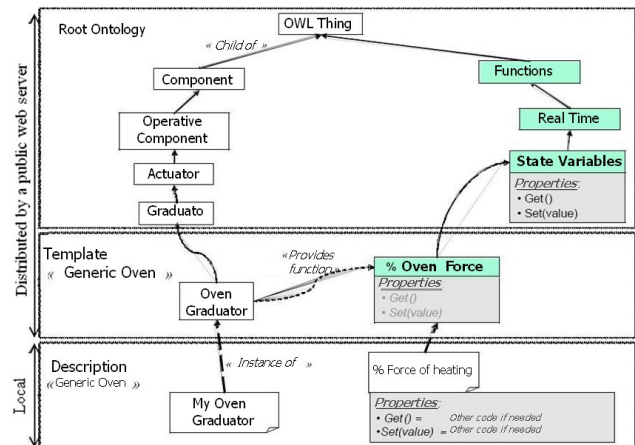

Figure 3. Links between the 3 levels of ontologies for a Graduator class and its state variables

3.5. Scenario Edition and Using Phases

Writing a generic scenario of an apparatus implies its association to an ontology describing the used functions in it; for every used function in the scenario an instance from the apparatus ontology is added. Actually, the author has to make links between the scenario and the components and the functions in a manual way, and then the scenario is stored on a public LCMS server.

When an instructor chooses a scenario of his preferences from LCMS, it may be associated to a same template as the apparatus or to similar template. Then he integrates it in the platform by addressing to ELaMS a compatibility verification process which analyzes if the required functions in the scenario correspond to the instances of functions from the apparatus template. If every required function is instantiated in the apparatus ontology, the scenario is compliant with the apparatus, and then the ELaMS automatically adapts the scenario to the designated apparatus thanks to the links template↔apparatus. The resources and the parameters of pedagogical objects of the scenario are transformed into URLs referring to the web interface of the ELaMS server, which intermediates with the apparatus corresponding to the requirements of the scenario. Like that, the scenario passes from generic phase to specific phase and becomes ready to be executed by LMS like any other classical pedagogical contents

3.6. Apparatus Virtualization Interface

Most apparatuses within an ELab are equipped with electronic controller enabling their connectivity with computerized systems. It is generally embedded into software installed on a dedicated hardware. To overcome hardware setup, a virtual interface should be built consisting of the apparatus ontology which maps

generic functions furnished by a given apparatus to URLs provided by these dedicated interfaces. Likewise, designers of an Elab can avoid designing apart the two closely linked parts: this web based interface and the apparatus controller. The global objective is to propose to external tools a virtual networked interface defined by the corresponding apparatus ontology, and hiding specific hardware behind a "generic" interface.

4. COMPONENT-BASED APPROACH FOR SYSTEM CONTROL DESIGN

LIMOS team began in 2004 a project, through which Fabien CHIRON proposed in his PHD work [7] a component-based approach for the design of discrete control to drive conveying systems. A methodology allowing to automatically generate the control programs has been proposed to provide an easy way to obtain source code compatible with the IEC 61131-3 standard. The methodology is based on a MDE (Model Driven Architecture) approach in which models are described using meta-models at each step of the process. Founded on the concept of Component Based Software Engineering, the approach aims likewise at reducing the required time to design the control part, in the context of conveying systems, and to facilitate the creation of controls in the context of reconfiguration, since it is necessary to provide several versions of control.
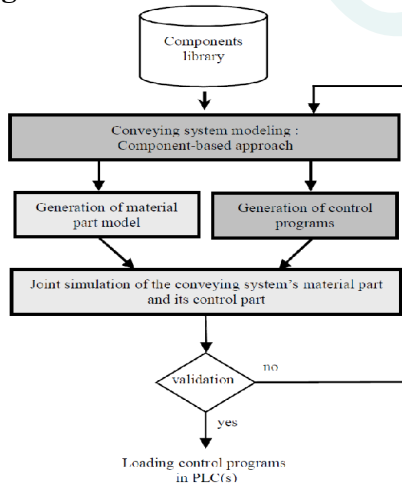
4.1. Design Process

Figure 4. Global design process

The global process is part of a usual flow based on a simulation to validate or modify the design parameters. The objective of this process is to design, validate and implement control of conveying systems. (Fig.4) describes the design process which invokes: system modeling, generation of material part model, generation of control programs and simulation.

The system model is built upon a component library. After validation, control programs can be loaded in PLC(s) (Programmable Logic Controller). If simulation results do not correspond to the specifications, the system model is modified.

4.2. Component-Based Approach

This approach provides a clear and easy way to reuse previously modelled elements or to modify the system's internal structure.

Components refer to operations which are performed by a resource of the system. The resources in turn (ex: sensor, hydraulic jack) can perform several operations which implement the resource functionalities. Based on the typology applied to generic and contextual functions, three different types of operations are applied: 1) Basic operations like detecting by a sensor, 2) Contextual operations like detection the position of a jack by a sensor where the sensor is associated with the jack, and 3) Effective contextual operations like transferring an object from one area to another by a jack on a conveyor (Illustrated in [18]). The typology of operations is represented by a class diagram (Fig.5).
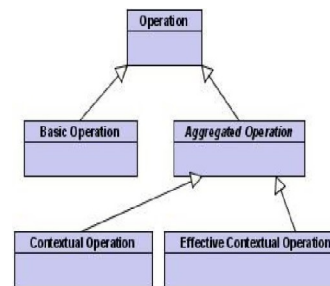

Figure 5. Typology of operations

A component is a set of operations including monitoring, supervision and control points of view. Besides functions, it takes into account the system structure and its physical organization. Components types are defined by analogy to operations types [18]: 1) Basic component which is a set of basic operations e.g. jack or sensor. 2) Basic enriched component which is a set of operations enriched with contextual ones, like a jack to which a sensor is attached. 3) Support component whose only function is to support and specify some spatial constraints, e.g. a belt conveyor. 4) Effective contextual component is a set of effective contextual operations put together, according with the part flow, e.g. a jack component and a motor component associated with a conveyor component enable to define an ejector component. 5) System component which models the whole system and refers to at least one effective contextual component.

The component description uses a black-box formalism. Inputs and Outputs relating to physical flow are separated from Inputs and Outputs dedicated to control. Basic and support components include parameters providing adaptability to different designs. They are stored in a library as validated ready-to-use models. An aggregation procedure has been developed. It consists in building a component of level L from several components of level L-1 brought together. Contextual components represent the first level of aggregation while the system component is the last level of aggregation (the whole system).

As explained in [18], a component is composed of four views: operating part, graphical, constraints and control views. Control part is described using sequential function charts (SFC). SFC has the advantage of manipulating simple concepts which are comely used by PLC program developers. When components of level L are selected to be aggregated, the co-ordination of the different control parts, named hierarchical control part, has to be generated for the level $L+1$ component.

The control structure is hierarchical, and two kinds of control part are considered: low level control part and hierarchical control part. Basic and support components which are stored in a library include low level control part. A hierarchical control part refers to an aggregated component.
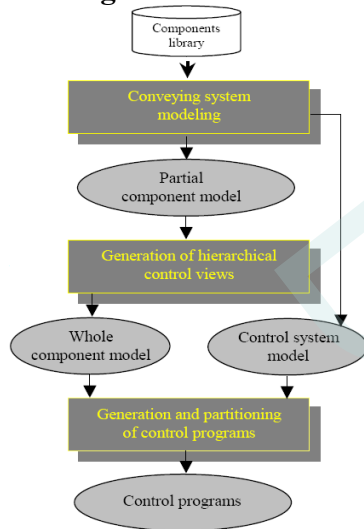
4.3. Control Design

Figure 6. Control design methodology

The control design methodology (Fig.6) involves three steps. The first step is dedicated to create the partial component model and the control system model by using a components library. A partial component model is seen as an assembling of components. It is partial because it does not contain control views of aggregated components. An algorithm (details in [18]) is proposed to automatically generate the control views of aggregated components. In the second step, the partial component model is referenced to obtain a whole component model. In turn, component model is refined to obtain the whole component model. In the third step, both the control system model and the whole component model are used to generate control programs. The control programs generated are IEC 61131-3 compliant, implemented by PLCs without any transcription and are expressed using XML. In this step the control system model captures all aspects of a control system in terms of implementation (hardware components) and the whole component model is used for the description of control functionalities.

The MDA methodology of OMG, is used in this approach. The system functionalities are defined by the Platform Independent Model (PIM), to which the component model corresponds. The projection of functionalities on the hardware architecture defines the Platform Specific Model (PSM). Thus in this approach, the PSM corresponds to control programs which can be implemented on PLCs. The hardware architecture which is mainly composed of PLCs, is described in the control system model.

5. SYNTHESIWING OF FUTURE DEVELOPMENT POINTS

Previously presented works are a basis for a collaboration between LIMOS and LIESP to help ELab designers to design their apparatuses. This project is founded on the notice that in a distant learning context, many operations have to be automated to permit the teleoperation of appliances which were manually handled. Therefore such automated systems have to be designed with two purposes: enabling their teleoperation and offering a network common interface to enable LMS cooperation in an Elearning context. However, this project aims at providing a semi-automated design process to help designers in this work, based on previous works of LIESP and LIMOS teams.

Currently, ELab designers have to design:

• the operative part (sensors, actuators, etc.) to enable automatic handling of an initially local apparatus;

• the control part embodied by a PLC or a PC;

• the network interface to enable teleoperation and LMS synchronization. This interface is based on the family template definitions which describe which apparatus function should be available remotely.

Control part and network interface design are really dependant: through the network interface, users will be able to run control part functions. So the design of both ones should be linked to prevent a double design and to prevent errors. When starting from scratch, the requirements for an ELab are defined from learning needs. These learning needs can then (manually) be converted to ELab components and required functions which can be embodied by a part of or a whole family template. The main idea is to use the LIMOS edition chain to automate the source code generation from the corresponding family template definitions for:

• the control part (PLC program skeleton) : desired functions and how to call them from a URL;

• the interface : filling the apparatus template with correct URL

• (in a second step) the web HMI: reusing widgets from a library, with corresponding parameters automatically set.

We remarked in the current stage of the project that there are several points to be improved. The root ontology we presented is in its beginnings and requires regular editions. The fact that it is

stored in a single file and available on-line means risking modifying some definitions already used by child classes. At this time we may propose to omit modifications and replace it with the possibility of overloading undesired descriptions (which are desired to be modified) by new ones, with conformance to overloading concept used in Object Oriented Programming (OOP).

At verification process of the compatibility (template ↔ apparatus), when ELaMS notifies the instructor of the lack of some function, he may manually define links between the scenario and the apparatus. This means that the formal description in the ontology was insufficient to represent all the available functions of an apparatus. In this case we can propose a method to automatically generate a sub ontology from the ontology associated to the apparatus, by inheriting its functions and adding to it the manually defined functions by the instructor. This way, when another scenario linked to the same ontology passes though the verification process, it can have the possibility of being automatically adapted. In parallel, simulation software can be employed to provide unavailable functions of an apparatus so that the generic scenario can find corresponding functions to its activities.

Inspired from the automated production systems, another idea is to propose an automatically executable generic scenario within ELaMS platform as a utility, to help learners to understand the essential functions of an apparatus (of a given family). Like an illustrative video clip, such a scenario would help the learners to observe real functioning of the apparatus (with the possibility to repeat some activities), in addition to read and interact with results (feedback). This can be considered as a preparation for the learner to be familiar with the environment he/she is about to use. Modifying the sequence of certain activities in such a scenario leads to new results, but it uses always the same functions of the apparatus. This idea could also be creatively used in automated production systems (with respect to another platform than ELaMS) for staff training. Its automated generation would simplify the operations of software update and afford a time gain in creating new training scenarios by reusing parts from a library.

6. CONCLUSION

Electronic laboratories have become an essential need in the core of distant learning as well as in classical learning. However, they are not fully integrated in Learning Environments yet. As apparatuses are often made of different hardware even if they fill the same learning purposes, this situation slows down the dynamics of exchanging and reusing ELab learning scenarios. Authors from LIESP and LIMOS laboratories have began to work on a semi-automated design process to help ELab

designers to provide apparatuses based on automated discrete systems which enable LMS synchronization and generic learning scenario re-usability. This paper sums up previous works of both teams and depicts basic lines of this common project.. Automating the design process of the control part, the network interface and the HMI should permit to get time and cost gains. The concept could also be extended to automatic generation of demo scenarios and staff training on specific industrial automated systems.

7. REFERENCES

[1] Ashish Mani and C. Patvardhan. "A study of ict enabled laboratories". In Annual India Conference, 2006, pages 1–6, 2006.

[2] A. Baccigalupi, C. De Capua, and A. Liccardo. "Overview on development of remote teaching laboratories: from labview to web services". In Instrumentation and Measurement Technology Conference, 2006. IMTC 2006. Proceedings of the IEEE, pages 992–997, 2006.

[3] A. Bagnasco, P. Buschiazzo, D. Ponta, and M. Scapolla, "A learning resources centre for simulation and remote experimentation in electronics," in Proceedings of the 1st international conference on Pervasive Technologies Related to Assistive Environments (PETRA '08). New York, NY, USA: ACM, 2008, pp. 1–7.

[4] O.G. Bellmunt, D.M. Miracle, S.G. Arellano, A. Sumper, and A.S. Andreu. "A distance plc programming course employing a remote laboratory based on a flexible manufacturing cell". Education, IEEE Transactions on, 49(2):278–284, 2006.

[5] BENMOHAMED, H. ICTT@Lab: un environment pour informatique la génération et l'exécution de scénarios de téléTPs L'Institut National des Sciences Appliquées de Lyon, 2007.

[6] Khalid Kouiss, Fabien Chiron. "Distributed control systems: from design to real implementation with uml 2.0.". Int Conf on Industrial Engineering and Syst. Management IESM 05, MARRAKECH, MAROC, 12 pages, (2005).

[7] CHIRON, F. Contribution à la flexibilité et à la rapidité de conception des systèmes automatisés avec l'utilisation d'UML L'université Blaise Pascal de Clermont-Ferrand, 2007.

[8] I. T. Christou, S. Efremidis, T. Tiropanis, and A. Kalis, "Grid-based virtual laboratory experiments for a graduate course on sensor networks," IEEE Transactions on Education, vol. 50, no. 1, pp. 17–26, 2007.

[9] J.E. Corter, J.V. Nickerson, S.K. Esche, and C. Chassapis. "Remote versus hands-on labs: a comparative study". In Frontiers in Education, 2004. FIE 2004. 34th Annual, pages F1G–17–21 Vol. 2, 2004.

[10] J. E. Corter, J. V. Nickerson, S. K. Esche, C. Chassapis, S. Im, and J. Ma, "Constructing reality: A study of remote, hands-on, and simulated laboratories," ACM Trans. Comput.-Hum. Interact., vol. 14, no. 2, p. 7, 2007.

access," IEEE Transactions on Instrumentation and Measurement, vol. 55, no. 5, pp. 1551–1558, 2006.

[11] F. Davoli, G. Spanospano, S. Vignola, and S. Zappatore, "Labnet: Towards remote laboratories with unified

[12] M. Reza Emami and Michael Helander. "edesign". Proceedings of the Second CDEN International Conference on Education, Innovation and Practice in Engineering Design Kananaskis, Alberta, Kanada, 2006.

[13] M.Reza Emami and M.R. Adrian Martin, A. Emami. "Dynamic load emulation for robotic hardware-in-the-loop simulation platforms". IEEE International Symposium on Industrial Electronics Cambridge, UK, 2008.

[14] D. Gillet, A. V. N. Ngoc, and Y. Rekik, "Collaborative web-based experimentation in flexible engineering education," IEEE Transactions on Education, vol. 48, no. 4, pp. 696–704, 2005.

[15] Hugo L. Gosmann Adolfo Bauchspiess Bruno Guimarães, Alexandre S. Souza. "Internet based remote laboratory: The level control of three coupled water reservoirs". IFAC, 2000.

[16] M. Ishutkina, E. Feron, M. Casini, and A. Vicino. "An internet based laboratory for control of a safety critical system". In Systems, Man and Cybernetics, 2004 IEEE International Conference on, volume 3, pages 2707–2712 vol.3, 2004.

[17] R. Jain. "A revolution in education". Multimedia, IEEE, 4(1):1, 5–, 1997.

[18] J. Keller and R. Naues. "A collaborative virtual computer security lab". In e-Science and Grid Computing, 2006. e-Science '06. Second IEEE International Conference on, pages 126–126, 2006.

[19] T. Kikuchi, S. Fukuda, A. Fukuzaki, K. Nagaoka, K. Tanaka, T. Kenjo, and D.A. Harris. "Dvts-based remote laboratory across the pacific over the gigabit network". Education, IEEE Transactions on, 47(1):26–32, 2004.

[20] D. Kolberg, S and; Courivaud and M. Ozbek, "Lms and interactivity - technical issues for remote laboratories," in Proceedings of IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2007), IEEE, Ed., Athens, Greece, Sept. 2007.

[21] F. Kuester and T. C. Hutchinson, "A virtualized laboratory for earthquake engineering education," Computer Applications in Engineering Education, vol. 15, no. 1, pp. 15–29, 2007.

[22] Jean-Louis Lallican, Pascal Berruet, André Rossi, and Jean Luc Philippe. "A component-based approach for conveying systems control design" kananaskis, alberta, kanada. In ICINCO-SPSMC, pages 329–338, 2007.

[23] H.M. Langa. "Design and implementation of a laboratory practicals management tool". AFRICON, 2004. 7th AFRICON Conference in Africa, 2:965–968 Vol.2, Sept. 2004. 15-17 Sept. 2004.

[24] C. Landi, A. Liccardo, and N. Polese. "Remote laboratory activities to support experimental session for undergraduate measurements courses". In Instrumentation and Measurement Technology Conference, 2006. IMTC 2006. Proceedings of the IEEE, pages 851–856, 2006.

[25] S. Li and A. Khan, "Applying it tools to a laboratory course for measurement, analysis, and design of electric and electronic circuits," IEEE Transactions on Education, vol. 48, no. 3, pp. 520–530, 2005.

[26] E.D. Lindsay and M.C. Good. "Effects of laboratory access modes upon learning outcomes". Education, IEEE Transactions on, 48(4):619–631, 2005.

[27] J. Ma and J. V. Nickerson, "Hands-on, simulated, and remote laboratories: A comparative literature review," ACM Comput. Surv., vol. 38, no. 3, p. 7, 2006.

[28] Miguel Montes, Janusz Swiatczak, Peter J. Lesniewski, John C. Thomas, and Matthew Golczak. "Universal laboratory networking system (ulabnet)". In Instrumentation and Measurement Technology Conference, 2006. IMTC 2006. Proceedings of the IEEE, pages 986–991, 2006.

[29] B. Ozdogru and N. Cagiltay, "How content management problem of a remote laboratory system can be handled by integrating an open source learning management system? problems and solutions," in Proceedings of IEEE 18th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2007), IEEE, Ed., Athens, Greece, Sept. 2007.

[30] B. Ramaswamy, Y. Chen, and K. Moore, "Omni-directional robotic wheel - a mobile real-time control systems laboratory," in American Control Conference, 2006, p. 6.

[31] S. Rapuano and F. Zoino. "A learning management system including laboratory experiments on measurement instrumentation". Instrumentation and Measurement, IEEE Transactions on, 55(5):1757–1766, 2006.

[32] C. Rohrig and A. Bischoff. "Web-based environment for collaborative remote experimentation". In Decision and Control, 2003. Proceedings. 42nd IEEE Conference on, volume 3, pages 2514–2518 Vol.3, 2003.

[33] M.-G. Sere, J. Leach, H. Niedderer, D. Psillos, A. Tiberghien, and M. Vicentini, "Improving science education: Issues and research on innovative empirical and computer-based approaches to labwork in europe. Final report of project labwork in science education,http://formation.etud.upsud.fr/didasco/index.htm ," European Commission, Tech. Rep., 1998.

[34] S. R. Singer, M. L. Hilton, and H. A. Schweingruber, "Americas lab report: Investigations in high school science, committee on high school science laboratories: Role and vision," National Research Council, Tech. Rep., 2005.

[35] S. Sivakumar, W. Robertson, M. Artimy, and N. Aslam, "A webbased remote interactive laboratory for internetworking education," IEEE Transactions on Education, vol. 48, no. 4, pp. 586–598, 2005.