

A Virtual Laboratory Structure for Developing Programming Labs

[doi:10.3991/ijet.v4s1.789](https://doi.org/10.3991/ijet.v4s1.789)

Josep Prieto-Blázquez¹, Jordi Herrera-Joancomartí² and Ana-Elena Guerrero-Roldán¹

¹ Open University of Catalonia, Barcelona, Spain

² Autonomous University of Barcelona, Barcelona, Spain

Abstract—Virtual Laboratories in a Virtual Learning Environment are indispensable spaces for developing practical activities. This paper proposes an integrated structure for Virtual Laboratories for undergraduate degree programmes in Computer Engineering and Software Engineering. The general structure proposed is based on the experience gained in designing and developing Virtual Laboratories during the past eleven years in a virtual university. Technological, pedagogical, strategic and academic staff resources are included in the Virtual Laboratories structure. The proposed structure has been applied to different computer areas and this paper presents the special case of a Virtual Programming Laboratory. Its effectiveness has been evaluated by the students using a web questionnaire.

Index Terms—A Practical Laboratory Activities, Virtual Laboratory, Virtual Learning Environment and Virtual Programming Laboratory

I. INTRODUCTION

During the past fifteen years, a huge growth in universities that need to diffuse knowledge and extend lifelong learning opportunities to students through distance learning environments has been observed.

Today, there exist several scenarios of distance learning. This paper deals with the virtual modality named *Virtual Distance Learning*, where all the processes, academic and administrative, are carried out in a virtual environment [1].

The *Open University of Catalonia* (UOC)¹ is a fully Virtual University that uses a *Virtual Campus* to teach in a Virtual Learning Environment (VLE). The UOC was created in 1994 to respond to the new reality and fulfill the new needs of students. The Virtual Campus contains similar facilities to those that can be found on a conventional university campus: teaching, research, dissemination of knowledge, student services, as well as Virtual Laboratories. Nowadays, there are three undergraduate degree programmes in engineering: *Software Engineering*, *Computer Engineering* and *Telecommunication Engineering* offered by the UOC.

According to *computer curricula 2005* [2], practical laboratory activities are an essential part of any computer curriculum since they strengthen concepts presented during lectures. For that reason, new virtual spaces are required in a Virtual University so practical activities can be carried out. Such spaces are called Virtual Laboratories

(VLabs) and should include different kind of activities ranging from designing and implementing solutions, to testing and documenting several applications, systems and processes.

The main contribution of this paper is the conception and definition of a general structure for VLabs, applied to a specific Virtual Programming Laboratory (VPLab), that allows students to perform practical activities on undergraduate degree programmes in *Computer Engineering* [3] and *Software Engineering* [4], in a VLE.

The paper is organized as follows. In Section II a revision of the existing proposals for VLab is provided. Section III details a general structure for VLabs and describes an implementation for a VPLab at UOC. This general structure is based on the experience gained in the design and development of VLabs during the past eleven years. The value of this proposal has been analysed through a survey that gathered students' feedback. The results of the survey are presented in Section IV. Finally, Section V provides a brief conclusion.

II. EXISTING PROPOSALS FOR VLABS

The term VLab is defined in different ways in the literature. For instance, Chiu [5] takes a simple vision of a VLab as a local computer hosting. Others authors include simulation capabilities [6] whilst some consider VLabs as an extension of a remote laboratory [7, 8]. A more accurate definition can be found in [9] where VLabs are defined as labs that “*leverage modeling, simulation, and information technologies to create an immersive, highly interactive virtual environment tailored to the needs of researchers and learners*”. Taking into account these previous definitions, this paper defines a Virtual Laboratory (VLab) as an interactive virtual space that incorporates all the technological, pedagogic and human resources for carrying out practical activities, adapted to the needs of the students and teachers in a virtual learning environment.

Different types of VLab have been proposed for specific purposes. The most important in the fields of Software Engineering and Computer Engineering are the following:

- Virtual Programming Laboratory (VPLab).
- Virtual Operating System Laboratory (VOSLab).
- Virtual Database Laboratory (VDBLab).
- Virtual Networking Laboratory (VNLab).
- Virtual Computing Architecture and Organization Laboratory (VCAOLab).
- Virtual Mathematic Laboratory (VMathLab).

¹ <http://www.uoc.edu/>

In the engineering programmes at the UOC, 62 different VLABs have been implemented (as of Spring 2008), distributed as follow: 29 VPLabs, 10 VNLabs, 6 VMathLabs, 4 VOSLabs, 11 VDBLabs and 2 VCAOLabs.

III. GENERAL STRUCTURE FOR A VIRTUAL PROGRAMMING LABORATORY

The fundamental concepts of procedural programming are an essential part of *Software Engineering* and *Computer Engineering* so Virtual Universities require VPLabs in order to improve the learning process of the practical programming activities [10]. Fundamental concepts include data types, control structures, functions, arrays, design of algorithms, and the mechanics of running, testing, and debugging [11]. Furthermore, students must learn how to use at least one programming language.

This section presents the implementation of a VPLab following the integrated structure of VLab that has been used at the UOC since 1997. The following key resources are identified in the general structure of VPLabs:

- **Technological resources.**
 - A. Virtual communication environment (VCE)
 - B. Simulator (SIM)
 - C. Remote laboratory (REM)
 - D. Virtual machine (VM)
 - E. Automatic assessment tool (AAT)
- **Pedagogic and strategic resources.**
 - F. Learning methodology (MET)
 - G. Support documentation and other materials (DOC)
 - H. Evaluation (EVA)
- **Academic staff resources.**
 - I. Teacher (TCH)

The VLab presented in this paper is implemented through the UOC *Virtual Campus*. However, other Learning Management System (LMS), like *Moodle*, could be used to implement the VLab proposal.

An overview of these resources is presented below. These resources are accessible from the Virtual Classroom at UOC.

A. Virtual communication environment

The virtual communication environment is a resource that allows communication between the users of the VPLab: students and teachers. It should stimulate the participation of students in the VPLab and provide for better interaction with the teacher. Virtual communication environments can be implemented using shared mailboxes, public forums and a teacher's board from different applications, e.g.: *Blackboard*, *BSCW*, *WebCT*, *LMS Moodle Classroom* or *UOC Virtual Classroom*.

The virtual communication environment at the UOC has been implemented as a *Virtual Classroom* as a particular tab for VCE. This VCE space has the following main features:

- **Mailboxes:** a common area that can be accessed by students and teacher, similar to a traditional classroom. This area contains a teacher's board and a forum.

- **List of members:** VPLab students and the teacher are listed in this area (photo, names and e-mail addresses).
- **Monitoring tools:** this feature allows teachers to check students' connections.

B. Simulator

A simulator is a tool that imitates experiments, states or processes. One of the main features of simulators is their interactive capability. Interactive simulations are quickly gaining importance as a means to explore, comprehend and communicate complex ideas [12, 7]. Interactive simulators can be implemented using a combination of computers, high-resolution graphics, simulation programming languages, and an Internet connection. The purpose of the simulator in a VLab is to allow the students to understand different scenarios as part of their practical work that may be difficult to observe in a real situation.

Simulators can be used for different proposals and activities: for example, computer architecture [13], programming (e.g., *Jeliot 3* [14] is a program visualisation tool that is based on the automatic animation of Java programs), networking (e.g., the *BOSON* and the *SEMSIM* simulators) and operating system concepts [15].

At the UOC, the VPLab uses an interactive simulator in order to help understand the concepts of algorithms and to learn the basic programming assignments of design and implementation. The simulator is a *Java Applet* that runs on web browsers [16] and is accessible in the *Resources tab* in the *Virtual Classroom*. In the future, it will include statistical tools and data mining capabilities that will inform the teacher about the results of students' interactions with the simulator.

C. Remote laboratory

A remote laboratory consists of one or more computers and/or devices installed and set up to allow remote access. The remote laboratory provides students with a real environment to carry out practical work. It should be noted that a remote laboratory is different from a simulator in that the latter is not a real environment [7, 8].

In a VLE, remote laboratories may be the only possible solution to work with some devices since it cannot be assumed that students will have them at home. Traditionally, remote laboratories have been developed for computer network [17, 18] and operating system subjects. For instance, router configuration can be performed using a remote laboratory like *CISCO Netlab* [19].

The VPLab implemented at the UOC does not include a remote laboratory. The reason is twofold. Firstly, all programming activities can be carried out by students using their home. And secondly, one of the things programming students must be able to do is to carry out compiler installations.

D. Virtual machine

A virtual machine is a tool that lets users create separate environments, each of which emulates the hardware of a complete physical computer. Students and teachers can run multiple operating systems at once on a single physical computer and switch between them. Virtual machines can be found implemented in different software solutions. The most important ones are *VMWARE*, *Xen virtual machine monitor*, and *Microsoft Virtual PC*.

Practical activities often need software installation, software set-up, and hardware set-up which students may find especially difficult. Therefore, these tasks need to be simplified with the use of virtual machines. Furthermore, virtual machines allow for the standardisation of the initial work environment, allowing all students to start the practical activity with the same computer configuration [20, 21].

The virtual machine at the UOC is implemented via *DVD-Live*. Students are sent a DVD that runs automatically, without installation on a hard drive. Virtual machine is based on free software programs: the *GNU/Linux* operating system based on *Ubuntu* distribution with a *Dev-C++* compiler for the *C/C++* programming language. The DVD provides students with a full environment on their local computer in order to perform programming activities.

E. Automatic assessment tool

The automatic assessment tool enables interactive processes to take place between students and the automatic assessment system that allows students to learn about iteration processes. Besides, the automatic system reduces the mechanical work teachers are required to do, allowing them to concentrate on other, more beneficial aspects. To improve its interconnection with other systems, it is very important for any assessment system to follow the standards set out in the *Question and Test Interoperability Specification* group drawn up by *Global IMS Learning Consortium* and the *Sharable Content Object Reference Model (SCORM)* group.

The assessment process for programming assignments is a web-based grading system that carries out a validation of the complexity, typography and structure of the source code in the programming assignment, and checks that a predetermined benchmark [22] has been attained. One of the most representative examples of an automatic assessment tool for programming assignments is the *CourseMaster system* [23].

The automatic assessment tool for programming assignments used at UOC's VPLab is a system developed by the university [24]. It is one of the most important resources in virtual programming courses because it allows students to know in real time whether they have been successful in the programming activity. The value to students of receiving feedback by means of the automatic assessment tool is shown in Section IV.

Essentially, the automatic assessment process consists of the compilation and execution against test data and user-specified data (benchmark). There are two different kinds of benchmarks: public and private. In the public tests, the input and output test data are known to the students, while for private benchmarks, these data are not publicly available. However, the automatic assessment tool provides to the students a report with the results of both private and public tests.

The automatic assessment tool also provides the teacher with all test reports of the students.

F. Learning methodology

Several changes in learning methodology have been required, moving away from a teacher-lecture-centred environment to a student-centred one [25]. Practical activities require a specific learning methodology where

the student is the central element of this educational model. Student-centred models must give students enough freedom to take advantage of the support offered, to plan their learning process and to regulate their own working rate, guaranteeing a dynamic learning experience for each student.

At the UOC, the learning methodology applied in the VPLab is the general UOC methodology [26] adapted to carry out practical activities. It is based on constructivist learning theory adapted to VLEs where asynchronous communication is allowed in space and time between the students and the academic staff. Such methodology allows students to have maximum flexibility, adapting their studies to their own rhythm and circumstances at any particular time [27]. The main features are: the teacher facilitates and monitors the learning process of the students, students are encouraged to be responsible and autonomous, the practical activities are divided into different parts to allow for continuous evaluation, and communication between students themselves and between students and teachers improves. In short, the VPLab is student-centred.

G. Support documentation and other materials

Support documentation and other materials consist of all the information and software that will help students achieve the objectives of the practical activities. Within the VPLab, students are provided with the following materials in electronic format:

- *Software (program files)*: VPLabs at the UOC use *Dev-C++* compiler (a full-featured Integrated Development Environment for the *C/C++* programming language).
- *Software support*: installation instructions, manuals, FAQs, and setting guides.
- *Material*: theoretical materials and programming language guides. For instance, a manual for the *GNU Compiler Collection* and *Standard C* library manual.
- *Virtual library*: recommended bibliography, electronic publications, full text databases and dictionaries.
- *Complementary material*: examples of practical programming activities and exams, sometimes with their solutions.

It is important to note that the quality of the material included in the VPLab must be maintained. However, such quality should be measured in terms of how useful the material is to students. Ideally, such materials should be tailored to every VPLab by the teacher. A good selection and presentation of these resources improves the time efficiency of the learning process.

H. Evaluation

The evaluation is a pedagogical resource that allows the students to achieve their learning objectives. In a VLE, it is very important to offer students a flexible model of continuous evaluation, providing activities to be completed throughout the semester [1].

At the UOC, the system evaluation used in VPLab follows a continuous evaluation model [26]. Students have to perform five activities, four of which are optional and

TABLE I.
Q5-Q12: FREQUENCY RESULTS OF STUDENTS' PROFILE.

Id	Question description:	Answer	Frequency
Q1	What is your Programming subject?	Programming Fundamentals using C++ Object-Oriented Programming using JAVA	158 (55,63%) 126 (44,37%)
Q2	How many semesters have you been studying at UOC in a VLE?	1-2 semesters 3-6 semesters More than 6 semesters	130 (45,77%) 93 (32,75%) 61 (21,48%)
Q3	Did you study in VLE before?	No Yes	207 (72,89%) 77 (27,11%)
Q4	How many years have you been working in a Programming area?	Never 0-1 year 1-3 years More than 3 years	143 (50,36%) 28 (9,86%) 39 (13,73%) 74 (26,06%)

consist of short programming activities. There is also a mandatory activity that is more difficult. Each activity is evaluated based on the results detailed in the report provided by the automatic assessment tool introduced in Subsection B. Notice that a properly chosen private test in the automatic assessment tool ensures the correction of the programming assignment.

I. Teacher

“Virtual teachers” in a VLab are members of the academic staff who help students reach their individual objectives, offering each student personalised attention. VLab teachers need specific skills. Among the most important are technological skills related to the tools used in the laboratory [18]. Another aspect that impacts on the role of the teacher is the fact that the students are usually isolated in a VLE. They are at home alone. This isolation means that mentorship and guidance must be reinforced to guide, motivate, plan and be more proactive.

On the other hand, VLabs need a team of teachers composed of at least two members of staff, each with different skill sets: one of them will need in-depth knowledge of the content related to the subject (the “Theory Teacher”), while the other will need more technical skills (“Lab Teacher”).

In the specific case of the VPLab at the UOC, the “Teacher Lab” is an expert in *DevC++* programming language, who prepared the installation instructions and the *C++* compiler FAQs. The “Theory Teacher” prepares the practical activities and the automatic assessments, and also resolves any questions related to programming contents in the virtual communication area.

IV. SURVEY: IMPACT OF THE VIRTUAL PROGRAMMING LABORATORIES

In order to evaluate the relevance of the proposed structure, a survey addressed to students that were enrolled in the main VPLabs was carried out in Autumn 2007. This section presents the objectives, design, results and analysis of the survey. The objectives of the survey were the following:

- To obtain feedback from the students concerning the eight resources of a VPLab presented in the previous section.
- To evaluate the robustness of the general structure of VPLab in a Virtual University.

The web questionnaire was available for eight days to 539 students (of whom 284 participated) through a link to the Virtual Classroom. The survey was anonymous and voluntary, and respondents were restricted to two subjects that use the VPLab: *Programming Fundamentals using C++* and *Object-Oriented Programming using JAVA*.

The web questionnaire was composed of twelve questions (Q1-Q12) divided into two parts: *Student profile* part and *VPLab's relevance resources* part.

TABLE II.
Q5-Q12: RESOURCES RELATED TO EACH QUESTION.

Id	ResourceId	Resource description
Q5	VCE	Virtual Communication Environment
Q6	AAT	Automatic Assessment Tool
Q7	SIM	Simulators
Q8	VM	Virtual Machines
Q9	MET	Learning Methodology
Q10	DOC	Support Documentation and Materials
Q11	EVA	Evaluation
Q12	TCH	Teacher Lab

TABLE III.
Q5-Q12: DESCRIPTIVE AND FREQUENCY ANALYSIS OF VPLAB RESOURCES.

	N	1	2	3	4	5	4-5	Mean	Std. Dev.	Variance
VCE	282	2	13	48	79	140	77,7%	4,213	0,934	0,873
AAT	263	6	28	44	85	100	70,3%	3,932	1,086	1,179
SIM	254	10	20	52	119	53	67,7%	3,728	1,006	1,013
VM	242	9	20	51	108	54	66,9%	3,736	1,017	1,034
MET	277	10	12	36	104	115	79,1%	4,090	1,019	1,039
DOC	280	14	26	28	96	116	75,7%	3,979	1,158	1,340
EVA	279	2	10	43	99	125	80,3%	4,201	0,879	0,773
TCH	279	8	15	30	91	135	81,0%	4,183	1,017	1,035

In the first part, question 1 to question 4 (Q1-Q4) were about the profile of the respondents. The results are shown in Table 3.9. The respondents were highly heterogeneous. Combining the responses to both programming subjects, the results show that 54% of students had more than one year's experience with the *Virtual Campus*. The majority of the respondents (73%) had not studied in a VLE before and 50% had professional experience in the programming field.

To obtain information about the VPLab structure, the second part of the web questionnaire had eight questions (Q5-Q12). The resource related to each question is given in Table 1. Students were asked to score from 1 to 5 the relevance of each of the different resource of the VPLab, as follows:

- 1) It is not a necessary resource
- 2) It is not an important resource
- 3) It is an interesting resource
- 4) It is an important resource
- 5) It is an indispensable or vital resource

The number of respondents, the frequency of each value, the percentage of 4 or 5 value, the mean value, the standard deviation and the variance of the eight resources are summarised in Table II.

A first analysis of the survey results shows the high relevance of all the VPLab resources. Between 65% and 81% of students considered them to be important or indispensable resources. All the mean values were greater than 3.6.

Secondly, in the descriptive and frequency analysis of the survey, results show a significant difference between a numbers of resources that can be classified, in order of relevance, in three groups:

- **Human factor:** the majority of students (81%) considered the *Teacher's* function as very important (option 4 or 5) in a VLE. Similar results are found in the literature [23] where students tend to depend more on the help of the teacher.
- **Pedagogic factor:** the *Evaluation* and the *Learning Methodology* were rated highly by students. These two resources are very important in the learning process of the students. Also, the results show that *Support Documentation and Other Materials* are a little less important in the learning process. This result is logical because the documentation and other materials are sometimes complementary.
- **Technological factor:** there is a group of tools connected to the technological resources - *Virtual Communication Environment*, *Automatic Assessment Tool*, *Simulator* and the *Virtual Machine* - that also received a very high score.

It is important to note that the VCE is the highest technological resource graded. Although the VCE is classified as a technological resource, it is closely related to academic staff. In a VLE, the VCE is absolutely necessary to create a real feeling of university community and is the only way teachers can reach their students.

To sum up, in order to learn programming in a VLE it is essential to have a VPLab composed of at least these eight resources and for these to be correctly integrated. Although technological resources were rated highly by the

students, they appeared to place more importance on pedagogical and human factors.

V. CONCLUSIONS

Virtual Laboratories in Virtual Universities have special needs due to the specific characteristics of this learning environment. Experience based on more than 21,000 students in computer curricula during the last eleven years has enabled us to identify the most important resources that a VLab should contain.

The survey conducted in Autumn 2007 has allowed for a classification of the resources based on the relevance given by the students. Results show that all resources were rated with high values, which means that the proposed resources were appropriate for a VLab. Beside this, the survey results enable resources to be classified in three groups based on their relevance: human factors, pedagogical factors and technological factors. Such a classification shows that, although technological factors are "enabling tools" for a VLab, students grant more importance to human factors and pedagogical factors.

These results may conclude that even in VLabs, the teacher and their interaction with students through the communication environment is an essential resource to ensure the success of the learning process in a VLE.

The proposed structure for VLabs is a general approach that contains several resources but not all of them may apply to a specific VLab. An example of the application of the general structure to a Virtual Programming Laboratory has been described, where a selection of resources has been used.

REFERENCES

- [1] L. C. Kaczmarczyk. "Accreditation and student assessment in distance education: why we all need to pay attention," in *ITiCSE '01: Proceedings of the 6th annual conference on Innovation and technology in computer science*, pages 113-116, New York, NY, USA, 2001. ACM Press.
- [2] IEEE Computer Society/ACM Task Force on Computing Curricula (CC). "Model Curricula for Computing," *IEEE Computer Society Press and ACM Press*, 2005.
- [3] IEEE/ACM Joint Task Force on Computing Curricula. "Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering," *IEEE Computer Society Press and ACM Press*, December 2004.
- [4] IEEE/ACM Joint Task Force on Computing Curricula. "Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering," *IEEE Computer Society Press and ACM Press*, August 2004.
- [5] K. Chiu. "What are the benefits of a virtual laboratory for student learning?," in *HERDSA Annual International Conference*, pages 12-15, Melbourne, 1999.
- [6] L. J. Leitner and J. W. Cane. "A virtual laboratory environment for online IT education," in *SIGITE '05: Proceedings of the 6th conference on Information technology education*, pages 283-289, New York, NY, USA, 2005. ACM Press.
- [7] J. E. Corter and J. V. Nickerson and S. K. Esche and C. Chassapis. "Remote versus hands-on labs: a comparative study," in *FIE'04: 34th Annual Conference on Frontiers in Education*, 2:F1G:17-21, 2004.
- [8] D. Z. Deniz and A. Bulancak and G. Ozcan. "A novel approach to remote laboratories," in *FIE'03: 33rd Annual: Conference on Frontiers in Education*, pages T3E:8-12, 2003.
- [9] A. K. Noor and T. M. Wasfy. "Simulation of physical experiments in immersive virtual environments," *Engineering Computations: Int. J. for Computer-aided Engineering and Software*, 18(3-4):515-538, 2001. ([doi:10.1108/02644400110695062](https://doi.org/10.1108/02644400110695062))

- [10] M. Vallim and J. Farines and J. Cury. "Practicing engineering in a freshman introductory course," *IEEE J_EDL*, 49(1):74-79, 2006.
- [11] J. Cao and A. Chan and W. Cao and C. Yeung. "Virtual Programming Lab for Online Distance Learning," in *ICWL '02: Proceedings of the First International Conference on Advances in Web-Based Learning*, pages 216-227, London, UK, 2002. Springer-Verlag.
- [12] A. Repenning and J. Ioannidou. "Collaborative Use and Design of Interactive Simulations," in *CSCL Proceedings*, Stanford, CA, 1999.
- [13] G. S. Wolffe and W. Yurcik and H. Osborne and M. A. Holliday. "Teaching computer organization/architecture with limited resources using simulators," *SIGCSE Bull.*, 34(1):176--180, 2002. ([doi:10.1145/563517.563408](https://doi.org/10.1145/563517.563408))
- [14] R. B. Levy and M. Ben-Ari and P. A. Uronen. "The Jeliot 2000 program animation system," *Computer & Education*, 40(1):1--15, 2003. ([doi:10.1016/S0360-1315\(02\)00076-3](https://doi.org/10.1016/S0360-1315(02)00076-3))
- [15] L. P. Maia and A. C. Pacheco Jr. "A simulator supporting lectures on operating systems," in *FIE'03: 33th Annual Conference on Frontiers in Education*, pages F2C-13-17, Vol.2, 2003.
- [16] X. Marco-Galindo and J. Prieto-Blázquez. "Necesidades Especificas para la Docencia de Programacion en un Entorno Virtual," in *Actas de VIII Jornadas de Enseñanza Universitaria de Informática, Jenui 2002*, pages 5 -- 12, Cáceres, 2002.
- [17] L. Toderick and T. Mohammed and M. H. N. Tabrizi. "A Reservation and Equipment Management System for Secure Hands-on Remote Labs for Information Technology Students," in *FIE '05: 35th Annual Conference on Frontiers in Education*, pages S3F:13-18, 2005.
- [18] D. C. Sicker and T. Lookabaugh and J. Santos and F. Barnes. "Assessing the Effectiveness of Remote Networking Laboratories," in *FIE, 2005: 35th Annual Conference on Frontiers in Education*, pages S3F-7; S3F-12, 2005.
- [19] J. Prieto-Blázquez and J. Arnedo-Moreno and J. Herrera-Joancomartí. "An Integrated Structure for a Virtual Networking Laboratory," in *IEEE Transactions on Industrial Electronics*, 55(6):2334-2342, 2008.
- [20] B. Kneale and A. Y. De Horta and I. Box. "Velnet: virtual environment for learning networking," in *ACE '04: Proceedings of the sixth conference on Australasian computing education*, pages 161--168, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.
- [21] E. Damiani and F. Frati and D. Rebecani and M. Anisetti, V. Bellandi and U. Raimondi. "The Open Source Virtual Lab: a Case Study," in *OSS 2006: Workshop on Free and Open Source Learning Environments and Tools FOSLET*, 2006.
- [22] K. Ala-Mutka and H. Järvinen. "Assessment Process for Programming Assignments," in *ICALT '04: Proceedings of the IEEE International Conference on Advanced Learning Technologies*, pages 297-301, 2004. IEEE Computer Society.
- [23] C Higgins and P Symeonidis and A Tsintsifas. "Diagram-based CBA using DATsys and CourseMaster," in *Proceedings of the International Conference on Computers in Education.*, 1:167-172, 3-6 Dec. 2002.
- [24] J. Prieto-Blázquez, X. Marco-Galindo and E. Serradell. "Requerimientos tecnológicos para la docencia de programación en un entorno de aprendizaje a distancia," in *Proceedings of the 4rt Ibero-American Conference in Systems, Cybernetics and Information technology*, pages 113-118, 2005.
- [25] T. Brush and J. Soye. "Implementation and evaluation of a student-centered learning unit: A case study," *Journal Educational Technology Research and Development*, 48(3):79-100, 2000. ([doi:10.1007/BF02319859](https://doi.org/10.1007/BF02319859))
- [26] Albert Sangra. "A New Learning Model for the Information and Knowledge Society: The case of the Universitat Oberta de Catalunya (UOC)," in *The International Review of Research in Open and Distance Learning*, Barcelona (Spain), 2002.
- [27] M. R. Simonson. *Teaching and Learning at a Distance*. Merrill Prentice Hall, Upper Saddle River, NJ, 2000.

AUTHORS

J. Prieto-Blázquez is with the Open University of Catalonia, Rambla de Poblenou 156, Barcelona, Spain (e-mail: jprieto@uoc.edu).

J. Herrera-Joancomartí, is Autonomous University of Barcelona, Campus de Bellaterra, Barcelona, Spain (e-mail: jherrera@deic.uab.cat).

A. Guerrero-Roldán is with the Open University of Catalonia, Rambla de Poblenou 156, Barcelona, Spain (e-mail: aguerreror@uoc.edu).

This work is partially supported by the Spanish Ministry of Science and Innovation and the FEDER funds under the grants TSI2007-65406-C03-03 E-AEGIS and CONSOLIDER CSD2007-00004 ARES.

This article was modified from a presentation at X International Symposium on Computers in Education (SIIE2008) 1st-3rd October 2008, Salamanca, Spain. Manuscript received 5 January 2009. Published as submitted by the authors.