

# Design and implementation of a virtual lab for supporting students in modeling, evaluating and programming secure systems

*Roberto Laschi<sup>1</sup>, Anna Riccioni<sup>1</sup>*

<sup>1</sup>DEIS (Department of Electronics, Computer Sciences and Systems),  
University of Bologna, Italy

**Key words:** *e-learning, virtual laboratories, IEEE LTSA*

## **Abstract:**

*In this paper we refer on the design and implementation of a learning environment for supporting Computer Engineering students in the evaluation of algorithms and protocols for Information Security. Such an environment helps students to reinforce conceptual knowledge and to gain practical programming skills through several tools for continuous and automatic support. The virtual lab includes several assessment tools and offers various formative paths for different user preferences.*

## **1 Introduction**

Broad consensus has been reached within the education research field on the need to facilitate the transposition of theoretical knowledge into practice [1]. Several advantages arise from this educational approach ranging from the reinforce of students learning, to the increased opportunity for a student to gain greater insight into subjects, to the possibility to acquire practical skills and long-lasting proficiency.

Virtual laboratories now offer interesting opportunities to address the above issues [2]. In particular in the context of Computer Engineering education, where integrating conceptual understanding with technical skills assumes a strategic importance, virtual labs can play a primary role in fostering teaching and learning.

Information Security represents a particularly interesting case study, since it relies on multiple principles related to different disciplines, like mathematics, statistics, number theory, cryptography, but also software engineering. The main goal for learners is to model, evaluate and develop secure systems, and in order to achieve it students must gain a deep knowledge and understanding at different levels of abstraction: they should learn to verify mathematical theorems, analyze statistical properties, recognize and design the practical applications of the studied concepts, evaluate the effectiveness, the efficiency and the robustness of their intended solution. To this end, it is crucial for learners to rely on a suitable experimental environment for supporting their practical activities.

## **2 S-vLab**

In order to address the depicted issues we designed S-vLab, an open-source virtual laboratory for supporting teaching and learning within an Information Security course [3]. The S-vLab project started in 2005 at the University of Bologna and is pursuing a multi-year development process, that aims to accomplish a step-by-step quality assessment strategy.

S-vLab is inspired to the constructivist approach, which highlights the strategic importance of providing students with different tools to let them autonomously pursue their formative goals, according to their specific needs and preferred learning procedure [4, 5]. One of our virtual lab main features is the graphical editor where students can model their systems or algorithms and then simulate their behavior. S-vLab automatically stores the experimental data yielded in each exercise, to make it easier for students to analyze them through several specific tools related to different scopes: for instance, some tools are aimed at verifying statistical properties, while others allow to evaluate the performance of a system in terms of execution time and needed resources. S-vLab also helps students to implement and test a software prototype of their system; there, Java is the reference language, because of its widespread adoption both in the academic and in the professional context. In fact, the virtual lab provides a close connection to a standard Java Integrated Development Environment where learners can build their prototype, by writing Java code from scratch or by reusing some automatically generated didactic samples. At any time during the exercitation, students can consult one of the three help figures available: the Mentor, the Expert and the Judge. The Mentor purpose is to provide real-time feedback to the learner, in order to make him aware of his actions, and to keep him motivated and involved within the learning experience. The Expert allows browsing a wide set of resources related to the ongoing exercise, ranging from security standards to Java technical documentation. Finally, the Judge observes and evaluates the student's activities within the virtual lab; it also guides his learning path according to the settled formative goals and to several learner's characteristics, such as the starting competence and the preferred learning procedure. Specifically, in the depicted reference context three learning procedures appear to be particularly interesting: learning by doing, learning by example and learning by exploring. Students following the learning by doing style usually learn through autonomous experimentation and self-determined case studies; on the other hand, the learning by example procedure relies on step-by-step, guided activities. Finally, the learning by exploring strategy is based on a preliminary study of the subject of interest, followed by some related exercises.

Figure 1 outlines the main S-vLab features.

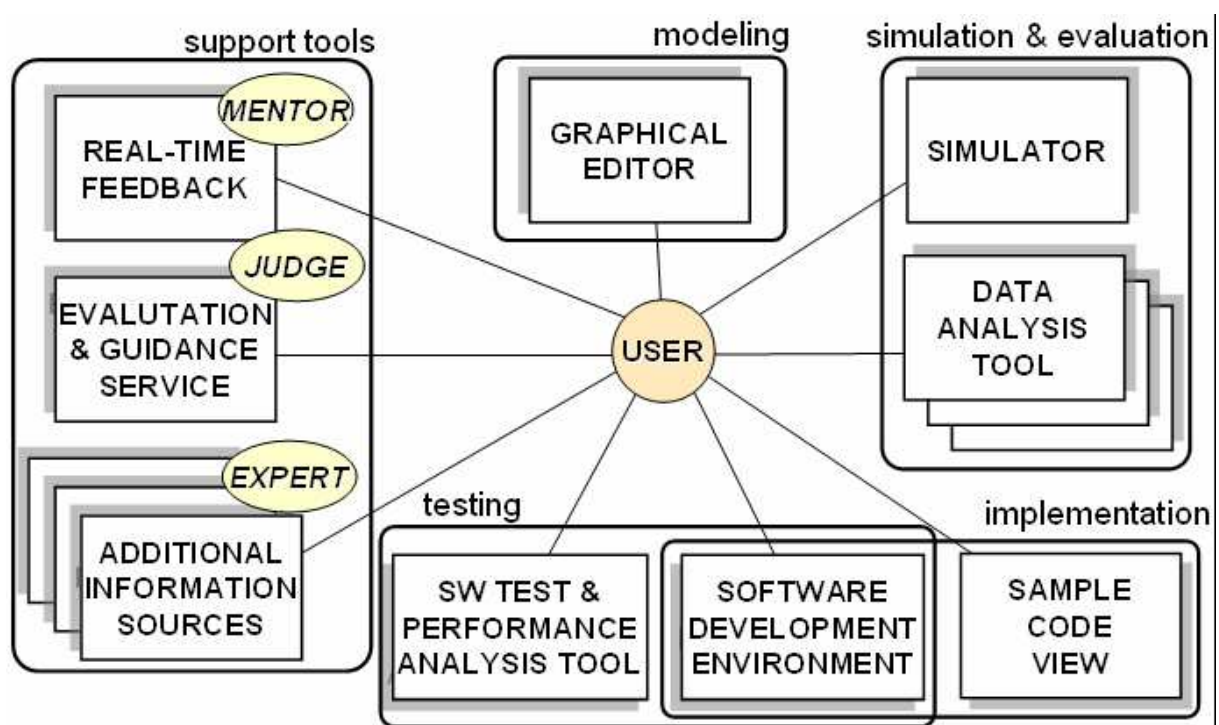


Figure 1: S-vLab main features

A typical exercitation session within S-vLab follows the depicted protocol.

1. the learner performs a login operation
2. the learner negotiates with the Judge the desired learning goals
3. if the learner is performing his first access, he could be required to fill a survey in order to allow the Judge to build a starting user profile
4. the learner takes a brief quiz aimed at estimating his starting knowledge with regards to the settled formative goals
5. the Judge, according to the user's competence and goals, and following a decision strategy defined by the teacher, proposes a suitable list of tasks
6. the learner performs the suggested exercise
7. the learner performs a second quiz, aimed at identifying improvements with respect to the placing test
8. the Judge automatically updates the user profile

The depicted protocol reveals how the students activity tracking capability is a crucial issue. In fact, it provides a set of aggregated information related to the students behavior, suitable for extracting useful hints for learners self-evaluation, and data regarding the students profile on which the Judge's activity can be based. Furthermore, also teachers can greatly benefit from the tracking feature, as it allows to identify the hardest tasks for the students to complete, and, as a consequence, to find the related exercises and notions that probably need to be revised. Finally, teachers can leverage such information for searching for possible winning learning procedures. In fact, by analyzing the provided data it is possible to identify whether there is a strategy which is more likely to help students to achieve the given goal, by reducing, on the average, time and efforts needed to complete a task. In this case, teachers can modify accordingly the preferred pedagogical policy used by the Judge to compose the exercise proposals.

The discussed pedagogical features imply three main non-functional requirements for the virtual lab implementation: extensibility, portability and reuse. In fact, it is crucial to allow gradual extensions in order to support additional learning styles and to provide new functionalities for widening the exercitation options. Also the planned multi-year development process strongly relies on extensibility for guaranteeing systematic assessment of successive, expanded releases of the ongoing project. Finally, portability and reuse, by simplifying integration and interaction with external or pre-existent tools and services, significantly facilitate the provision of new features.

Since the adoption of a plug-in architecture, along with compliance with standards, can effectively support the accomplishment of the depicted requirement, S-vLab is designed according to the IEEE Learning Technology System Architecture (LTSA) reference model and implemented within the Eclipse Community framework [6, 7].

IEEE LTSA proposes a five-layer structure specifying a high level architecture for information technology-supported learning, education and training systems: there, Layer 3 lists the architectural components corresponding to the main interfaces that should be considered, and also describes how they interact with each other, as shown in Figure 2. There, the Learner Entity represents an abstraction of a learner, while the Coach symbolizes a tutor or a teacher. The Evaluation process purpose is to assess the learner entity, and finally the Delivery manages the presentation of learning contents and tasks.

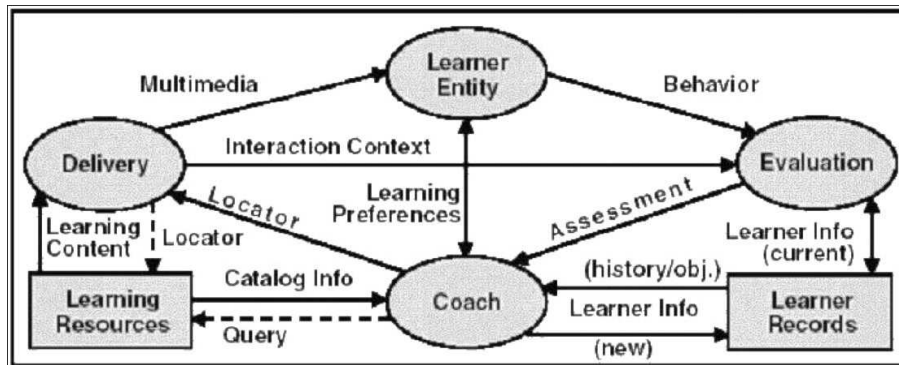


Figure 2: IEEE LTSA reference model, layer 3

For supporting such a model, we rely on the adoption of a plug-in architecture, which is essential for assuring extensibility. Once chosen this option, the Eclipse Community was a natural framework for building the virtual lab, since the great availability of open source projects offers a precious help for speeding up the implementation process.

## 2.1 Project history

The whole design and development process of S-vLab has been scheduled across three main steps, in order to effectively assess the project quality at each milestone.

The first evaluation phase, whose main purpose was to verify the effectiveness of the intended functional requirements, relied on the experimentation of vLab 1.0, a Java-based stand-alone application where students can only perform experiments strongly guided by the teacher [8]. The experimentation conducted by making such a virtual lab available to around 100 students confirmed the effectiveness of the chosen approach and served as a successful acceptance test, yet remarking the urgent need for supporting an active role for the students.

Thus, vLab 2.0 represents the second project milestone: it is based on the Eclipse platform and provides learners with several tools for performing their desired investigations, also allowing students with different starting competence and different formative goals suggested by the teacher to forge various learning paths. The survey conducted after vLab 2.0 experimentation on a second student generation showed that learners greatly appreciated their involvement in building their formative paths, as well as the provision of a wide set of freely available tools for autonomous exercises [4].

However, in order to provide a scored evaluation at the end of each exercise, and to suggest a customized formative experience to each learner, it is crucial to extend the virtual lab tracking features, and to integrate S-vLab within a distributed framework: in fact, a distributed architecture is essential for allowing teachers to gain a deeper insight into the learners' activities, their profiles and the strategies for evaluating their formative processes. The experimentation of such an environment, representing the intended outcome of our third milestone, will start in the next academic year and will be supported by a new release of the virtual lab.

## 2.2 Architecture and implementation insights

The S-vLab architecture adopts a client-server approach: it includes two different clients, one for the students and one for the teacher, and a data server for supporting tracking and updating operations.

The student client realizes the virtual lab application. It consists of two logical tiers rooted on a support platform: the upper tier, called User Interface, includes several modules corresponding to the main tools and functionalities offered to students, and it relies on the

Elaboration layer for accessing the local database. In fact, each student client includes a database storing information related to the available activities within the virtual lab, the history of the performed exercises, some user's data such as its learning goals and preferences, and finally the decision strategies which guides, within the virtual lab, the selection of the proposed formative task which is best suited to the learner's needs.

Each student client is independent and can be run as a stand-alone, concentrated application. This allows students to perform their exercitations wherever and whenever they want, independently of availability of an internet connection. However, each student client can occasionally establish a connection with the data server for possible upgrades and for tracking purposes. In fact, the data server hosts a global database for archiving information on the students' activities and performance within the virtual lab: there, the learners behavior is described in terms of most used tools and features, time spent on different tasks, scores obtained in quizzes, type and context of occurred errors, settled learning goals. Such information contribute to the inference of the student's preferred learning procedure. In addition, the global database stores the list of possible activities within the virtual lab and the strategies, defined by the teacher, for choosing the most suitable task to be suggested. The student clients can thus connect to the data server both for sending information about the performed exercises, and for synchronizing the activities and pedagogical strategies database. Such an approach allows a global traceability, related to the whole set of students who exercise through the virtual lab, yet .

On the other hand, teacher accesses the data server through a different client, whose purpose is to facilitate data analysis and inference of information for possibly modifying the adopted pedagogical strategies. It is also possible to leverage such clients in order to update the data server archives by adding new assignments.

The resulting architecture is shown in Figure 3.

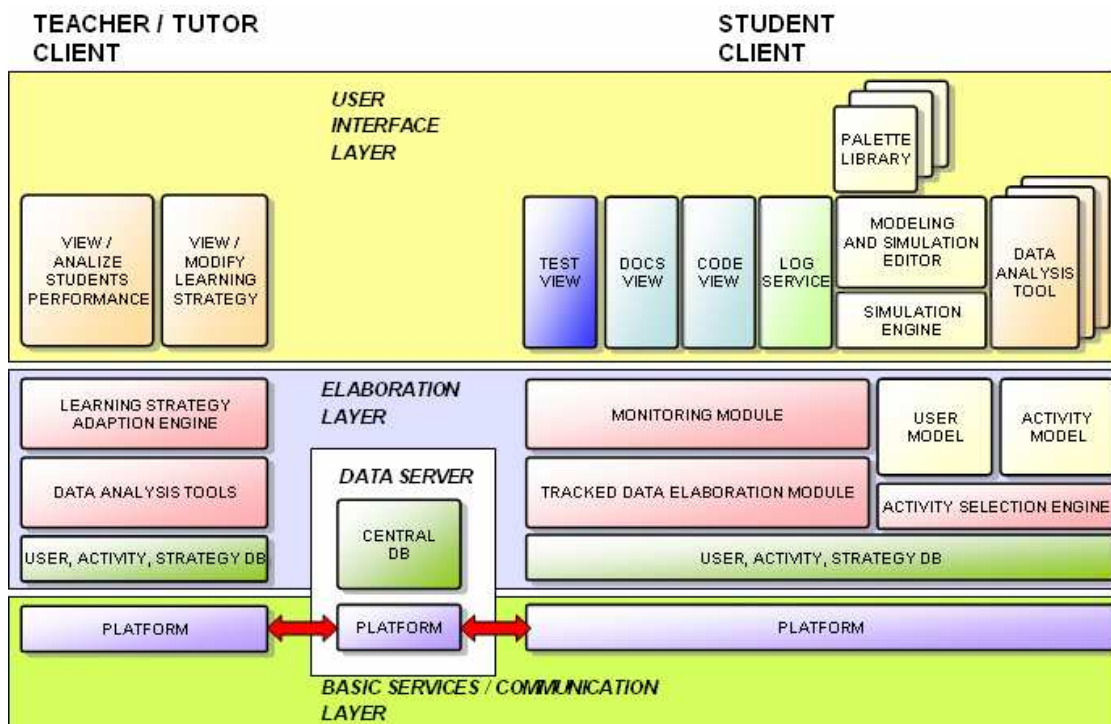


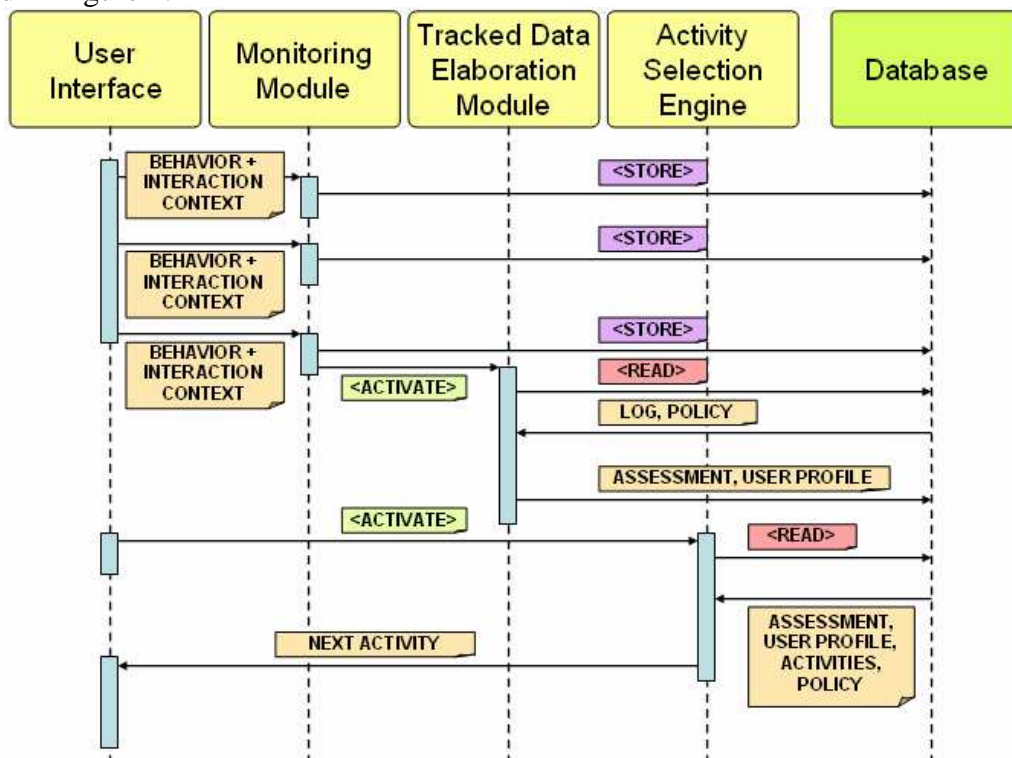
Figure 3: S-vLab distributed architecture

The implementation of the depicted architecture relies on the Eclipse technology: in fact the Eclipse platform provides basic services for coordinating the several interacting modules

and for supporting client-server communication. All the upper modules consist of Eclipse plug-ins and features.

In the IEEE LTSA reference model, the Evaluation process gathers information regarding the student’s behavior, the interaction context and the learner’s current profile: such data are essential for building a set of assessment information which is then sent to the Coach process. Once received this judgment, the Coach updates the user profile accordingly, in order to rely on up-to-date information when it will be called to identify the learning activity which best suits the student’s starting competence, preferences and previously negotiated learning goals.

The described process in S-vLab derives from a continuous interaction between three main modules within the student client: the Monitoring Module (MM) the Tracked Data Elaboration Module (TDEM), and the Activity Selection Engine (ASE). Their interaction is outlined in Figure 4.



- *User Interface*: contributes to the student’s activities monitoring; each module notifies to the *Monitoring Module* the learner’s behavior along with the interaction context
- *Monitoring Module*: stores in the *database* each single notified action
- *Tracked Data Elaboration Module*: is activated when the MM receives a message notifying the end of the exercitation or the end of the work session; it retrieves from the *database* the raw logs stored during the last exercitation or work session, as well as the assessment policies settled by the teacher; then, it aggregates such data and provides:
  - a. an evaluation of the latest activities performed by the student;
  - b. the corresponding updates for the current user profile;
- *Activity Selection Engine* purpose is to guide students through the available formative activities, by suggesting a customized learning path: thus, each time a student starts a new exercitation, the *Activity Selection Engine* first manages a learning goal negotiation step, and then proposes the tasks best suited to the student’s current profile, formative objectives and preferences

Figure 4: Activity tracking sequence diagram

### 3 Evaluation and quality assessment process

For S-vLab 3.0, that implements the depicted architecture, we designed a complete and detailed test plan. Our purpose is to formalize a systematized assessment strategy for evaluating the effectiveness of S-vLab in fostering students' learning and for verifying the users' acceptance and satisfaction with the tool.

#### 3.1 Evaluation tools

Our evaluation plan is rooted on three main tools, which are summarized in Table 1, and aimed at making the virtual lab assessment the most neutral, accurate and fair as possible.

Tool	Description		Goal
Quizzes	Pre-test and post-test before and after each exercise		Measure starting level and possible knowledge increment
Exercitation	<i>Single student oriented</i>	The given assignment is completed by each student working on his own	<ul style="list-style-type: none"> <li>• <i>formative goal</i>: student is expected to complete the given assignment</li> <li>• <i>experimental goal</i>: to identify each student's preferred learning procedure</li> </ul>
	<i>Team oriented</i>	The given assignment requires students to work in groups	<ul style="list-style-type: none"> <li>• <i>formative goal</i>: students will learn how to forge a path towards a goal, by working in group and dealing with different learning procedures</li> <li>• <i>experimental goal</i>: to observe cooperation between students relying on different learning procedures</li> </ul>
Feedback	Students can provide personal opinions on their formative experience		<ul style="list-style-type: none"> <li>• to make students sum up their formative experience through the virtual lab</li> <li>• to collect suggestions for improving S-vLab features and quality</li> </ul>

**Table 1: Evaluation tools**

The depicted tools play a strategic role within the intended evaluation plan, which is described in the following section.

#### 3.2 Evaluation plan

The typical exercitation is structured as follows.

At the end of each week during the course we propose a not compulsory exercitation. Each exercise is introduced through a brief face-to-face lecture, and also published on the course website along with an user guide and tips on how S-vLab should be used: students can thus autonomously perform the proposed tasks at home. The exercitation starts with each student taking on his own a quick, preliminary test, aimed at estimating his starting knowledge. The quiz includes a few closed questions, such as multiple choices, true/false or

fill-the blank, and it focuses the main themes the exercise concerns. Its purpose is to probe the student's prerequisites and his mastery of the notions and skills related to the topics covered in the assignment, and to estimate a starting knowledge level that will be compared with the final level, so as to provide a rough evaluation of possible after the practice with S-vLab.

Depending on the type of exercitation (single or group oriented), students perform their tasks working on their own or cooperating in small teams. In the latter case, groups will likely include three people, since small groups should be preferred in order to be able to recognize each member contribution, and research shows that working in pairs is not the best choice, as usually there is one component who prevails on the other [9].

At the end of the work session, each learner takes a second, equivalent test on the covered topics. Also this quiz includes a few closed questions, but is slightly different from the first: in fact, facing new questions can help in making students answer thinking about the practice they have preformed, instead of recalling the answers given in the first quiz. Since each test yields a score, it is possible to compare the results obtained by each student before and after the exercise, to estimate whether it helped in gaining a better understanding of the subject.

In addition, students can voluntarily compose written reports on their work, possibly including additional autonomous exercises and their related considerations, and then send it to the teacher before the final exam.

Finally, a key point of the evaluation process is represented by the collection of students' feedback, which includes both a personal judgment of the experience with the virtual lab, and critiques and suggestions for future versions. The former kind of information, in order to be neutral and honest, will be gathered through an interview run by an external person, after the student completed the final exam of the course, and only for those who provide evidence of having used S-vLab. On the other hand, suggestions and feedback related to identification of possible bugs in the application can be collected at any time, via mail, during the course, in order to promptly respond to urgent issues.

### **3.3 *Expected outcomes***

The discussed evaluation plan allows students to gather continuous feedback on their performance, and teacher to collect a conspicuous set of information to evaluate the effectiveness of the chosen approach, as well as to identify the difficulty level perceived by students for the covered topics.

To this end, the activity logs tracked by the virtual laboratory are particularly important: in fact, for each student the logs store the improvements through the various work sessions, in terms of reached objectives and scores obtained in subsequent quizzes. Furthermore, the activity traceability is especially useful for teachers: in fact, it allows a deep analysis of several aggregated information related to the students behavior. This significantly facilitates both the identification of tasks or materials to be revised, as well as of possible winning learning procedures; teachers can thus update assignments, lecture notes and preferred pedagogical policies accordingly.

Table 2 summarizes the data available for teachers at the end of the course.



<b>Outcomes – information available for teachers</b>	
<b>Written documents</b>	Written reports about the exercitations, that students can autonomously compose and send to the teacher
<b>Logs</b>	<i>Aggregate data</i> <ul style="list-style-type: none"> <li>• percentages showing the estimated students' preferred learning style among the three considered (learning-by-doing, learning-by-example, learning-by-discovery)</li> <li>• correlation between the preferred learning style and the success rate in the exercitations, that can be intended as the highest improvements (related to the highest differences between first and second test for each exercitation), or the highest score (where the score can result from a series of indicators related to the scores obtained in the tests, to the elapsed time in lab, ...)</li> <li>• which tasks / tests students are likely to fail</li> <li>• which tasks / tests students are likely to perform too easily</li> </ul>
	<i>Detailed data</i> <ul style="list-style-type: none"> <li>• detailed reports on each student's activities</li> <li>• each report can be edited by the teacher, in order to add evaluation related to written documents, to extra activities, to the student observed behavior</li> </ul>
<b>Interviews</b>	At the end of the course, after the final exam, an external person interviews the students in order to gather: <ul style="list-style-type: none"> <li>• personal judgment over the learning experience</li> <li>• possible feedback on how to improve the virtual lab, as well as the course contents and organization</li> </ul>

Table 2: Evaluation outcomes

## 4 Concluding remarks and future work

In this paper we discuss our approach to the design of a virtual lab for fostering teaching and learning within Information Security courses. This virtual lab revealed as a valuable support both for teachers and learners, by allowing the former to analyze the students behavior and by suggesting the latter some customized, and thus more effective, learning experiences. To this end, the implementation relies on a distributed and plug-in based architecture and consists of two different clients, for teachers and for students, in order to provide each involved actor with only and all the features needed for his role. In addition, both clients refer to a data server aimed at collecting information on students performance and profiles: such an approach permits to build a comprehensive archive by collecting data at different times, related to each student's individual exercitations and even addressing different student generations. This is a crucial issue for supporting statistical analysis of possible learning procedures that, in well defined contexts, tend to facilitate students learning: in fact, the more data will be available for examination, the more the inferred results can be considered noteworthy and accurate.

In our experience, structuring such a project on a multi-year process is a winning choice, since it facilitates the gradual development of a complex system by dealing, at each step, with a subset of the defined constraints and requirements, that can also be revised or extended according to the users' feedback. Furthermore, relying on standards offers significant

opportunities for integrating existent or third-parties tools with the ongoing project, in order both to speed up the implementation phase and to provide easily new features. Finally, adopting a step-by-step development approach significantly simplifies the project quality assessment.

Given the encouraging feedback collected during the first two years of S-vLab experimentation, our future efforts will be addressed both at designing new features and at fostering the virtual lab evaluation services. Our primary goal for the upcoming experimentation session consists in a systematic and formalized assessment plan for verifying S-vLab suitability in supporting learning, and for classifying the user profiles, in order to gather novel information for improving the personalization and customization features. Experimenting specialized versions of S-vLab within different formative domains represents a further interesting direction.

## References:

- [1] D. Kolb: *Experiential learning: experience as the source of learning and development*; Prentice-Hall Englewood Cliffs. 1984.
- [2] J. E. Corter, J. V. Nickerson, S. K. Esche, C. Chassapis, S. Im, J. Ma: *Constructing reality: A study of remote, hands-on, and simulated laboratories*; *ACM Transactions on Computer-Human Interactions*, 14(2):7. 2007.
- [3] Website of the course on Information Security, University of Bologna: <http://lia.deis.unibo.it/Courses/TecnologieSicurezzaAK/S-vLab.html>.
- [4] A. Riccioni, E. Denti, R. Laschi: *An experimental environment for teaching Java Security*; *Proc. of the Sixth International Conference on Principles and Practice of Programming in Java*. Modena, Italy, September 2008.
- [5] J. B. Black, R. O. McClintock: *An interpretation construction approach to Constructivist Design*; B. Wilson (Ed.) *Constructivist Learning Environments*. 1995.
- [6] IEEE P1484.1/D9, 2001-11-30, Draft standard for Learning Technology – Learning Technology Systems Architecture (LTSA)
- [7] J. McAffer, J.-M. Lemieux: *Eclipse Rich Client Platform: designing, coding, and packaging Java applications*; Addison Wesley Professional. October 2005.
- [8] R. Laschi, A. Riccioni, P. Suzzi: *Learning by doing: vLab, a virtual laboratory for Computer Engineering education*; *Proceedings of the Thirteenth International Conference on Distributed Multimedia Systems*. San Francisco Bay, September 2007.
- [9] C. S. Claxton, P. H. Murrell: *Learning styles: implications for improving educational practices*; ASHE-ERIC Higher Education Report No. 4. 1987.

## Authors:

Roberto Laschi, Full Professor  
University of Bologna, DEIS (Department of Electronics, Computer Sciences and Systems)  
v. Risorgimento 2, 40136 Bologna, Italy  
[roberto.laschi@unibo.it](mailto:roberto.laschi@unibo.it)

Anna Riccioni, PhD Student  
University of Bologna, DEIS (Department of Electronics, Computer Sciences and Systems)  
v. Risorgimento 2, 40136 Bologna, Italy  
[anna.riccioni@unibo.it](mailto:anna.riccioni@unibo.it)