The 7th International Scientific Conference
eLearning and Software for Education
Bucharest, April 28-29, 2011

ON DESIGNING LEARNING OBJECTS FOR A SOFTWARE RELIABILITY ENGINEERING COURSE

Grigore ALBEANU
Faculty of Mathematics and Informatics, Spiru Haret University, 13 Ion Ghica Str., Bucharest, Romania
galbeanu@gmail.com

Florin POPENTIU-VLADICESCU
UNESCO Department, University of Oradea, 1 Universitatii Str., Oradea, Romania
popentiu@imm.dtu.dk

***Abstract**: A software reliability engineering course for higher education asks for various resources based on recent developments in virtual learning. Establishing the required learning objects, methods of development and methods of integration with existing educational materials are the main objectives of the project described in this paper.*

Keywords:** courseware, e-learning, multimedia, learning objects, software reliability**.

I. INTRODUCTION

The design of learning objects by instructional designers asks for an adequate methodology in order to assure flexibility, adaptability, interoperability, and current availability requirements generated by the increasing request of the actors playing their roles into the framework of the present and future knowledge society.

Various Learning Content Management Systems (LCMSs) promote layouts, templates, and ways of building and composing new pieces of learning material updated according to the new developments into the field under training. Any LCMS should have the following functionalities:

1. content management;
2. provides tools for content creation;
3. support assessment creation, evaluation and feedback;
4. permits access to a searchable library of learning objects;
5. provides tools for collaboration/synchronous learning and is able to provide the learning object requested by the learner using specific protocols.

The LMCS could be a part of more complex software called Learning Management System (LMS). However, some LMS can incorporate only a Content Management System (CMS). As described by Irlbeck&Mowat (2005), a major difference between a CMS and a LCMS is that "the former is a horizontal software application and the latter is a vertical market software application". Various aspects concerning the differences between LMCS, LMS and CMS can be found in the mentioned reference.

This paper addresses the designing aspects of learning objects adequate to a software reliability engineering course to be accessed by learners located in different geographical places. The material is structured in the following way. The next section outlines the most important characteristics/features of learning objects. Software reliability engineering objectives are covered in the third section, where classes of assets will be identified. The learning objects and corresponding assets will be used into the framework of an International Master program driven both by Moodle and Blackboard Learning System.

II. DESIGNING LEARNING OBJECTS

As considered by Albeanu & Duda (2011), "the usage of Information and Communication Technologies (ICT) during training, teaching, self-learning or balanced instruction is compulsory nowadays in order to achieve a better understanding of the subject in short time". ICT based learning objects belong to the so called educational software. During modern lectures/lessons various types of educational software can be used: courseware, classroom aids, educational games, etc.

A courseware or an educational package contains not only lessons and tutorials, but also tests, and additional pieces of software, like references (dictionaries or encyclopedias), visualization or simulation applets, macros, games etc. If pedagogical aspects are considered then the courseware or pieces of it represents learning objects.

That software useful to trainers or teachers both for classroom management and assistance (simultaneously visualization on learners' display) belongs also to the education software family. Also educational games can be thought as additional piece of software to help the understanding of experiments, the efficiency of some scenario, and the adequate usage of some interface (hardware or software), including virtual reality interfaces. According to this acceptance the educational games can be considered as learning objects.

To conclude, *a learning object is a set of resources, viewed as independent and reusable entities, useful to create various educational pieces suitable to a pedagogical hierarchy*. According to CISCO (2003), a reusable learning object strategy "provides a five-level hierarchy that authors can use to ensure consistent structures across multiple courses: course, module, lesson, topic, and sub-topic."

From technical point of view learning objects have metadata and content. Not only technical attributes are important but also such attributes addressing the learning objects' quality. According to Morales et al (2009), the following classes of aspects are also important: psycho-pedagogical (motivation and attention capability, adaptation to the audience, professional competency, accessibility, presentation design, level of difficulty), curricular-teaching (objectives achievement, content validity, adaptability to learning strategies, the adequacy of activities, feedback), functional (interface design, interaction usability and creativity by promoting self learning, cognitive domain development), and technical (standards compliance including interoperability issues). During interface design the following types of basic pieces will be considered: text, images, animations, audio, video, multimedia. In this way, multimedia materials are a combination of text, audio, images, animation, and video supporting interactivity. Finally, the multimedia quality depends also on the platform (hardware resolution) and communication channels (quality of service).

In order to develop/deliver high quality learning objects an appropriate methodology taking into account steps like: analysis, design, development, implementation, and evaluation, should be used.

The design will address instructional strategies, visual and technical aspects, and prototype evaluation. The learner profile plays an important role for the instructional design strategy. After establishing the objectives, the content will be structured logically and covering a subject from simple to complex, from known to unknown, starting to concrete things and continue with abstractisation. A content descriptor will be obtained. Two strategically levels can be considered: macro, and micro. By mixing of audios, graphics, animations and simulation, an interesting class of learning objects is obtained. These objects bring more understanding of subject and the learners are more interested and motivated. The learner evaluation is based on various assessment techniques (practice, project, essay, games & quizzes, pre-and-post tests, etc.) depending on the declared learning objectives.

The following processes are considered during visual design phase:

1. Designing the template of the Graphical User Interface;
2. Establishing the visual appearance (fonts, sizes, layouts, etc.).

The interface should be intuitive. The fonts should be legible, and the layouts should reflect the material organization.

The most important technical aspect is related to interoperability. Other aspects deal with navigation, file sizes/resolutions, and customization. Finally, the obtained prototype should be tested according to the learners' profile to confirm (full, partial, none) the suitability of the design methodology.

In order to deploy learning objects to be used for many platforms, the designers should provide conformance to standards, like SCORM.

Bellow, it is presented a logical structure of a learning object embedded in a partial design in order to illustrate the composition of the generic learning objects.

Learning object ID:	1
Object Title:	Fundamentals
Course ID:	ENG101
Language (Romanian, Hungarian, English):	E
Abstract (maximum 50 words):	This object deals with fundamental definitions and concepts of reliability
Keywords (maximum 4 items):	Reliability, System Mean Time to Failure, Failure Rate Function, Maintainability, Availability
Asset ID:	*1.1*
Asset type (**C**ontent, **P**ractice, **A**ssessment):	C
Content ID:	1.1.1
Content Lifecycle (Version, Status):	(1, Development)
Content type (Presentation, Glossary):	P
Content media (Text, Web-page, Audio, Video):	T
Content container (filename):	1.1.1(Definitions).pdf
Content ID:	1.1.2
Content Lifecycle (Version, Status):	(1, Development)
Content type (Presentation, Glossary):	P
Content media (Text, Web-page, Audio, Video):	V
Content container (filename):	1.1.2(Definitions).flv
Content ID:	1.1.3
Content Lifecycle (Version, Status):	(1, Development)
Content type (Presentation, Glossary):	G
Content media (Text, Web-page, Audio, Video):	W
Content container (filename):	1.1.3(Definitions).html
Asset ID:	*1.2*
Asset type (**C**ontent, **P**ractice, **A**ssessment):	P
Content ID:	1.2.1
Content Lifecycle (Version, Status):	(1, Development)
Content type (Presentation, Glossary):	P
Content media (Text, Web-page, Audio, Video):	T
Content container (filename):	1.2.1(Practice).pdf
Content ID:	1.2.2
Content Lifecycle (Version, Status):	(1, Development)
Content type (Presentation, Glossary):	P
Content media (Text, Web-page, Audio, Video):	A
Content container (filename):	1.2.2(Practice).mp3
Asset ID:	*1.3*
Asset type (**C**ontent, **P**ractice, **A**ssessment):	A
Content ID:	1.3.1
Content Lifecycle (Version, Status):	(1, Development)
Content type (**Q**uiz, **P**roject, **E**ssay):	Q
Content media (Text, Web-page, Audio, Video):	T
Content container (filename):	1.3.1(Quiz).pdf
Content ID:	1.3.2
Content Lifecycle (Version, Status):	(1, Development)
Content type (**Q**uiz, **P**roject, **E**ssay):	P

Content media (Text, Web-page, Audio, Video):	V
Content container (filename):	1.3.2(Project).flv
Content ID:	1.3.3
Content Lifecycle (Version, Status):	(1, Development)
Content type (**Q**uiz, **P**roject, **E**ssay):	E
Content media (Text, Web-page, Audio, Video):	T
Content container (filename):	1.3.3(Essay).pdf
Copyright:	Popentiu & Albeanu, 2011
Prerequisites:	*Special Chapters of Mathematics for Engineers* or *Probabilities and Statistics*
Typical learning time:	1 hours
Other aspects:	Level: graduate, Difficulty: medium, Cost: Free for students, 30€ for Industry.

III. A SOFTWARE RELIABILITY ENGINEERING COURSEWARE

The software reliability engineering process is a complex one, not only due to the mathematical estimation procedures, but mainly due to the current software complexity. The software reliability engineer needs a deep understanding of the activities to be developed during software reliability management.

A course on software reliability engineering is difficult to build due to the existence of an increasing software development methodologies which ask for a different approach, and to the increased speed of technological inovation. It is clear the difference between hardware (calendar oriented evaluation) and software (not depending on the software age) reliability. The operational profile approach and the necessary reliability concept proposed by Musa (1999) are necessary to be included during the training of future software reliability engineers. A large plethora of models are available, some of them considering the imperfect debugging and the presence of environmental factors as in Pham (2000). Taking into consideration the software reliability increasing by design, recently new theoretical approaches were proposed for automatic software verification and validation, as Doron (2001) explained.

In order to design an updated software reliability engineering course a learning object methodology was selected. The advantage of this approach consists in the posibility of replacing some old assets with new assests developed recently and keeping the course up-to-date.

Using ICT facilitates the course development, the online participation of learners including for collaborative activities.

	Lesson / Topic	Learning Objects	Number of assets (type/media: **T**ext, **A**udio, **V**ideo, **W**ebpage, **Q**uiz, **P**roject, **E**ssay, etc)		
			Content	Practice	Assessment
1	Introduction	The need for software reliability	1(T), 1(V)	1(V)	1(Q)
		The software reliability engineering objectives	1(T)	-	1(Q)
		Basic definitions and terminology	1(T)	1(W)	1(Q)
2	Reliability Engineering Fundamentals	Reliability Fundamentals	1(T), 1(V), 1(W)	1(T), 1(A)	1(Q), 1(P), 1(E)
		Reliability Function for Common Distributions	5(T)	3(T)	2(E)

3	Software Engineering Assessment	Software lifecycle	1(T)	1(T)	1(Q)
		Developing Operational Profiles	1(T)	1(T)	1(Q)
		Software Failures and Failure Processes	1(T), 1(V)	1(T)	1(P)
		Factors that affect Software Reliability	1(T), 1(V)	1(T)	1(Q), 1(P)
4	Software Verification and Validation I	Preparing for test	1(T)	1(T)	1(Q)
		Executing Test	-	1(T)	1(P)
		Software Reliability Data Collection	1(T)	1(T)	1(P)
		Applying Failure Data to Guide Decisions	1(T), 1(V)	1(T)	1(Q), 1(P)
5	Software Verification and Validation II	Modelling Formalisms	3(T), 1(W)	1(W)	1(Q)
		Automatic Verification	3(T)	1(T)	1(P)
		Deductive Software Verification	3(T)	1(T)	1(E)
		Combining Testing and Model Checking	2(T), 1(V)	1(T)	1(Q)
6	Software Reliability Modelling I	Static Models	1(T)	1(T)	1(P)
		Markovian Models	4(T)	2(T)	1(E)
		Using Software Reliability Tools I	1(T), 1(V)	3(W)	1(P)
7	Software Reliability Modelling II	NHPP Software Reliability Models	4(T)	1(T)	1(P)
		Bayesian Software Reliability Models	2(T)	1(T)	1(E)
		Using Software Reliability Tools II	2(T), 2(V)	3(W)	1(P)
8	Software Reliability Modelling III	Fault-Tolerant Software	2(T)	1(T)	1(Q)
		Software Reliability Models with Environmental Factors	2(T), 1(V)	1(T)	1(P)
9	Optimal Software Reliability Allocation	Software Cost Models	2(T)	1(T)	1(Q)
		Software Reliability Optimization	2(T)	1(T)	1(P)
10	Management of Software Reliability	Implementing Software Reliability on a Particular Project	1(T), 2(V)	1(T)	1(P)
		Lessons Learned	1(T)	1(T)	1(E)

During training, various videos (public available or proprietary), and software reliability tools will be used: SMERFS (see Farr (1982)), CASRE (see Musa (1999)), SREPT (see Ramani et al. (1998)), NHPP (see Pham (2000)), and SPIN (formal checking). The learners have to use the software during projects or essays preparation.

IV. CONCLUSIONS

Designing of learning objects for a software reliability engineering courseware offered online to international students is an important task having as main objective the increasing of quality of service when online or distance learning programs are offered by a higher education unit.

The paper has described the most important aspects of designing learning objects addressing various parts of a course on software reliability engineering. The beneficiaries of this project will be

international students from Hungary and Romania (the first phase) and, France and Denmark (the second phase).

Acknowledgements

The authors acknowledge the support of their departments during the development of this project which is part of their internal research plans: UNESCO Department of the University of Oradea (the first author), The Research Center in Mathematics and Informatics of the *Spiru Haret* University (the second author).

References

[1] Albeanu, G. Duda, I.G., 2011. Intuitionistic Fuzzy Approaches for Quality Evaluation of Learning Objects, *Proceedings of ISSAT* (submitted).

[2] CISCO, 2003. *Reusable Learning Object Authoring Guidelines: How to Build Modules, Lessons, and Topics*, White Paper.

[3] Costa A.P., Loureiro M.J., Reis L.P., 2009. Development methodologies for educational software: the practical case of courseware SERe, *International Conference on Education and New Learning Technologies (EDULEARN09)*.

[4] Doron, P., 2001. Software Reliability Methods, Springer.

[5] Farr, W., 1982, SMERFS: http://www.slingcode.com/smerfs/

[6] Garzia, M., Hudepohl, J., Snipes, W., Lyu, M., Musa, J., Smidts, C., Williams, L., 2006. How Should Software Reliability Engineering (SRE) be Taught, *ACM SIGSOFT Software Engineering Notes*, 31(4), pp. 1-5.

[7] Irlbeck, S., Mowat, J., 2005. Learning Content Management System (LCMS), http://www.herridgegroup.com/pdfs/LCMS_chptr_Aug29%202005_%20Final%20Sept%206%2005.pdf

[8] Karanta, I., 2006. *Methods and problems of software reliability estimation*, VTT Technical Research Centre of Finland.

[9] Lyu, R.M., 2007. *Software Reliability Engineering: A Roadmap, Future of Software Engineering*, L. Briand and A. Wolf (eds.), IEEE-CS Press.

[10] Lyu, R.M., Schönwälder, J., 1998. WEB-CASRE: A Web-Based Tool for Software Reliability Modeling, *Proceedings Ninth International Symposium on Software Reliability Engineering (ISSRE'98) Industry Track*, Paderborn, Germany, November 4-7 1998, pp. 151-160.

[11] Mills, Sandy, 2002. *Learning about Learning Objects with Learning Objects*, http://www.alivetek.com/learningobjects/site_paper.htm

[12] Moore, M.G. (Ed.), 2007. *The Handbook of Distance Education*. Second Edition. Mahwah, N.J. Lawrence Erlbaum Associates.

[13] Morales Morgado E.M., Gómez Aguilar D.A., García Peñalvo F.J., Therón Sánchez R., 2009. Supporting the Quality of Learning Objects through their Ranking Visualization, *iJET – 4(Special Issue 1): "SIIE'2008"*, pp 24-29.

[14] Musa, J. 1998. *Software Reliability Engineering. More Reliable Software. Faster Development and Testing*, McGraw-Hill.

[15] Neufelder, A.M., 1993. *Ensuring Software Reliability*, Marcel Dekker Incorporated.

[16] Pham, H. 2000. *Software Reliability*, Springer.

[17] Ramani, S., Gokhale, S.S., Trivedi, K.S., 1998. SREPT: Software Reliability Estimation and Prediction Tool, LNCS 1469, Springer.

[18] Shenvi, A., 2009. Reliability: A Software Engineering Perspective, *Proceedings of ISSRE 2009*, http://www.issre2009.org/proceedings

[19] Holzmann, G. J., 2004. *The SPIN Model Checker: Primer and Reference Manual*, Addison-Wesley.

[20] Wang, W.-L., Scannell, D., 2005. An Architecture-Based Software Reliability Modeling Tool and Its Support for Teaching, *Proceedings of 35th ASEE/IEEE Frontiers in Education Conference*, Indianapolis, T4C: 15-20.

[21] ***, SCORM, http://www.adlnet.gov/Technologies/scorm/default.aspx