Learning Design for Software Engineering Courses

Itana M. S. Gimenes[1], Leonor Barroca[2], Ellen F. Barbosa[3] and Edson A. Oliveira Júnior[1]
[1]*Departamento de Informática, Universidade Estadual de Maringá, Maringá-PR, Brazil*
[2]*Department of Computing, The Open University, Milton Keynes, U.K.*
[3]*ICMC, Universidade de São Paulo, São Carlos-SP, Brazil*

Keywords: Learning Design, Software Engineering, Course Design.

Abstract: This paper presents a customization of a learning design approach, OULDI, to designing and implementing Software Engineering courses. We propose an iterative process for the application of the OULDI views. This process starts with a course map view and follows a series of steps that ends with the evaluation of the design reflecting on the balance of the proposed activities. A case study is presented in which two institutions were involved in the design and implementation of an experimental software engineering course. Feedback from students, designers and lecturers was collected to support the validation of the design and implementation of the course. This showed that learning design, with the process proposed here, is a feasible approach for the design of software engineering courses.

1 INTRODUCTION

Information and Communication Technology (ICT) resources that have been used predominantly in distance education to improve the experience of the learner are now available at lower costs and finding their way into many educational contexts. This wide use calls for strategies to integrate ICT resources in the learning process that take into account different educational modes and different domains.

Education has always required planning and design; however, in a face-to-face context, learning relies often on implicit practice. The widespread use of ICT and the opening of new educational practices, for example, the integration of distance education elements and of open educational resources, make a stronger demand on support for preparation and planning. Learning design is an approach that supports teachers and designers to make informed decisions about course activities, resources, technologies and pedagogical approaches (Conole, 2013). Learning design can be used at different levels of granularity, from the representation of learning activities that are performed by different actors in the context of a course, to the planning of curriculum for whole programmes (Koper, 2006). When learning design is applied to develop a course, it allows for the sharing, discussion, validation and evolution of the course designs; when applied to activities it will facilitate the discussion of their learning outcomes and pedagogical approaches. Both the process of planning and the product of that planning can be made explicit through design representations supported by methods and tools.

In the design of a software engineering course, pedagogical decisions are influenced by the nature of professional activities. These require specific skills that can be strengthened by the activities, and the experiences that students engage with. The teaching of technical skills needs to be integrated with that of soft skills such as: cooperation and effective communication, leadership, negotiation, feasibility analysis, and adaptation to new models and technologies. Learning design can facilitate this integration through planning activities that promote the dialogue between learners and educators. Learning design has similarities with software engineering in terms of making abstractions and models explicit before implementation (Caeiro-rodríguez et al., 2010). The core of a software engineering course is about learning to extract requirements from stakeholders and the real world and making them explicit in a design language and in code (Sommerville, 2010). One such language, the Unified Modelling Language (UML) (OMG, 2013), has also been referred to as a design tool in learning design (Dalziel, 2012; Grainne Conole, 2013). Both software designers and educators rely

M. S. Gimenes I., Barroca L., F. Barbosa E. and A. Oliveira Júnior E..
Learning Design for Software Engineering Courses.
DOI: 10.5220/0004838002410249
In *Proceedings of the 6th International Conference on Computer Supported Education* (CSEDU-2014), pages 241-249
ISBN: 978-989-758-020-8

extensively on their prior experience and context for development (Wilson, 2007). This highlights commonalities between learning and software engineering design techniques that should be exploited further.

Several research projects developed tools to support learning design (Koper, 2006; Dalziel, 2012). The Open University Learning Design Initiative (OULDI) is such a project that developed a set of concepts together with computer-supported tools (Cross et al., 2012). It supports explicit course design representations and provides mechanisms to foster sharing of material and collaboration amongst course team members.

In this paper, we propose a customisation of OULDI for software engineering education. This customisation includes an explicit design process conceived to organise the development of the views proposed by OULDI. We applied this customisation to the design and implementation of an Experimental Software Engineering (ESE) course, in the context of a master program in Computer Science.

The paper is organized as follows. Section 2 gives the background for this work. Section 3 presents the customised OULDI process. Section 4 describes a case study in which an ESE course was designed and implemented by two institutions in Brazil with the collaboration of the Open University, UK. Section 5 discusses the feedback from designers, lecturers and students. Section 6 presents conclusions and further work.

2 BACKGROUND

Learning design as a research field has emerged in the last 10 years mainly from researchers in Europe and Australia (Koper 2006; Grainne Conole, 2013; Dalziel, 2012). It has a strong emphasis on making the design process and artefacts explicit and shareable. Design in education is not a new field though, and instructional design has been a well-established discipline for several decades (Eckel, 1993). However, learning design takes a broader approach, moving away from the production of instructions derived from learning goals, towards a more learner centred approach that is dynamic and takes into account a supporting environment and all stakeholders involved in planning the learning process; it builds also on research on learning sciences and design languages.

The learning design process and representation can be considered as pedagogically neutral as they can be used to represent the activities, tools and roles of any pedagogical approach. In this sense, learning design is more flexible than instructional design; it provides a framework where different pedagogical approaches can be implemented.

Our work is based on OULDI (Conole, 2013; Cross et al., 2012). It supports the design of courses with views, guidelines and tools. It allows the structured design of activities and their articulation with the learning outcomes, content and tools in such a way that the educators can envision the overall course to make decisions and carry out necessary adjustments before proceeding to production. It also provides a set of support tools, namely: CompendiumLD (CompendiumLD, 2008) which is a workflow design tool that contains special templates for course designs; and Cloudworks (Conole and Cuvel, 2009), that provides an open public space to which users can contribute, and where they can discuss learning and teaching designs and experiences. We chose to work with OULDI because of the set of support tools and its ease of use for higher education and for designers who are familiar with technology. Approaches, such as CADMOS (Katsamani and Retalis, 2008), LDSE (Laurillard et al., 2011) and LAMS (Dalziel, 2009) provide similar resources, but are more self-contained environments which would be difficult to customize. Their tools are also more directed to school teachers; our purpose is to support software engineering educators who are used to work with workflow techniques similar to the approach supported by CompendiumLD. We are aware that the OULDI has evolved and added more support mechanisms like the course features cards (Cross et al., 2012) but we did not incorporated them at this stage.

3 LEARNING DESIGN IN SOFTWARE ENGINEERING

OULDI (Cross et al., 2012) provides a set of shareable artefacts of design that represents a course around five conceptual views. These views are: (i) a `course map` which represents an overview of the course; (ii) a `course dimension`, which gives detail on the nature of the course (collaboration, assessment, user content, etc); (iii) a `pedagogy profile` which indicates the learners' participation in the designed types of activities; (iv) the `learning outcomes map` which links these to activities and assessment; and (v) the `task swimlane` which relates tasks to resources and tools. OULDI promotes an iterative approach of problem identification, solutions development, use, evaluation and refinement, but does not define an explicit design process. With our software engineering background and expertise we felt the need for a more detailed process to the application of OULDI to the design of courses in the software engineering domain. We detailed a process where the OULDI views are iteratively developed in three phases, as shown in Figure 1, and described below.

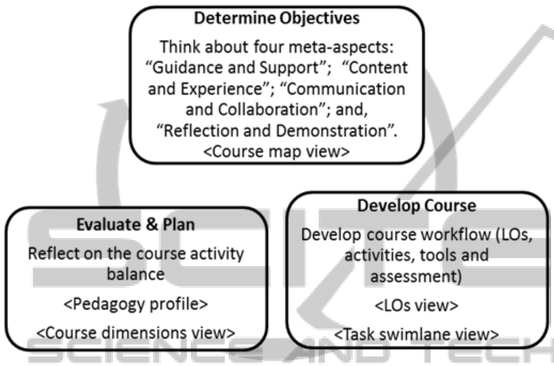

Figure 1: OULDI adapted process.

`Determine Course Objectives`: produces the course map view which is the first overview of the course. This helps educators think about the design of the course around four aspects: (i) Guidance & Support; (ii) Content & Experience; (iii) Communication & Collaboration; and, (iv) Reflection & Demonstration. Inputs to this phase are the course context and the ideas and objectives discussed by the course team.

`Develop Course`: develops the sequence of course activities generating the Learning Outcomes (`LOs`) and task swimlane view. Activities can be developed at hierarchical levels starting from composed activities and going down to atomic tasks. Activities and tasks can be associated with roles, tools and course material.

`Evaluate & Plan`: reflects on the balance of the activity types of the course to evaluate the design and evolve it. The designers have to establish the amount of each type of activity and the amount of assessment. This information produces the pedagogy profile and course dimension views. The pedagogy profile classifies the activities into: (i) Assimilative, attending and understanding content; (ii) Finding and handling information, gathering and classifying resources or manipulating data; (iii) Interactive and adaptive, using modeling or simulation software; (iv) Communicative**,** carrying out dialogic activities (e.g., group-based discussions); (v) Productive, constructing an experimental study; (vi) Experiential, practicing skills in the context of an experiment; and, (vii) Assessment, performing formative and summative evaluations.

Each phase produces a set of outputs which can be used in the next phase and refined iteratively. The availability of these artefacts facilitates the process of collaborative design of the learning experience.

4 CASE STUDY: THE ESE COURSE

The design of the ESE course was carried out taking into account the context of a master's program in Computer Science and a Brazilian multi-institutional project, funded by PROCAD/CAPES (www.capes.gov.br), whose objectives included the offer of collaborative courses. In Brazil, master's programmes are research driven programs where students engage with the development of new ideas and the proposal of new approaches (Barroca and Gimenes, 2013).

ESE is a subarea of software engineering focusing on the evidence of validity of methods and tools (Kitchenham et al., 2002). It is an important topic to teach in postgraduate computer science (and software engineering) programs geared to research; students need to provide evidence of the feasibility of proposed new methods and tools.

4.1 Course Design

The design of the ESE course started with the **`Determine Course Objectives`** phase by understanding the course context and educators's intentions and constraints. An ESE course teaches principles and techniques for evaluation applied to software engineering. It should instigate students to discuss collectively the value and means of evaluating research methods, tools and experiments. Students need to learn a well-defined process ranging from the planning of an experiment to its packaging for replication (Wohlin, 2000). This process should be supported by statistical methods to guide data collection and analysis. The course has to make sure that the theoretical principles are well understood, and that there are opportunities for learning and practising the development and replication of practical studies. Group work should be encouraged and supported. It is important to learn that the participation of individuals with an appropriate profile in the experiments is valuable to enhance the meaning of the collected data, thus improving confidence in the results. It is often the case that experiments involve more undergraduate and graduate students than practitioners. Therefore, the course should seek to involve external participants, mainly from industry, in the execution of the experiments. Students should be aware of existing tools to select and use in their experimental studies. The students should learn how to package their experiments for replication. As a result of the phase **`Determine Course Objectives`** the ESE Course map view was produced as described in Table 1.

Table 1: ESE Course map view.

Guidance & Support	Content & Experience
15 weeks (August to December 2012), 30 hours, 2 credits 2 hours of classes plus 4 hours of study per week which leads to a total of 90 hours Theory but more practice Course team and tutors Google (Website, Calendar, Wikis, Document sharing)	Existing text books: Wohlin , Juristo and Moreno Selected papers and theses Study guides Activities in the study guides Data collection and analysis
Reflection & Demonstration	**Communication & Collaboration**
Brainstorming of research evaluation Essays Small exercises Replication of an experiment Design and conduction of an experiment with result discussion Packing of an experiment Learning journal Self-assessment questions Formative and summative assessments Final examination	News Alerts for meetings and deadlines Work in groups, peer-to-peer and group discussions, log of discussions Google+ to support networks of experimental studies Network of experimental studies involving external participations from industry and postgraduate programs

The **`Develop Course`** phase designed the course activities taking as input the ESE course map view. It produced the `LOs` view of the ESE course in hierarchical levels as shown in Figures 2 and 3.

The 1st level of the `LOs` view consists of three activities: `Main Activities`; `Discussions`; and `Keep a Network of Participants`. Each activity is associated with roles, resources and tools which are represented in Figure 2 with the respective CompendiumLD icons. The activity `Discussions` is designed to aggregate students into groups to exchange ideas and carry out course assessments. The activity `Keep a Network of Participants` is designed to maintain a network of people who can act as participants in the course experiments.

The `Main activities` were further detailed, in a 2nd level, as shown in Figure 3. It shows lower granularity activities which compose the core of the course. In this figure we can see four CompendiumLD stencils: (i) `What is to be learnt` marking the line of `LOs`; (ii) `Student activity` for course activities; (iii) `Media and tools` for tools used in the activities; and (iv) `Learning output` for Summative Assessments (`SA`) produced by the activities. The core activities are: `Brainstorm the evaluation of software techniques and tools`; `Study concepts and principles`; `Replicate an experiment`; and, `Develop an experiment`. These activities were decomposed to atomic tasks. As an example, Figure 4 shows the task swimlane view of the `Develop an Experiment` activity from Figure 3. It contains atomic tasks associated with their respective Formative Assessments (`FA`).

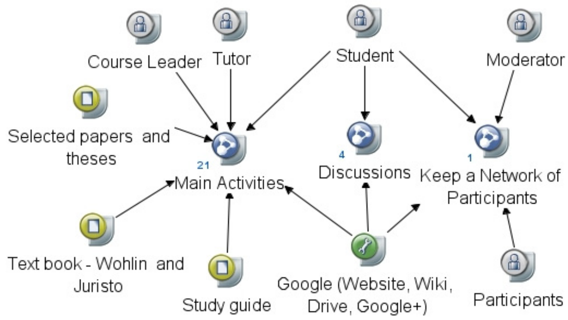

Figure 2: `LOs` view of the ESE course – 1st level.

The `Evaluate & Plan` phase has iteratively produced several versions of the pedagogy profile which were used to adjust and evolve the course design until the course team was satisfied with the distribution of activities. The fact that the course design is explicit and shareable allows designers to discuss and propose improvements, and facilitates the iterative refinement process of collaborative design.

The final balance of activity types was represented in a graph with: 10%-Assimilative; 3%-Finding and handling information; 45%-Communicative; 35%-Productive; 0%-Experiential; 7%-Assessment. There are no Experiential activities because the students were not supposed to participate in didactic experiments; Assessment activities have a low contribution as the course team counted the deliverables under Productive activities; the contribution of Communicative tasks is high as the course was designed to stimulate interaction between the institutions and the work groups.

The course dimension view, as it is a crosscut view, does not add extra information and it is not used here.

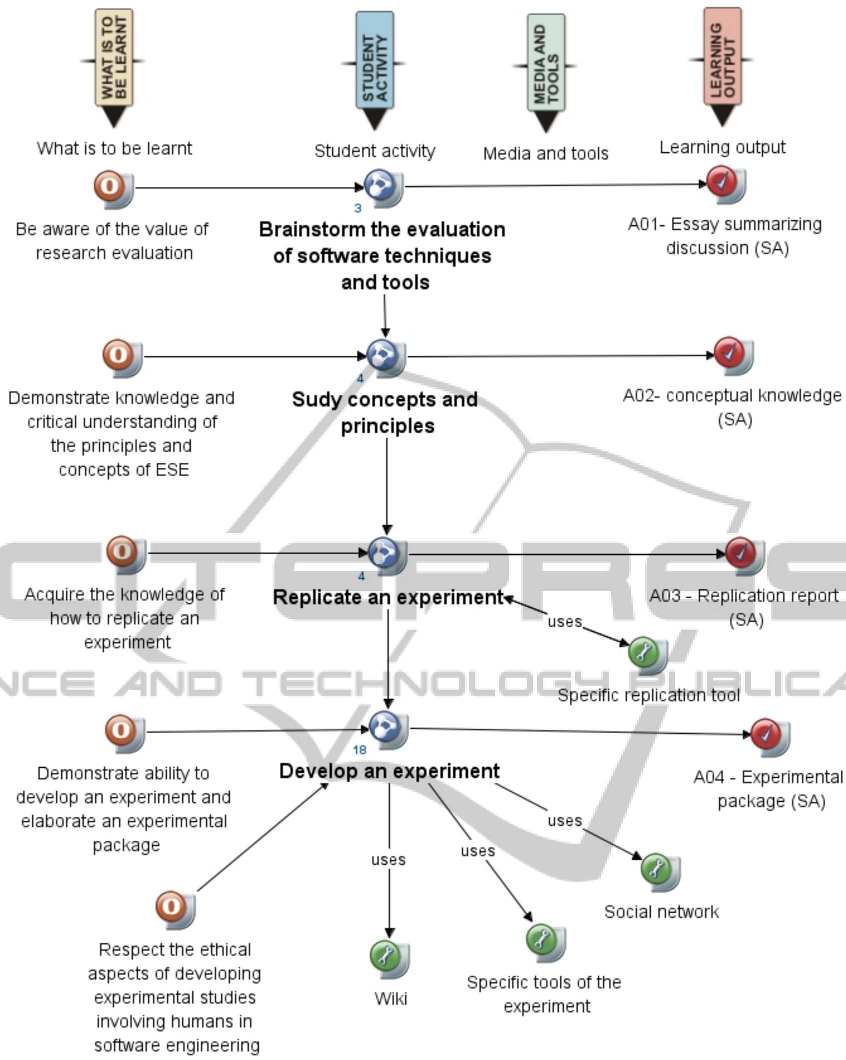

Figure 3: `LOs` view of the ESE course – 2nd level.

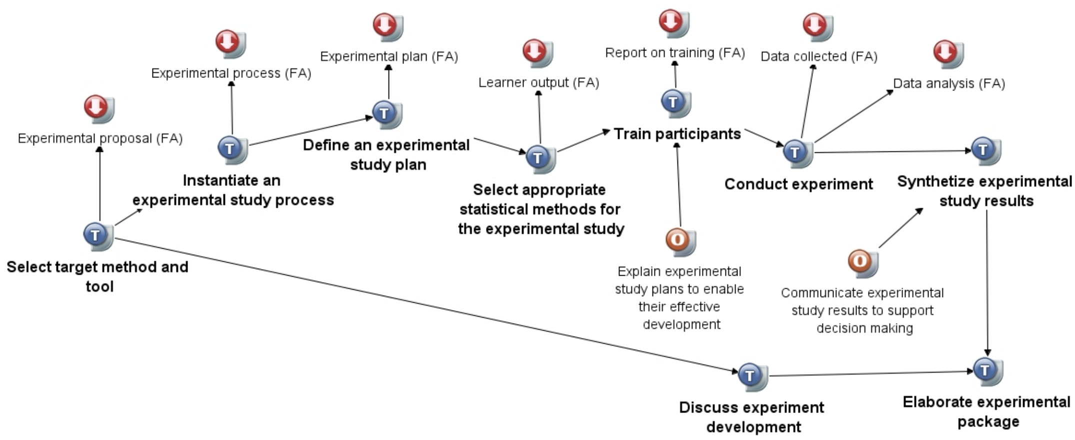

Figure 4: Task swimlane for `Develop an experiment`.

4.2 Course Implementation

The ESE course was implemented within the master's program in Computer Science of the Universidade Estadual de Maringá (DIN/UEM, http://www.din.uem.br) and the Universidade de São Paulo (ICMC/USP, http://www.icmc.usp.br). Delivering a multi-institutional course within research degree programs in Brazil is an innovative initiative which fosters collaboration between institutions that provide these programs. The use of OULDI was important to facilitate and support the collaboration between the institutions both in the planning and in the implementation of the course. The process proposed here, also facilitated this joint development.

Twenty-four students were enrolled in the course, nine from DIN/UEM and fifteen from ICMC/USP. The third and fourth authors were the lecturers in ICMC/USP and DIN/UEM, respectively. All authors participated actively in the course design, following the proposed phases of the OULDI customisation.

The implementation of the course followed the Guidance & Support designed in the course map view (Table 1). There were difficulties in the implementation of Google as the support learning environment. Moodle was chosen, as Google Apps for Education did not have a front end to encapsulate its tools. Also, students and institutions were more familiar with Moodle.

The course was implemented in a blended instruction mode. It was scheduled with the same timetable in both institutions and used the same Moodle site. The support materials were selected collaboratively and made available to students. Classes were intended to be alternated between the institutions and transmitted through video-conference. However, there were technical problems and, in the end, the same material was used but the classes were delivered locally by the lecturers in each institution. Students in each institution teamed up in groups of three. The final presentations of the experiments were successfully transmitted by video-conference and students and lecturers could interact to discuss the projects.

There were no changes to the designed `LOs` and task swimlane (Figures 2, 3 and 4), but one activity, `Replication`, was carried out differently, in each institution, due to individual decision of the local lecturers. At DIN/UEM, each team planned and replicated the same experiment, whereas at ICMC/USP all the students were participants in the replication of one experiment conducted by a researcher with the lecturer's support.

In the end, the `Keep a Network of Participants` was not implemented due to the time constraints of the course and the lack of involvement of the external community. We think this is an activity that should have been planned and developed by the institutions before the course started.

5 DISCUSSION

Results of the ESE course design presented in this paper are discussed from the perspectives of both designers/lecturers and students. One questionnaire was given to students and another to the designers/lecturers; both contained 12 questions regarding course design and implementation, content and structure, didactic and technical resources, learning environment, communication and collaboration issues, among others, as shown in the Appendix.

In summary, designers and lecturers made the following remarks:

- Course design and implementation: They agreed that the OULDI process was interesting and effective both in designing and evolving the course. It provided the course with representations that were used to share ideas within the design team as well as to guide the development of the learning environment for the course. In particular, the course map view helped to think in advance about the course goals and the main structure. Also, the Evaluation & Plan phase was crucial in planning better the course activities. The class schedule was followed accordingly and only small adjustments had to be made, mainly due to technical problems.
- Communication and collaboration: Skype meetings supported communication and collaboration effectively amongst designers/lecturers. However, a more integrated environment both for designing and collaboration is necessary.
- Virtual learning environment: They agreed that Moodle provided a complete set of resources to be used during the course which were easy to use.
- Technical issues: Contrary to what is published that technical resources for distance/blended learning are widely available at low cost, this is still not a reality in many places; both institutions involved in the ESE course had to set up the environments for video-conference. In particular,

DIN/UEM struggled to set up this environment which could only operate satisfactorily by the end of the course.

The students made the following observations:

- Course structure and content: 68% agreed that the content was adequate and well organized in the learning environment.
- Pedagogical resources: the wiki was the most used resource (59%), followed by the calendar (54%), the forum (50%) and email (36%), showing, therefore, a balance in the use of pedagogical resources.
- Communication and collaboration: Despite the results in the usage of pedagogical resources, there were problems regarding the effectiveness of communication and collaboration. At the start, there was some resistance in using the resources available. In particular, it was difficult to engage students in forum discussions. Many activities involving communication and collaboration were proposed as an attempt to encourage them to take part in the discussions. As a result, interesting questions and discussions were gradually arising in the forum, in the wiki, and by email. In the end, 46% of students considered the use of didactic resources effective, 32% were neutral and 22% thought they were of little or no effect.
- Technical resources: As discussed before, the use of technical resources, especially the video-conference environment was the most criticised aspect of the course; 50% of the students evaluated this item as “bad” or “very bad”.
- Motivation, autonomy and self-organization: 82% of the students considered themselves as motivated or highly motivated during the course. Also, 68% of students considered they were autonomous and self-organized to study the theory and perform the suggested practical activities.
- Interaction between students: 95% of students considered the interaction amongst students in the same institution, as “very high”. This is mainly because most students also acted as “participants” of the experiments designed by the other teams. Students organised themselves in this way. However, when considering interaction amongst students of different institutions, 77% of them evaluated it as “very poor”. Only in the final presentations of the experiments, were students of ICMC/USP and DIN/UEM able to interact and discuss the results of their experiments.
- Open comments: Students showed a positive attitude towards the ESE course, especially regarding the pedagogical approach used. The negative aspect pointed out by almost all of them was the duration of the course. Instead of two hours/week, they suggested at least three hours/week for designing and conducting the experiment.

6 CONCLUSIONS AND FURTHER WORK

This paper proposed and applied a customisation of a learning design approach, to a specific domain, that of software engineering. Overall, the learning design approach, and in particular OULDI, proved to be effective to design software engineering courses. In addition, it proved to be efficient in the support of the collaboration between Brazilian institutions in the ESE course design and implementation.

Although there are other learning design approaches in the literature (Koper, 2006; Dalziel, 2012), we are not aware of work being done of their customisation to software engineering. The specificity of the need for professional engagement, the knowledge and experience of design, the familiarity with workflow techniques and tools, and the engagement with the open movement make software engineering education an area that calls for such customisation.

The population of the case study was small, but of a typical size for postgraduate courses. We are aware of the threats to validity regarding the need of its application to a larger and more independent group of designers and students. We intend to evolve the ESE course and the OULDI process, with different groups of designers and students.

This paper carried out an experiment with OULDI and detailed comparison of the same experiment with other methods was out of scope. Further work is needed to comparatively assess and customise other LD methods to SE.

Further work also includes the design of a support environment for the collaborative design of software engineering courses; including mechanisms to access and produce open educational resources.

ACKNOWLEDGEMENTS

We are grateful to CAPES for the funding of Itana Gimenes postdoctoral research at the Open University, UK and the project PROCAD 191/2007.

REFERENCES

Barroca, L., and I. M. S. Gimenes. 2013. "Computing Postgraduate Programmes in the UK and Brazil: Learning from Experience in Distance Education with Web 2.0 Support." In *Cases on Web 2.0 in Developing Countries: Studies on Implementation, Application, and Use*, edited by N. Azab, 1st ed., 147–171. Hershey: IGI Global. doi:10.4018/978-1-4666-2515-0.ch006. http://www.igi-global.com/chapter/computing-postgraduate-programmes-brazil/73057.

Caeiro-rodríguez, Manuel, Luis Anido-rifón, and Martín Llamas-nistal. 2010. "Challenges in Educational Modelling : Expressiveness of IMS Learning Design EMLs and IMS Learning Design." *Educational Technology & Society* 13 (4): 215–226.

CompendiumLD. 2008. "CompendiumLD Learning Design Software." *The Open University*. http://compendiumld.open.ac.uk/.

Conole, G., and J. Cuvel. 2009. "Cloudworks: Social Networking for Learning Design." *Australasian Journal of Educational Technology* 25 (5): 763–782. http://www.ascilite.org.au/ajet/ajet25/conole.html.

Conole, Grainne. 2013. *Designing for Learning in an Open World*. London: Springer London.

Cross, Simon, R. Galley, Andrew Brasher, and Martin Weller. 2012. "Final Project Report of the OULDI-JISC Project : Challenge and Change in Curriculum Design Process , Communities , Visualisation and Practice". Milton Keynes, UK. http://oro.open.ac.uk/34140/.

Dalziel, James. 2009. "Prospects for Learning Design Research and LAMS." *Teaching English with Technology – Special Issue on LAMS and Learning Design* 1 (9): i–iv.

———. 2012. "The Larnaca Declaration on Learning Design." http://www.larnacadeclaration.org/.

Eckel, Karl. 1993. *Instructional Language: Foundations of a Strict Science of Instruction*. Englewood Cliffs: Educational Technology Publications.

Katsamani, Mary, and Symeon Retalis. 2008. "CADMOS Learning Design Tool." http://cloudworks.ac.uk/cloud/view/5789/links#contribute.

Kitchenham, Barbara A, Shari Lawrence Pfleeger, Lesley M Pickard, Peter W Jones, David C Hoaglin, Khaled El Emam, and Jarrett Rosenberg. 2002. "Preliminary Guidelines for Empirical Research in Software Engineering." *IEEE TRANSACTIONS ON SOFTWARE ENGINEERING* 28 (8): 721–734.

Koper, Rob. 2006. "Current Research in Learning Design." *Educational Technology & Society* 9: 13–22.

Laurillard, D., P. Charlton, B. Craft, D. Dimakopoulos, D. Ljubojevic, G. Magoulas, E. Masterman, R. Pujadas, E.A. Whitley, and K. Whittlestone. 2011. "A Constructionist Learning Environment for Teachers to Model Learning Designs." *Journal of Computer Assisted Learning* (December 15): no–no. doi:10.1111/j.1365-2729.2011.00458.x. http://doi.wiley.com/10.1111/j.1365-2729.2011.00458.x.

OMG. 2013. "Unified Modeling Language." http://www.omg.org/spec/UML/.

Sommerville, Ian. 2010. *Software Engineering (9th Edition)*. Addison-Wesley.

Wilson, P. 2007. "Progress Report on Capturing eLearning Case Studies."

Wohlin, Claes. 2000. *Experimentation in Software Engineering: An Introduction*. Springer.

APPENDIX

Questionnaire for the lecturers of the ESE module	**Questionnaire for the students of the ESE module**
**All answers to questions have discrete alternatives ranging from 1 (very bad) to five (very good). 1. How do you assess the module's learning outcomes? 2. How do you assess the planning of the module? (Teaching plan, workload, modules' support materials, bibliography, media resources, implementation of the projects, etc.) 3. How do you assess the organisation of the resources available in the module's Virtual Learning Environment (VLE)? 4. How do you assess the communication between students and lecturers? (Consider also communication using the forum, by email, etc.) 5. Which means of communication were used? (Select one answer for each [Forum, Wiki, E-mail, Calendar]: 1 for least used, 4 for most used). 6. How do you assess the effectiveness of the use of resources available in the module's VLE? (Forum, wiki, email and calendar). 7. How do you assess the videoconferencing sessions that took place? 8. How do you assess the students' performance in the assessment? 9. How do you assess the students' independence and self-discipline during the study of this module? 10. How do you assess the communication amongst students in your institution doing this module? 11. How do you assess the communication amongst students of involved institutions (DIN/UEM and ICMC/USP) doing this module? 12. How do you assess the collaboration amongst all lecturers involved in this module? 13. Other comments (add whatever you consider important).	**All answers to questions have discrete alternatives ranging from 1 (very bad) to five (very good) 1. How do you assess the organisation of the resources available in the module's Virtual Learning Environment (VLE)? 2. How do you assess the communication between students and lecturers? (Consider also communication using the forum, by email, etc.) 3. Which means of communication were used? (Select one answer for each [Forum, Wiki, E-mail, Calendar]: 1 for least used, 4 for most used). 4. How do you assess the effectiveness of the use of resources available in the module's VLE? (Forum, wiki, email and calendar) 5. How do you assess the videoconferencing sessions that took place? 6. How do you assess your performance in the assessment? 7. How do you assess your independence and self-discipline during the study of this module? 8. How do you assess your motivation and learning? 9. How do you assess the communication amongst students in your institution doing this module? 10. How do you assess the communication amongst students of the involved institutions (DIN/UEM and ICMC/USP) doing this module? 11. How do you assess the skills and competence of your lecturers? 12. How do you assess the collaboration amongst all lecturers involved in this module? 13. Other comments (add whatever you consider important).