USING REENGINEERING TECHNIQUES FOR DISTANCE STUDY COURSE IMPROVEMENT

Lina Tankeleviciene

Abstract: The paper analyses the reengineering concept as it comes from software engineering and management fields. We combine two viewpoints and apply them to solve a problem of reengineering of a distance study system, in general, and the unit of learning, in particular. We propose a framework for reengineering of unit of learning, based on general model of software reengineering, and present a case study, in which we describe, how one topic of distance study course was reengineered, considering triple consistency principle and requirements for computer science. The proposed framework contributes to increasing quality, effectiveness and systematization of delivering distance studies.

Keywords: distance study course, reengineering, triple consistency principle.

ACM Classification Keywords: K.3.1 Computer Uses in Education

Conference: The paper is selected from Third International Conference "Modern (e-) Learning" MeL 2008, Varna, Bulgaria, June-July 2008

Introduction

The distance study systems still face problems of different types:

1) Study programmes are evolving.

2) Students need up-to-date subject domain and learning material.

3) Course management systems (CMS) are quite often updated.

4) Courses must be reallocated into other CMS.

Dealing with these problems requires much of the lecturer’s time. On the other hand, lecturers wish to have easy-to-use CMS and institutional support in the form of clear policy and division of responsibility.

Therefore, a systematic view is necessary towards reengineering of a distance study system. Systematic view towards design and reengineering also poses the ability to evaluate not only results, but processes too. Different kinds of standards and other regulating documents can be employed in mentioned processes.

Reengineering of materials is necessary before importing them into the e-learning system. In this context, we introduce the concept reengineering. Unfortunately, this step is often omitted, and the prepared material goes online, but sound e-Learning principles are not implemented.

The aim of our paper is to show how the concept and methodology of reengineering can be used in the e-Learning system, in general, and in deploying the units of learning (further - UL), in particular.

Reengineering Concept in Software Engineering and Management Fields

Reengineering concept with its different interpretations is used in software engineering and management sciences. Software reengineering is concerned with reimplementing a system in order to make it more maintainable [Sommerville, 2000]. In [Chikofsky, 1990], reengineering is defined as „the examination and alteration of a subject system to reconstitute it in a new form and the subsequent implementation of the new form“.

The activities in the software reengineering process are: a) Source code translation; b) Reverse engineering; c) Program structure improvement; d) Program modularization; e) Data reengineering [Sommerville, 2000]. They are not all necessary. It depends on from what level we want to renew the system.

The difference between engineering and reengineering is shown in Figure 1. In reengineering the old system acts as a specification for the new system.

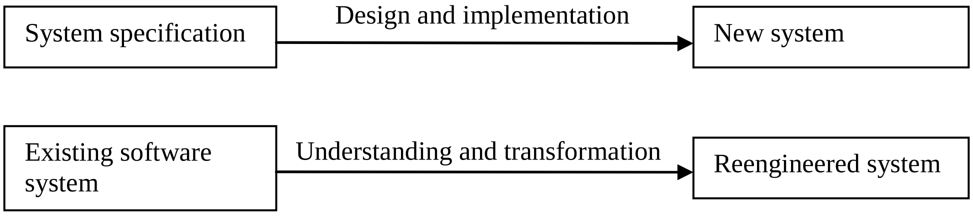


Figure 1. Forward engineering and reengineering (according to [Sommerville, 2000]).

The main advantages of reengineering are: a) Reduced risk; b) Reduced cost. The objective of system re-engineering is to improve the system structure and make it easier to understand. The cost of future system maintenance should therefore be reduced [Sommerville, 2000].

Almost at the same time the concept of “reengineering” was introduced in enterprise management area. Hammer [Hammer, 1994] defines reengineering as “the fundamental rethinking and radical redesign of business processes to bring about dramatic improvements in performance”. The objectives of reengineering from this point are: a) To increase competitiveness; b) To improve financial results; c) To promote teamwork.

Instructional design theories combine two mentioned viewpoints. Comprehensive analysis of reengineering of TEL (Technology Enhanced Learning) is presented in [Choquet&Corbière, 2006]. The framework proposed is based on Reference Model of Open Distributes Processing (RM-ODP, ISO/IEC 10746-1, 1998). The authors analyze 5 viewpoints on the design of system architecture:

- Enterprise viewpoint focuses on the purpose, scope and policies of that system.
- Information viewpoint focuses on the semantics of information and information processing.
- Computational viewpoint focuses on functional decomposition of the system into objects which interact at interfaces.
- Engineering viewpoint focuses on the mechanisms and functions required to support distributed interaction between objects in a system.
- Technology viewpoint focuses on the choice of technology in that system.

Reengineering is also often associated with business process reengineering. In the context of e-Leaning, it is common understanding, that organization must change itself in order to meet new challenges and requirements. Difficulties and problems of the implementation of e-Learning in universities of Lithuania are discussed in [Bulanova, 2006]. The authors also accentuate that situation is rather inertial, and educational organizations need more professional management.

Macro and Micro Levels for Reengineering in e-Learning System

Reengineering can be applied both to a complete system and to the individual components, simultaneously in the appropriate level of abstraction (ISO/IEC 10746-1, p. 12). Here we present the layers of e-Learning system and roles of the staff of organization in reengineering of e-Learning system.

The typical layers for e-Learning system from the lecturer perspective are presented in Figure 2.

Reengineering processes, based on these layers, can be conducted by different persons in organization. The situation at Siauliai University (typical organization for Higher Education) is depicted in Figure 3.

Further reengineering will consider only lower levels, which can be managed by academic staff.

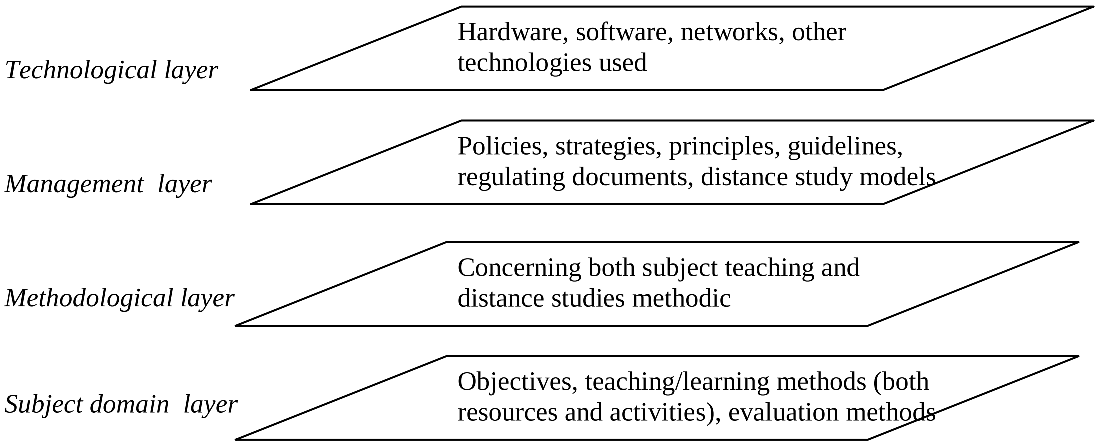

Technological layer
Hardware, software, networks, other technologies used
Management layer
Policies, strategies, principles, guidelines, regulating documents, distance study models
Methodological layer
Concerning both subject teaching and distance studies methodic
Subject domain layer
Objectives, teaching/learning methods (both resources and activities), evaluation methods

Figure 2. Layers of e-learning system from the lecturer perspective (as in [Gruslyte, 2007])

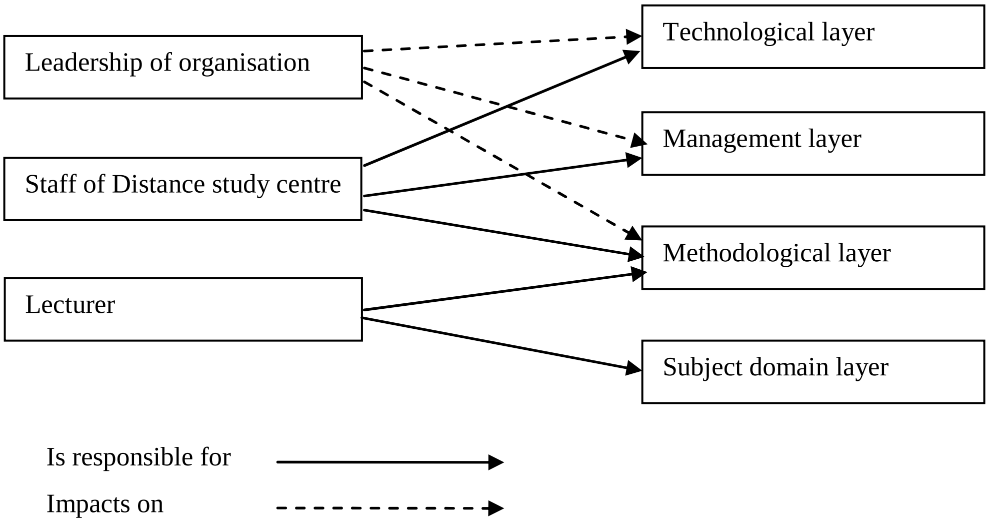


Figure 3. Roles of the staff of organisation in reengineering of e-Learning system

Framework for Reengineering of Units of Learning

Subject domain layer lets us concentrate on separate distance study course. The base, on which distance study course for formal studies is developed, consists of the following elements: a) Objectives; b) Teaching/learning methods (include both resources and activities); c) Evaluation methods. This decomposition is made on the higher abstraction level. On the technological level the course can be analyzed as the collection of learning objects and information objects, which are aggregated into lessons, topics, sections or other structural units. So, these objects must be reengineered.

Unit of learning (UL) is “an abstract term used to refer to any delimited piece of education or training, such as a course, a module, a lesson, etc.”, as defined in [IMS-LD, 2003]. From the lecturer perspective, there can be mainly differentiated two types of UL according to granularity: overall distance study course and topic (as used in MOODLE, also different terms can be used, e.g., lesson, module). Our proposed reengineering framework can be applied to both mentioned types.

Our proposed UL reengineering framework is shown in Figure 4. It is based on general model of software reengineering, proposed in [Byrne, 1992].

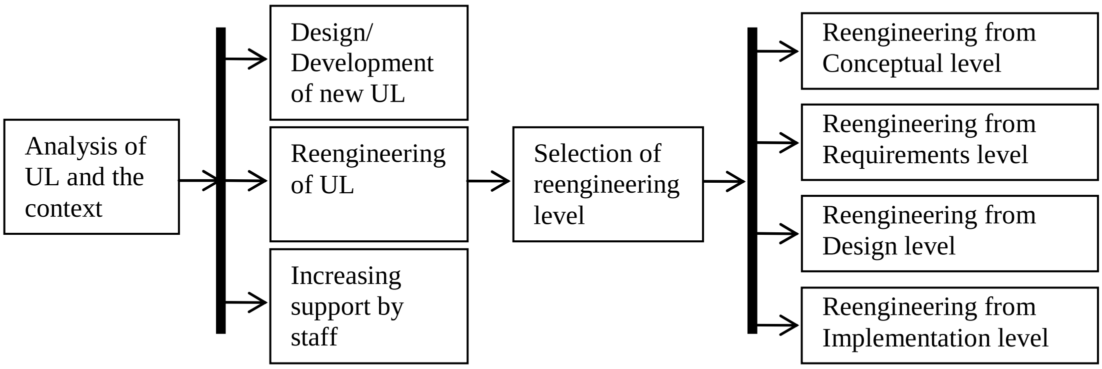

Analysis of UL and the context
Design/ Development of new UL
Reengineering of UL
Increasing support by staff
Selection of reengineering level
Reengineering from Conceptual level
Reengineering from Requirements level
Reengineering from Design level
Reengineering from Implementation level

Figure 4. UL Reengineering framework

The first step means making decision, if we need reengineering. In our practice we need to adapt distance study courses in different context: 1) For different modes of studies: full-time daytime, part-time evening and part-time extra-mural; 2) For different study programs, where the same module must correspond the different amount of student working hours; 3) For different study programs, where the different aspects of the same subject must be emphasized. Usually the copy of distance study course is made and further modified in order to meet new requirements. However, it requires a big amount of workload time each time when new teaching/learning session starts.

There are two other alternatives besides reengineering of UL: 1) Design/Development of new UL. This way must be chosen, when UL essentially differs from the previous instance. 2) Increasing support by staff. Distance studies are oriented towards and based on active participation of learners, who take control over learning results. However, extra contribution by academic staff to teaching/learning process can reduce negative impact of pure learning material. Therefore, this way is suitable for immediate and one-time delivery of distance study course.

The solution must be chosen carefully, because it influences further expenditures of finances, time, etc. Also it impact on further possibilities to reuse the UL.

The second important step is selection of reengineering level. We use the classification, proposed in [Byrne, 1992]. Levels, related types of change and adoption of mentioned for e-Learning domain are presented in Table 1.

Table 1. Reengineering levels

Reengineering level	Types of change	Deals with:
Conceptual	Re-think	The background for reengineering, e.g., new pedagogical strategy accepted.
Requirements	Re-specify	- Requirements of a module, or, as stated in Study Module Program (SMP), aims and provided knowledge, abilities and competencies. - Requirements of an organization.
Design	Re-design	- Learning activities; - Assessment types, or, as stated in SMP, evaluation procedure of knowledge and abilities; - Informational, navigational, presentational structure;

		- Pedagogical scenarios.
Implementation	Re-code	Learning resources, activities, scenarios.

Higher levels influence lower levels, and, for e.g., if we have raised new requirements, we must also reconsider design and implementation.

Case study: Reengineering of the UL

The module "Programming in graphical user interface (GUI)" was chosen for experimental works. The aim of the module is to present basic knowledge about tools and methods for creation of programs for Windows environment. Also students gain and/or improve skills while performing practical tasks. According to SMP, the following teaching methods are foreseen: explanation, instruction, consultation, discussions, analysis of examples, exercises, laboratory works, and projects. The distance learning course "Programming in GUI" was developed in 2003. It was implemented using Lotus LearningSpace Forum course management system (CMS). In 2007 it is redeployed in MOODLE – free, open source CMS.

The topic *Developing of user interfaces* occupies 10 academic hours of pure students' work. Also the results of studied materials are incorporated in further topics. When students use GUI toolkits to create applications, they also consider principles and recommendations for better design of GUI.

The mentioned topic was as UL for reengineering, considering triple consistency principle and requirements for computer science students. Here we present our works in a sequence, emphasizing conceptual, requirements and design levels.

1) On the conceptual level as the background for reengineering UL were chosen: a) Aligning to standards, requirements, regulation documents; b) Considering triple consistency principle.

As requirements for computer science students we accepted Computing Curricula 2001 (CC 2001) – world wide acceptable curricular guidelines for undergraduate programs in computing.

Triple consistency principle defines consistency between objectives, methods and evaluation (Kovertaite, 2006). An objective is the description of a performance you want learners to be able to exhibit, before you consider them competent (Mager, 1984). There are three components to a learning objective: 1) A performance: what the student will be doing, saying, or accomplishing; 2) Specific conditions: under what work circumstances and using what support tools does the task happen; 3) A criterion: a standard or required level of proficiency, how one will recognize success. When objectives are stated, evaluation methods must be picked. Evaluation must let us measure the accomplishment of the objective. Further, learning activities are foreseen. (Horton, 2006) separates activities into 3 types, according to their aim: 1) absorb knowledge; 2) do practice; 3) connect to life and work. The last one in formal studies can be transformed into assessment/evaluation.

Also we considered shifting learning paradigm, as it concerns computer science studies; and pedagogical Bloom and Marzano taxonomies, as they correlate well with nowadays pedagogical strategies.

2) On the requirements level concrete requirements from picked documents were retrieved. The UL covers the area HC (Human-Computer Interaction, 8 core hours) from CC 2001. It consists of two parts: a) HC1 - Foundations of human-computer interaction (6); and b) HC2 - Building a simple GUI (2). In our prior UL the second part was emphasized much more than the first, and now we tend to accentuate cognitive, socio-technical aspects of human-computer interaction. Therefore, the list of learning objectives was augmented with the following: a) Discuss the reasons for human-centered software development; b) Develop a conceptual vocabulary for analyzing human interaction with software; etc.

Aligning the lesson's material and activities according to shifting learning paradigm was also important aspect. As stated in [CC 2001], development of a computer science curriculum must be sensitive to changes in technology, new developments in pedagogy, and the importance of lifelong learning. Shifting learning paradigm in this case is very important due to the following pragmatic reasons: a) computer science is a very broad discipline; b) it is evolving rapidly; c) computer science students (at Universities) get a universal base of knowledge and skills, which further must be tailored according to the specific needs.

We also analysed the comparative study, described in [Ven, 2005], in order to formulate provided by the UL competences of higher level according to Bloom taxonomy.

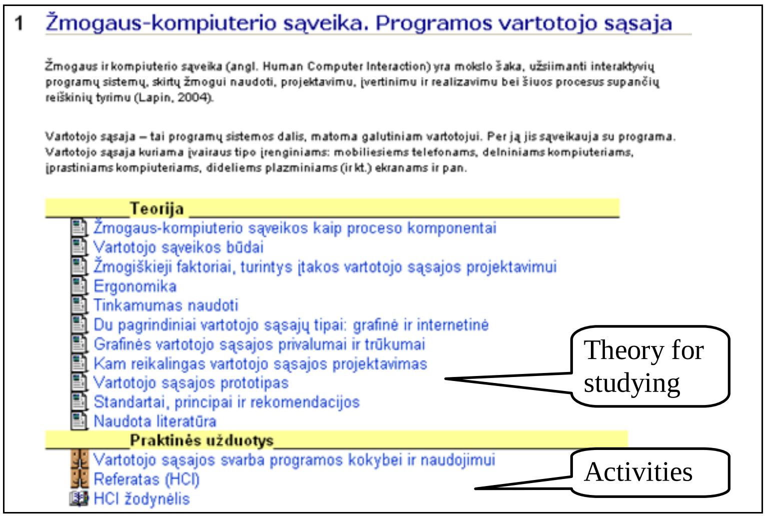
3) On the design level we made the following modifications:

- Increased level of learning material for absorbing knowledge, including – glossary added;
- Redesigned learning activities: two discussion forums added;
- Changed assessment types, or, as stated in SMP, evaluation procedure of knowledge and abilities;

Navigational and presentational structure and the essence of pedagogical scenario have not been modified.

4) On the implementation level accordingly to the framework learning resources, activities, scenarios were reimplemented. For e.g., the learning objective "Illustrate the effect of fundamental design principles on the structure of a graphical user interface" influenced the demand of extra library of examples for different principles.

In Figure 5 the representation of the structure of modified UL is shown. Here assessment is incorporated in "Activities" part of UL: participating is discussions – critique and debate on presented examples of user interfaces - are evaluated by the lecturer.



1 Žmogaus-kompiuterio sąveika. Programos vartotojo sąsaja

Žmogaus ir kompiuterio sąveika (angl. Human Computer Interaction) yra mokslo šaka, užsiimanti interaktyvių programų sistemų, skirtų žmogui naudoti, projektavimu, įvertinimu ir realizavimu bei šiuos procesus supančių reiškinių tyrimu (Lapin, 2004).

Vartotojo sąsaja – tai programų sistemos dalis, matoma galutiniam vartotojui. Per ją jis sąveikauja su programa. Vartotojo sąsaja kuriama įvairaus tipo įrenginiams: mobiliesiems telefonams, delniniams kompiuteriams, įprastiniams kompiuteriams, dideliems plazminiams (ir kt.) ekranams ir pan.

Teorija

- Žmogaus-kompiuterio sąveikos kaip proceso komponentai
- Vartotojo sąveikos būdai
- Žmogiškieji faktoriai, turintys įtakos vartotojo sąsajos projektavimui
- Ergonomika
- Tinkamumas naudoti
- Du pagrindiniai vartotojo sąsajų tipai: grafinė ir internetinė
- Grafinės vartotojo sąsajos privalumai ir trūkumai
- Kam reikalingas vartotojo sąsajos projektavimas
- Vartotojo sąsajos prototipas
- Standartai, principai ir rekomendacijos
- Naudota literatūra

Praktinės užduotys

- Vartotojo sąsajos svarba programos kokybei ir naudojimui
- Referatas (HCI)
- HCI žodynėlis

Figure 5. Representation of UL in MOODLE environment (excerpt)

Conclusion

The proposed framework contributes to increasing quality and effectiveness in delivering distance studies. Organization must follow common framework for distance study course reengineering, in order to gain minimum expenditures for overall e-Learning system maintenance.

The systematic approach and the use of software engineering methodologies in instructional engineering allow us:

1) To decline expenditures of time for direct (design and development) and indirect (redesign and redevelopment due to misunderstanding) processes;

2) To accept and manage complexity;

3) To facilitate reuse;

4) To prepare better for adaptation and personalization.

The case study shows reengineering of one topic, but the framework can be used while reengineering ULs of different granularity. Also, it can be extended in order to provide clarity and systematisation in higher layers of e-Learning system.

Further, the results of analysis of empirical student evaluation of the course and course log files can be employed in this framework.

Bibliography

[Byrne, 1992] E.J. Byrne. A conceptual foundation for software re-engineering. In Proceedings of the Conference on Software Maintenance, pp. 226–235. IEEE Computer Society Press, 1992.

[Bulanova, 2006] T. Bulajeva, L. Duoblienė, V. Targamadzė. Management of e-learning in educational organizations: how to change the university. In: V. Dagiene, R. Mittermeir (Eds.). Selected papers of the 2nd International Conference „Informatics in Secondary Schoolls: Evolution and Perspectives“, 2006, Vilnius, Lithuania, pp. 137-148.

[CC 2001] Computing Curricula 2001: Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. Available at http://www.computer.org/portal/cms_docs_ieeecs/ieeecs/education/cc2001/cc2001.pdf.

[CC 2005] Computing Curricula 2005: The Overview Report.
Available at http://www.acm.org/education/curric_vols/CC2005-March06Final.pdf.

[Chikofsky, 1990] E.J. Chikofsky, J.H. Cross. Reverse engineering and design recovery: a taxonomy. Software, IEEE, 1990, 7(1), pp. 13–17.

[Choquet, 2006] C.Choquet, A.Corbière. Reengineering Framework for Systems in Education. Educational Technology & Society, 2006, 9(4), pp. 228-241.

[Gruslyte, 2007] M.Gruslyte, L.Tankeleviciene. Institutional Approach to Distance Studies: Experience of the Case. Submitted to: 3rd International Conference ISSEP: Informatics in Secondary Schools: Evolution and Perspectives Informatics Education – Contributing Across the Curriculum.

[Hammer, 1994] M. Hammer, J. Champy. Reengineering the Corporation: A Manifesto for Business Revolution. HarperBusiness, 1994.

[Horton, 2006] W. Horton. E-Learning by Design. Pfeiffer, 2006.

[IMS-LD, 2003] IMS-LD: IMS Learning Design Information Model, 2003.
http://www.imsglobal.org/learningdesign/ldv1p0/imsld_infov1p0.html.

[Kovertaite, 2006] V.R.Kovertaite, D.Leclercq. The Triple Consistency Illustrated by e-tivities to Help Understand National and International Policies in e-learning. International Journal of Technologies in Higher Education, 2006, 3(2).

[Mager, 1984] R.F. Mager(1984). Preparing instructional objectives (2nd ed.). Belmont, CA: David S. Lake, 1984.

[Mager, 1997] R.F. Mager. Preparing Instructional Objectives: A Critical Tool in the Development of Effective Instruction.

[Sommerville, 2000] I.Sommerville. Software Engineering (6th Ed.). Addison-Wesley, Reading Massachusetts, 2000.

[Tankeleviciene, 2007] L. Tankeleviciene, T. Demenis. Distance study course reengineering based on triple consistency principle and requirements for computer science students. Theses in Proceeding of Conference on Innovative Information Technologies IIT-2007, November 8-9, 2007, Vilnius, Lithuania, pp. 37-38.

[Ven, 2005] J.H.Ven, C.P.Chuang. The Comparative Study of Information Competencies – Using Bloom's Taxonomy. The Journal of American Academy of Business, Cambridge, Vol.7, N. 1, 2005, pp. 136-142.

Authors' Information

Lina Tankeleviciene – Lecturer, Department of Informatics, Siauliai University, Visinskio st. 19, LT-77156 Siauliai, Lithuania; email: linat@splius.lt.