



## Supporting learning scenario authoring for Electronic Laboratories

Arnaud Lelevé, Saher Arnous, Patrick Prévot

### ► To cite this version:

Arnaud Lelevé, Saher Arnous, Patrick Prévot. Supporting learning scenario authoring for Electronic Laboratories. International Conference on Management of Emergent Digital EcoSystems, Oct 2009, Lyon, France. pp.83, 10.1145/1643823.1643925 . hal-00432559

**HAL Id: hal-00432559**

**<https://hal.science/hal-00432559>**

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.

# Supporting learning scenario authoring for Electronic Laboratories

Arnaud Leleve  
University of Lyon  
INSA Lyon  
F69621 Villeurbanne Cedex  
arnaud.leleve@insa-lyon.fr

Saher Arnous  
University of Lyon  
INSA Lyon  
F69621 Villeurbanne Cedex  
saher.arnous@insa-lyon.fr

Patrick Prevot  
University of Lyon  
INSA Lyon  
F69621 Villeurbanne Cedex  
patrick.prevot@insa-lyon.fr

## ABSTRACT

For a few years, Electronic Laboratories (ELabs: computer-aided local laboratories as much as distant and virtual ones) have been growing in the E-Learning panorama. Meanwhile, in major E-Learning activities, learning scenarios are nowadays written in a standard way (SCORM, IMS-LD) to be used by standard Learning Management Systems. The use of such scenarios in ELab trainings is appearing just now. However such scenarios are dedicated to one specific ELab and cannot be easily used for other similar ELabs. Since 2002, we work on the integration of learning scenarios into ELab platforms to help to re-use learning scenarios corresponding to similar apparatuses. Paper describing the global corresponding life-cycle have already been published. This paper introduces a tool which intervenes in the first step of this life-cycle: the learning scenario design process. It should help authors to design "generic" ELab learning scenarios, available for functionally compliant ELabs. This tool also aims at pre-testing such learning scenarios on a virtual apparatus to check scenario-apparatus communication.

## Categories and Subject Descriptors

K.3.m [Computers and Education]: Miscellaneous — *Electronic Laboratories*; L.1.2 [Knowledge and Media]: Learning Objects—*E-Lab activities as Learning Objects*; L.1.3 [Knowledge and Media]: Ontology/Taxonomy and Classification—*Classification of ELabs*

## General Terms

Design, Standardization

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

MEDES 2009 October 27-30, 2009, Lyon, France

Copyright 2008 ACM 978-1-60558-829-2/08/0003 ...\$5.00.

## 1. INTRODUCTION

For a long time, in the education world, hands-on training has been considered to be a vital activity, especially in the scientific (mathematics, chemistry, physics, medicine, etc) and technical (electronics, fluid mechanics, robotics, etc) disciplines [3, 9, 18]. Because of the recent spread of distance learning, encouraged by the improvement of Information and Communication Technologies (ICTs), hands-on training is nowadays also required in the distance learning context. If modern e-learning solutions today are well tuned for classical training (as e-lessons, virtual classrooms, e-projects etc), this is not yet true for distance hands-on training. In a context where learners cannot afford (for distance and financial reasons) to travel and attend an in-situ hands-on session, the solution lies in Distant Laboratories (DLabs). In-situ hands-on training also evolve: software gains ground and helps users (learners and instructors) in their hands-on task [18].

Accordingly, we use the term "Electronic Laboratories" (ELabs) to represent DLabs as much as computer-aided in-situ hands-on laboratories. These local laboratories may also be split in virtual (only based on simulation) and real (featuring real hardware) ones. Hybrid versions may also be encountered (simulations besides real apparatuses). Figure 1 summarizes different kinds of ELabs we could identify.

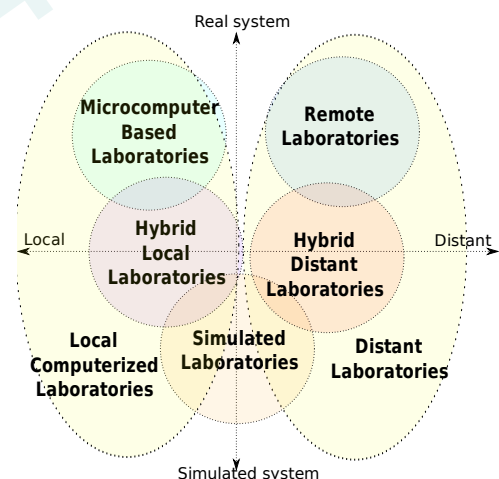


Figure 1: Different forms of ELabs according to distance from users and realness of handled system

Since nearly a decade, in major E-Learning activities, learning content authors have written learning scenarios in a standard way (using E-Learning standards such as AICC, SCORM or IMS-LD : see §3.1). These scenarios are to be used through standard Learning Management Systems (LMS), such as Moodle<sup>1</sup>, Claroline<sup>2</sup>, etc. Standards enable to use separate tools for authoring, indexing and delivering. So, authors can reuse learning material designed by others with other free or commercial software.

The use of learning scenarios to support ELab trainings is very recent, such as in [3, 15, 19]. As [20] reports, research to integrate DLabs into LMS has been slowed by the fact that first LMS were usually closed proprietary software systems that are often not customizable at all. This explains why the research carried out by scientists to provide laboratories has focused on means of teleoperating systems, rather than on integrating them into standard E-Learning platforms.

In this context, we began in 2002 [16] to work on a process which could help authors and instructors to re-use learning scenarios corresponding to similar apparatuses. A global learning scenario life-cycle was developed and a middleware (named “ELaMS”) to adapt “generic learning scenarios” to specific apparatuses has been designed.

This paper introduces a new tool, named “ESAT”, to help authors to design ELab learning scenarios. This tool enhances classical E-Learning authoring tools by delivering ELab specific user functions:

- it automates the declaration of required ELab functions inside the package manifest;
- starting from these requirements, it simulates a virtual ELab in order for the authors and instructors to test their ELab learning scenario from a standard LMS but without the real apparatuses (off-line).

This latter function should help authors to “debug” off-line their scenarios when the real ELab is busy or out-of-reach. Scenario-apparatus communication can be checked this way. It is clear that this tool can not reproduce the behavior of the real apparatus. So, it does not replace the real apparatus but it permits to track connections and data exchanges between the LMS launching an ELab scenario and the (fictive) apparatus.

Next section details the main principles which the generic life-cycle settles on. Part 3 introduces ESAT and part 4 summarizes the experimentation performed with this tool.

## 2. ELAB SCENARIO LIFE-CYCLE

### 2.1 Project genesis

Nowadays, learning scenario authors and instructors usually exchange standard learning scenarios (as SCORM packages for the most part). It helps the training community to design learning scenarios by reusing and adapting others. According to Klebl in [14], *a learning scenario is a social setting dedicated to learning, education or training. It is a process of interaction between people in a specific learning situation using resources for learning within a designed environment. People in role of learners perform activities directed towards learning objectives using resources for learn-*

*ing. Learners may work on their own or in a group of learners. They may be supported by teaching staff.*

ELabs could benefit from this creativeness if we could also design and exchange learning scenarios dedicated to hands-on activities. At the present time, the difficulty which prevents this, is that these scenarios are linked to specific apparatuses (the ones for which they have been written) and their own network interfaces (very dependant on used technologies). Moreover, adapting a given scenario to a similar apparatus forces the rewriting of the ELab-LMS interface code, which is not always possible when the source code is not available (Flash animations, Java applets, Labview VIs, ...).

Starting from these realizations, we began in 2002 to study a framework which could help in reusing DLab learning scenarios. The key idea is to help to reuse scenarios on similar apparatuses without having to re-develop the LMS-Apparatus interface as this step is not reachable by the most part of authors and instructors, who are not computer scientists. The interoperability of scenarios between similar apparatuses should help authors to design and improve (both local and distant) hands-on training scenarios by focusing on the training aspects rather than on technical aspects.

We define “similar apparatuses”, apparatuses having the same learning goals but possibly different hardware implementations. For instance, in optics, a classic optical bench features a lamp, one or two lenses and a screen. Lenses and screen can be moved as needed. Many manufacturers of optical benches exist but for a same curricula, the learning objectives are the same: to observe optical phenomena, create microscopes, to correct nearsightedness and farsightedness, etc. For distance learning, what this kind of bench requires is to be able to automate the manoeuvring of moving parts, visualise the resulting images on screen as well as an image of the whole bench.

So, our objective was to propose common tools to authors and instructors to help them in the design of ELab learning scenarios in order to enable these reusing features. The resulting architecture is depicted in figure 2. Every apparatus is available from the network through a Web server named “Apparatus virtualizer” which enables to call every necessary function through URLs and HTTP based protocols (HTTP-GET at least and, optionally, XML-RPC<sup>3</sup>, JSON<sup>4</sup>, SOAP<sup>5</sup>, etc). As ELab resources are addressed as URLs in standard packages, this permit to provide users with a link directly triggering a given apparatus function. ELaMS is a middleware aiming at managing a park of apparatuses. It provides functions to associate apparatuses to a given family and to test whether scenarios are compliant with them [5].

### 2.2 Related Works

Several projects have been conducted about Simulated Laboratory frameworks. The High Level Architecture (HLA) [10], issued for US Department of Defense was the first project as far as we know with this aim. More recent works about integration of HLA into SCORM packages have been published by Haynes [12]. Concepts developed in these works may be applicable in our project, but at the time of writing this paper, we did not have the opportunity to investi-

<sup>1</sup>see <http://moodle.org/>

<sup>2</sup>see <http://www.claroline.net/>

<sup>3</sup>see <http://www.xmlrpc.com/spec>

<sup>4</sup>see <http://www.json.org/>

<sup>5</sup>see <http://www.w3.org/TR/soap/>

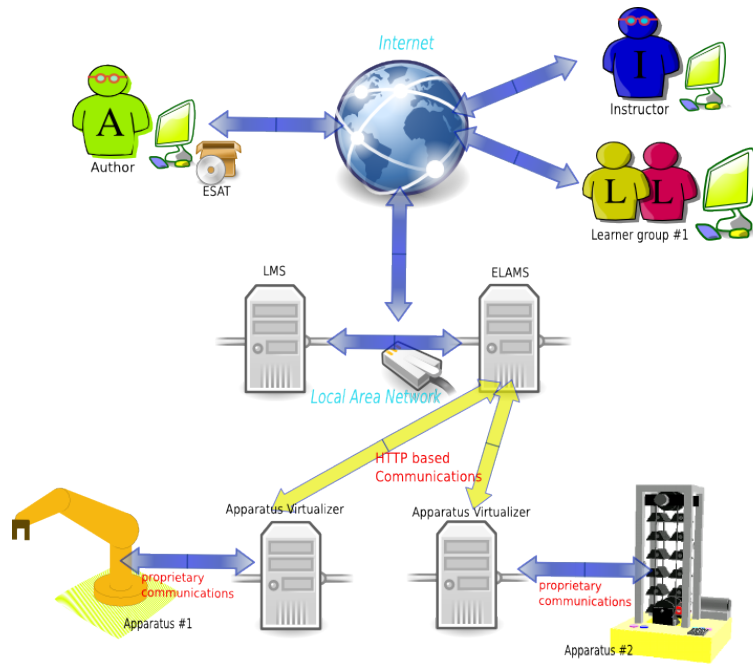


Figure 2: Global ELab architecture

gate more deeply. Much more recently, the Fraunhofer IFF Learning Platform[6] provides a VR to perform a very realistic virtual laboratory training. The platform features its own authoring system to create learning scenarios without being VR expert but rather learning content expert. This project is centered on unique and complex Simulated Laboratories. Our works apply at first to Remote Laboratories, even if they may be also used in the case of Simulated ones but with some conditions. This is why, these projects, interesting as they are, do not have the same primary goals and constraints.

A new EC-funded project, LiLa (Library of Labs, 2009-2011) [21], initiated by University of Stuttgart aims at developing an integrated platform for remote experiments and virtual laboratories. It provides learners and lecturers means to search for remote experiments and virtual laboratories, perform them through a 3D integrated web portal enabling their collaborative use. LiLa reuses BW-eLabs [13] open source framework dedicated to complex experiments (initially dedicated to nanotechnologies) and based on Web Services and Semantic Web technologies. BW-Labs are being extended to new disciplines such as Robotics under the frame of DFG Project "NetLabs". This architecture seems partly complementary with ours as we do not study the collaborative aspect of remote hands-on training nor the Human-System Interactions. A difference between our project and these ones concerns the scenarization: we found our project on E-Learning standards and standard authoring tools, which seems not to be the case in LiLa, as far as we know.

A very close work by Grout et al [11] has been recently published about an XML based language to describe an ELab structure and capabilities. This work is concurrent with our ontology OWL based ELab descriptions: it answers to the same need for formal description of ELabs but

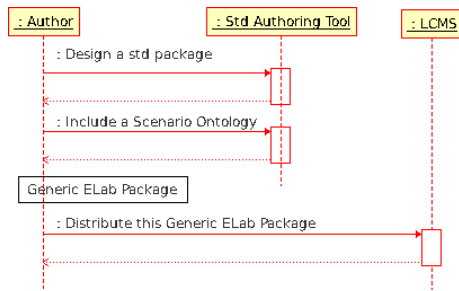
with another solution. This solution could be used to replace every ontology in our life-cycle (TO, ADO, SO).

The life-cycle presented in this paper is a concept which we brought, as far as we know, without concurrent work.

### 2.3 Life-cycle

This section sums up the whole ELab learning scenario life-cycle we proposed in [4]. As soon as a family of similar apparatuses has been identified (whose apparatuses are used under the same conditions on different hardware):

1. The "template manager" attributes a family name, for instance "optical bench with 1 lens".
2. Then, he specifies every generic (within the family) function necessary to teleoperate the apparatus ("move lens to position x", "switch the light on", "display web HMI", "display webcam view") into a "Template ontology" (TO) which is published on a unique web-based template repository: this is the step of "Generic Apparatus Specification".
3. Each apparatus manager describes his own apparatus (belonging to this family) by stating which function is provided by his own one and how to reach it (URL and available protocols : HTTP-GET by default and optionally: XML-RPC, JSON, ...in an ontology called "Apparatus Description Ontology" (ADO). As an example, generic function "*move lens to position x*" can be called up by sending a HTTP GET request with this function name and desired *x* value as arguments, to the apparatus virtualizer defined by a given URL. This is the step of "Specific Apparatus Description".
4. Each author designs "generic" learning scenarios which are associated with a family (in fact a template) of apparatuses. These scenarios embed calls to some of



**Figure 3: Learning scenario life-cycle: generic package design**

its previously mentioned generic functions. This is the “Generic ELab Scenario Writing” step (its applicability is limited to the family space, in this example the one of “optical benches with one lens”), see figure 3.

5. They then index their generic scenarios into public Learning Content Management Systems (LCMS) so that any instructor can find scenarios for a given apparatus family.
6. Once an instructor has downloaded a generic learning scenario belonging to the same family as his apparatus, the ELaMS middleware automatically tests the scenario–apparatus compliance to tell which functions are necessary for the scenario and which ones are available on his apparatus. This is the “Learning Scenario Compliance Test” step, see figure 4.
7. If compliant, a new (adapted) version of this scenario is automatically created by ELaMS (ELaMS updates scenario ELab related URLs to have them targeting towards this specific apparatus). This is the “Learning Scenario Adaptation” step.
8. Instructors have then to upload this adapted scenario into their LMS to make it available to their learners for this specific apparatus.
9. Instructors and learners use a standard LMS to interact with the previous scenario. This latter contains URLs targeted towards either ELaMS (which translates and redirect the call towards the corresponding apparatus) or directly the right apparatus, see figure 5.

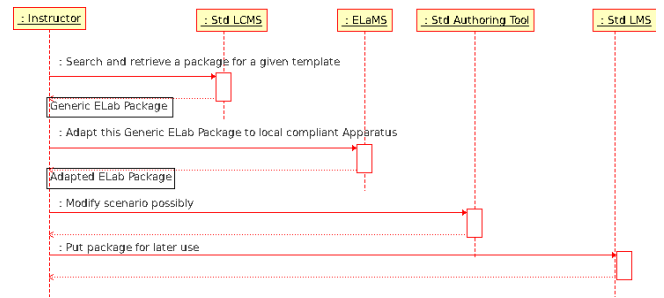
This paper content deals with the “ELab Scenario Writing” step.

## 2.4 Apparatus formal descriptions

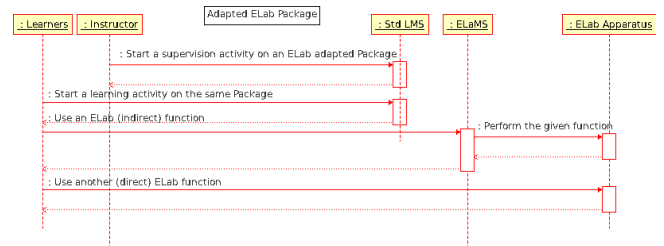
We needed representations of families of similar apparatuses (and the generic functions they should provide) manageable by both a software and humans. We have then taken inspiration from Semantic Web techniques and we have adopted the OWL<sup>6</sup> (Ontology Web Language) standard of World Wide Web Consortium (W3C)<sup>7</sup>. In order to avoid “reinventing the wheel” each time a new template is written for a new family of apparatuses, we have started to

<sup>6</sup>see <http://www.w3.org/2004/OWL/>

<sup>7</sup>see <http://www.w3.org>



**Figure 4: Learning scenario life-cycle: package adaptation to local apparatus**



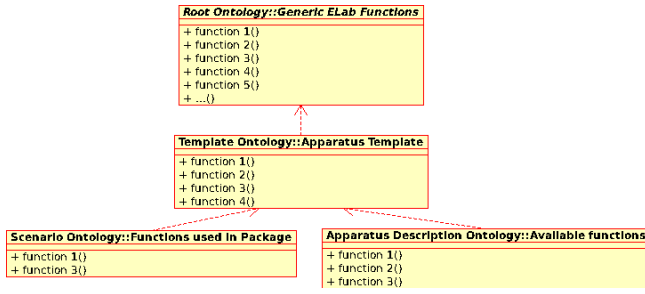
**Figure 5: Learning scenario life-cycle: package use for hands-on training**

build an extensible root ontology featuring standard components one can find in ELabs and their typical functions. This ontology should be extended each time new components (new instruments: voltmeters, wattmeters etc.) and corresponding functions (*get\_mean\_voltage*, *get\_peak\_power*, etc.) are necessary for the definition of a new template.

The main principle of using ontologies has been thought through and proposed to unfold in the following way (depicted in fig. 7):

1. Generic Apparatus Specification:  
when a new family of apparatuses is created, a new Template Ontology (TO) defining the corresponding template is written and published on a public web server. Templates declare usual functions necessary to handle apparatuses member of this family (for instance, for the family of industrial ovens, functions such as *raise\_power*, *decrease\_power*, *get\_oven\_temperature*, ... are declared). Apparatus functions are declared as classes in OWL syntax. These functions derive from basic ones defined in the root ontology; for instance, the function *get\_oven\_temperature*, defined in the oven template, derives from the function *get\_temperature* which is defined, in the root ontology, inside the group of “acquisition” functions.
2. Specific Apparatus Description:  
when a new apparatus is installed in a laboratory, it is assigned an ontology (Apparatus Description Ontology: ADO) which tells which functions declared in the associated template are really available (and how to call them) with this specific apparatus. This ontology imports the template ontology (TO) as a basis and adds necessary instances (individuals) of the classes defined in the template. This way, it formally describes available functions of one apparatus of a given family





**Figure 6: Ontologies used to represent ELab apparatus functions**

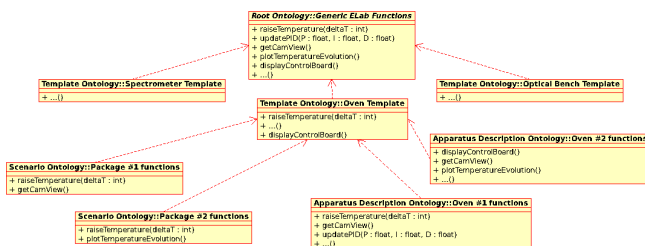
(for instance, *get\_oven\_temperature*, *decrease\_power*, ... are available and are called through URLs :  
[http://my\\_apparatus/get\\_temperature](http://my_apparatus/get_temperature),  
[http://my\\_apparatus/decrease\\_power](http://my_apparatus/decrease_power), ...).

### 3. Learning ELab Scenario Requirements:

when a new scenario is written for any apparatus in a family, it is assigned a Scenario Ontology (SO) which tells which functions of the Template Ontology is used in this specific learning scenario. As for the Apparatus Description Ontology, instances are declared in the SO in order to signal which template functions are necessary for the scenario to work properly.

When an instructor wants to use a given learning scenario on an apparatus of his, he uses ELAMS middleware to perform a "Learning Scenario vs Apparatus Compliance Test". At first, ELAMS checks that the Scenario Ontology and the Apparatus Description Ontology refer to the same Template Ontology. Next, ELAMS compares instances of required functions declared in the (SO) with instances of functions declared in the (ADO). When every required function is instantiated in the apparatus, the given learning scenario is fully compliant with it. When certain functions do not exist in this apparatus the instructor is notified about this lack and the corresponding learning activities which would require these functions. He will then have to decide how to proceed; either to modify corresponding resources and activities or to enhance his apparatus.

Note that using such ontologies is independent from and compliant with E-Learning standards: these ontologies are provided in the learning packages as resource files such as any other file necessary for the training process. The difference is that this file is intended towards ESAT and ELAMS softwares but not instructors and learners.



**Figure 7: Sample of a hierarchy of ontologies for ELabs**

## 3. ESAT AUTHORIZING TOOL

ELab authors are encouraged to use free or commercial authoring tools to edit their ELab learning scenarios. But these tools are not designed to manage laboratory resources, moreover according to the aforementioned life-cycle. In order to help these authors in their work, we propose here a design of ELab related functions, which we named "ELab Supporting Authoring Tool" (ESAT). In a first approach, we propose to integrate these functions in a software complementary to traditional authoring tools. In this approach, this software modifies a learning package which has already been edited by a standard authoring tool. This approach has the advantage to depend only on packaging standards (IMS-CP) and not on authoring tool technologies. Yet, it has the drawback to be less ergonomic than the second approach.

In a second time, we plan to integrate ELab functions into an open source authoring tool. Advantages and drawbacks are typically at the opposite of the first solution.

### 3.1 ELearning standards

To be able to exchange learning scenarios between different ELearning tools (mainly authoring tools, repositories and LMS), some standards nowadays co-exist:

- to identify and efficiently index training resources:  
Learning Object Metadata (LOM) [17] : widespread and integrated in most of next standards;
- to run training scenarios exchanged as packages:
  - Specifications from Aviation Industry CBT Committee (AICC) [2] : a little deprecated now but still widespread;
  - Shareable Content Object Reference Model (SCORM) [1], including AICC and some other standards: the most widespread nowadays (Moodle, Claroline, etc);
  - IMS Global Learning Consortium: Learning Design (IMS-LD) [8] : as for SCORM but more flexible. This standard is slowly spreading. As far as we know, only Coppercore runtime environment<sup>8</sup> is able to play IMS-LD packages but nowadays no open-source LMS really implements it.

More information about the expectations of the standardization according to the actors involved in a process of e-training are provided in [22].

### 3.2 Classic authoring tools

According to the E-Learning standard authors want to rely on, one can find many free and commercial authoring tools. We can cite, for SCORM: Reload<sup>9</sup> and eXe<sup>10</sup>, and, for IMS-LD: Reload Learning Design Editor<sup>11</sup>.

Such softwares help authors to create and edit learning objects (content display, exercises, quiz, ) without having to dive into XML, HTML, javascript, ... code.

<sup>8</sup>see <http://coppercore.sourceforge.net/documentation/ccrt.shtml>

<sup>9</sup>see <http://www.reload.ac.uk/editor.html>

<sup>10</sup>see <http://exelearning.org/>

<sup>11</sup>see <http://www.reload.ac.uk/ldeditor.html>

### 3.3 ESAT functions

Several functions are provided by ESAT, according to the steps in the global design process :

1. Association of a learning scenario with a family of apparatuses:
  - (a) a file resource is added to the scenario package. It is the "Apparatus Description Ontology" as defined earlier. It derives from a TO which URL is defined inside and which we will call here  $URL_{TO}$ . This file is used later by ESAT for creating a test package and by ELaMS to adapt this scenario to a specific apparatus.
  - (b) The resulting package must be independent from any specific apparatus. So URLs featured in resources related to ELab use are targeted towards the TO accompanied with arguments which tells which function to call. This way, ELaMS and ESAT will be able later to replace them by the real corresponding ELab URLs.
2. Simulation of the ELab interface:
  - (a) A generic ELab scenario is adapted towards the ESAT simulator, such as for a specific apparatus: its ELab related URLs are replaced by URLs targeting local ESAT simulator;
  - (b) an HTTP server is launched and is able to respond to requests from LMS on loading corresponding ELab resources;
  - (c) default responses are defined in the TO. User can override in real-time these defaults through ESAT GUI to test its scenario.

Technically speaking, the association steps are made by manipulating the manifest file present in each IMS-CP [7] standard package (used by SCORM and IMS-LD). This file is coded in XML. The coding of resources is very close in both SCORM and IMS-LD as it is mainly defined in IMS-CP standard. So ESAT is available for both standards without having to program twice the same functions. For the handling of the ontologies (TO and ADO for reading and SO for editing), libraries such as Jena<sup>12</sup> are useful.

### 3.4 ESAT as standalone tool

The following section introduces ESAT designed as a standalone application, to be used besides a standard authoring tool.

When used as a standalone tool, ESAT interacts with the ELab learning scenario through its generic package (in SCORM or IMS-LD format). To be able to efficiently edit a package and use simultaneously ESAT, this requires that the authoring tool loads and saves (at least, imports and exports) the standard generic ELab package (GEP).

Figure 8 details how steps are spread among authoring tool and ESAT. Steps  $Ax$  correspond to the authoring process. Steps  $Tx$  correspond to the testing process. Test Package can be generated directly at the first use of ESAT, beside the generation of the GEP. In a global continuous enhancement process, author has to modify his scenario. Therefore, he has to edit the GEP. This is why, generally, the test

ELab package (TEP) is generated from the GEP. This TEP is then launched on a LMS or a testing software such as "Reload Player"<sup>13</sup> so that author can "debug" his scenario off-line. This will not prevent him to test it with the real apparatus but it can make him gain much time when testing it at first off-line with ESAT as ESAT plays the role of a debugger for LMS-ELab data exchanges, in a certain way.

### 3.5 ESAT integrated into a standard authoring tool

Previous process shows that a continuous improvement process requires to open an authoring tool and ESAT several times, one after the other, as they exchange information only through the GEP. We plan to try to test the integration of ESAT into an open-source authoring tool in order to validate this concept. An interesting solution consists in providing ESAT as an extension or plug-in to an existing authoring tool.

The gain would reside, for the most part of it, in ergonomics: author would have no longer to open alternatively his packages with two different softwares.

### 3.6 Current state of this project

The life-cycle principles have been experimented in [5]. We proceeded to other experiments which results and lessons are about to be published. Preliminary experimentations on ESAT are detailed in next section. ESAT is now just a basic prototype aiming to validate the aforementioned concepts. It has not been featured with a convenient GUI yet.

## 4. EXPERIMENTATIONS

A preliminary experimentation has been proceeded in order to test the technical feasibility of the concept. A Simulated laboratory (SLab) has been used for this test. This SLab proposes the same functions as a real laboratory apparatus : an industrial oven. This SLab is a set of Python scripts which simulate the thermal model of the oven and a universal PID controller. We use a single instance of this simulator which runs in real-time to emulate a single real apparatus which would provide the same functions (same ADO) and the same way to call them through the network. This way, we can focus on ESAT functional test while we do not depend on a real apparatus vacancy.

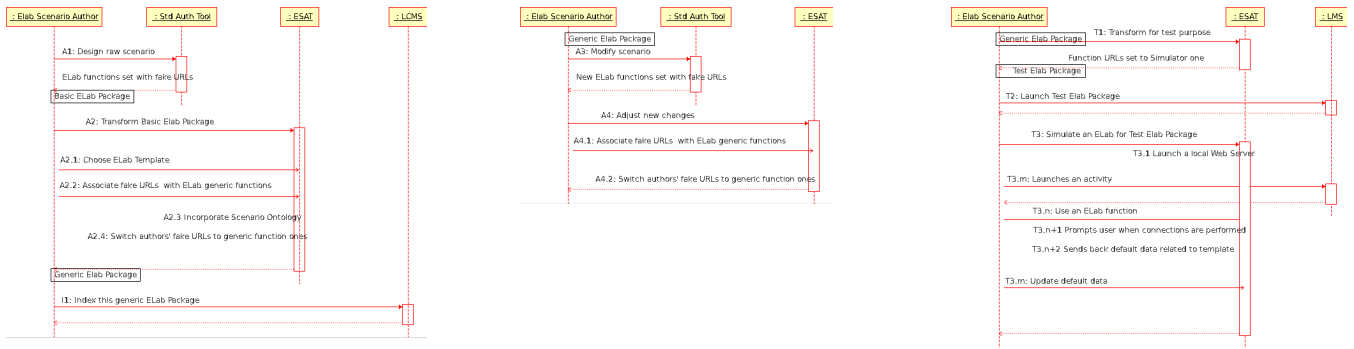
A Template Ontology (TO) had already been written for this kind of apparatus, so the author could start the experimentation at the beginning of the aforementioned process. Reload editor was used to generate a SCORM package of a typical lab. work for beginners in Automatic Control.

At the time of writing this paper, tests are not over. A first incomplete ESAT prototype has been coded in Python. However, we managed successfully to make ESAT simulate the opening of a web page which provides an AJAX based control board. This was performed through the declarations of the Scenario Ontology (SO), featured in the test package. The test package and the generic one were successfully generated by ESAT from the initial package.

The first conclusions of this experiment are that the generic ELab learning scenario is technically feasible. It only requires manipulations of IMS CP compliant content and ontologies written in RDF/OWL (XML format). Libraries can help to do this work but it seems that we have to stay

<sup>12</sup>see <http://jena.sourceforge.net/>

<sup>13</sup>see <http://www.reload.ac.uk/scormplayer.html>



**Figure 8: Chronology of ELab package authoring with ESAT as standalone tool for three main phases (from left to right): creation, modification and simulation**

into the Java world to make profit of them. This is unexpected realization constraint as we prototype in Python, despite OWL is an open specification broadly used in Semantic Web. In the worst case, XML editing is sufficient but requires much more coding. Tests are now to be performed on every defined function. We hope to get enough information to validate and then enhance the whole ELab learning scenario life-cycle and associated ontologies.

## 5. CONCLUSIONS

This paper introduced a solution to help ELab authors to design learning scenarios, written in SCORM or IMS-LD standard, belonging to a common family of apparatuses. The "ELab Supporting Authoring Tool" (ESAT) has been thought as functions complementary to standard authoring softwares. It can be embodied as a standalone application or integrated into an existing open-source authoring tool. This tool helps authors to define apparatus family and required functions. It can also help authors in testing and improving their scenarios by creating a test package which can be used to simulate the apparatus network interface when launched from a LMS. This tool can also be used with virtual laboratories which provide a network control interface and as soon as a corresponding "template" has been defined. Experimentations are to be continued and we also study related works such as [12] about integration HLA simulations and SCORM instructions.

## 6. GLOSSARY

- Kinds of laboratories :

**DLab** Distant Laboratory ;

**ELab** Electronic Laboratory (includes DLabs, LCLabs and SLabs);

**LCLab** Local Computerized Laboratory ;

**SLab** Simulated Laboratory ;

- Our frameworks for ELabs :

**ELaMS** ELab Management System ;

**ESAT** ELab Supporting Authoring Tool.

- Ontologies :

**ADO** Apparatus Description Ontology : to describe functions of a given apparatus ;

**SO** Scenario Ontology : to describe apparatus functions used in a given learning scenario

**TO** Template Ontology : to describe a family of apparatuses.

## 7. REFERENCES

- [1] Advanced Distributed Learning. *SCORM 2004*.
- [2] Aviation Industry CBT Committee. *AICC*.
- [3] A. Bagnasco, P. Buschiazzo, D. Ponta, and M. Scapolla. A learning resources centre for simulation and remote experimentation in electronics. In *Proc. of the 1st intl. conf. on PErvasive Technologies Related to Assistive Environments (PETRA '08)*, pages 1–7, New York, NY, USA, 2008. ACM.
- [4] H. Benmohamed, A. Leleve, and P. Prevot. Generic framework for remote laboratory integration. In *Proc. of 6th intl. conf. on Information Technology Based Higher Education and Training (IThET 2005)*, pages T2B/11–T2B/16. IEEE, July 2005. ISBN: 0-7803-9141-1.
- [5] H. Benmohamed, A. Leleve, and P. Prevot. Electronic laboratories: Ictt@lab experimentation. In *Proc. of 4th intl. conf. on Education and Information Systems: Technologies and Applications (EISTA'06)*, volume 1, pages 94–99, Orlando, Florida, USA, July 2006.
- [6] E. Bluemel, W. Termath, and T. Haase. Virtual reality platforms for education and training in industry. *International Journal of Advanced Corporate Learning (iJAC)*, 2(2):4–12, 2009.
- [7] I. G. L. Consortium. *Content Packaging Specification*.
- [8] I. G. L. Consortium. *Learning Design Specification*.
- [9] 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.*, 14(2):7, 2007.
- [10] J. Dahmann, R. M. Fujimoto, and R. M. Weatherly. The dod high level architecture: an update. In *WSC '98: Proceedings of the 30th conference on Winter simulation*, pages 797–804, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press.



- [11] I. A. Grout and A. C. R. da Silva. Remote laboratory description language based on xml. *International Journal of Online Engineering (iJOE)*, 5(Special Issue REV2009):25–34, 2009.
- [12] J. Haynes, S. Marshall, V. Manikonda, and P. Maloor. Enriching adl: Integrating hla simulation and scorm instruction using sita (simulation-based intelligent training and assessment). In *Interservices/Interindustry Training Education and Simulation Conference*, Orlando, Florida, USA, December 2004.
- [13] S. Jeschke. (bw) - elabs - knowledge management for virtual & remote labs. In *eSciDoc Days 2009*, FIZ Karlsruhe, June 2009. June 15-16.
- [14] M. Klebl. *Educational interoperability standards: IMS learning design and DIN didactical object model*, chapter 16. Springer Berlin Heidelberg, 2006.
- [15] D. Kolberg, S and; Courivaud and M. Ozbek. LMS and interactivity - technical issues for remote laboratories. In IEEE, editor, *Proc. of IEEE 18th intl. Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2007)*, Athens, Greece, Sept. 2007.
- [16] A. Leleve, H. Benmohamed, P. Prevot, and C. Meyer. Remote laboratory - towards an integrated training system. In *Proc. of 4th intl. conf. on Information Technology Based Higher Education and Training (ITHET 2003)*, Marrakech, Marocco, June 2003.
- [17] L. T. S. C. (LTSC). *Learning Object Metadata (LOM)*. IEEE.
- [18] J. Ma and J. V. Nickerson. Hands-on, simulated, and remote laboratories: A comparative literature review. *ACM Comput. Surv.*, 38(3):7, 2006.
- [19] 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 *Proc. of IEEE 18th intl. Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC 2007)*, Athens, Greece, Sept. 2007.
- [20] S. Rapuano and F. Zoino. A learning management system including laboratory experiments on measurement instrumentation. *IEEE Trans. on Instrumentation and Measurement*, 55(5):1757–1766, 2006.
- [21] T. Richter, D. Boehringer, and S. Jeschke. Lila: A european project on networked experiments. In *Proceedings of the Sixth International Conference on Remote Engineering and Virtual Instrumentation (REV 2009)*, Bridgeport, CT, USA, June 2009.
- [22] I. Varlamis and I. Varlamis. The present and future of standards for e-learning technologies. *Interdisciplinary Journal of Knowledge and Learning Objects*, 2, 2006.