

UNIVERSITY OF JOENSUU
COMPUTER SCIENCE
DISSERTATIONS 10

JARKKO SUHONEN

**A FORMATIVE DEVELOPMENT METHOD FOR DIGITAL
LEARNING ENVIRONMENTS IN SPARSE LEARNING
COMMUNITIES**

ACADEMIC DISSERTATION

To be presented, with the permission of the Faculty of Science of the University of Joensuu, for public criticism in Louhela Auditorium of the Science Park, Länsikatu 15, Joensuu, on June 13th, 2005, at 12 noon.

UNIVERSITY OF JOENSUU
2005

Julkaisija
Publisher

Joensuun yliopisto
Tietojenkäsittelytieteen laitos
University of Joensuu
Department of Computer Science

Vaihdot

Joensuun yliopiston kirjasto/Vaihdot
PL 107, 80101 Joensuu
Puh. 013-251 2677, fax 013-251 2691
email: vaihdot@joensuu.fi

Exchanges

Joensuu University Library/Exchanges
P.O. Box 107, FI-80101 Joensuu, FINLAND
Tel. +358-13-251 2677, fax +358-13-251 2691
email: vaihdot@joensuu.fi

Myynti

Joensuun yliopiston kirjasto/Julkaisujen myynti
PL 107, 80101 Joensuu
Puh. 013-251 4509, fax 013-251 2691
email: joepub@joensuu.fi

Sales

Joensuu University Library/Sales of publications
P.O. Box 107, FI-80101 Joensuu, FINLAND
Tel. +358-13-251 4509, fax +358-13-251 2691
email: joepub@joensuu.fi

ISBN 952-458-663-0

ISSN 1795-7931

Computing Reviews (1998) Classification: K.3.1, K.3.2, D.2.10

Joensuun yliopistopaino

Joensuu 2005

Supervisors

Professor Erkki Sutinen
Department of Computer Science
University of Joensuu, Finland

Professor Kinshuk
Department of Information Systems
Massey University, New Zealand

Reviewers

Dr Mike Joy
Department of Computer Science
University of Warwick, United Kingdom

Professor Tapio Salakoski
Department of Computer Science
University of Turku, Finland

Opponent

Professor Ruth de Villiers
School of Computing
University of South Africa, Pretoria, South Africa

A FORMATIVE DEVELOPMENT METHOD FOR DIGITAL LEARNING ENVIRONMENTS IN SPARSE LEARNING COMMUNITIES

Jarkko Suhonen

Department of Computer Science
University of Joensuu
P.O. Box 111, FI-80101 Joensuu, FINLAND
jarkko.suhonen@cs.joensuu.fi

University of Joensuu, Computer Science, Dissertations 10
Joensuu, 2005, 154 pages
ISBN 952-458-663-0
ISSN 1795-7931

Abstract

In this study, the FORMative DEVELOPMENT Method (FODEM) is constructed for developing digital learning environments in sparse learning communities. FODEM is a thread-based method consisting of three components: (1) needs analysis, (2) implementation, and (3) formative evaluation. First, in needs analysis both theory and practice are used to define specifications. Secondly, in implementation fast prototyping in authentic learning settings is emphasized. Thirdly, formative evaluation is used to evaluate the use of the environment. An action research framework is applied to construct and evaluate FODEM method. FODEM was applied in LEAP (LEARNING Process companion) digital learning tool and ViSCoS (Virtual Studies of Computer Science) online study program development cases. The evaluation of the two cases shows that the method works in the context of sparse learning communities. FODEM has produced fast results with low investments, and it can also be used to produce contextual digital learning environments. The method itself has proved to be a simple, yet structured, design method. Finally, the structure of FODEM allows that the development process can be modeled. Therefore, a corresponding technical design environment can be developed to support the use of FODEM.

Keywords: design of digital learning environments, formative development, online learning, sparse learning communities

Acknowledgments

The work presented in this thesis was carried out at the Department of Computer Science, University of Joensuu, Finland.

The writing process of the thesis included both joyful and painful moments. There were times when I thought the work will never finish. When one task was completed, more work seem to appear. Two persons who have motivated me most during my studies are my loving wife Satu and daughter Oona. You have given me the strength to finish this work. Satu read through the manuscript several times, and she gave me important feedback and improvement suggestions throughout the writing process.

I am sincerely grateful to my supervisors, Professors Erkki Sutinen and Kinshuk. Without Erkki's encouragement and support, this thesis would have never been reality. My whole family is especially thankful to Kinshuk for arranging our nine month visit to Massey University in New Zealand. We will never forget the time spent in the Land of the Long White Cloud.

I am also thankful to Dr Mike Joy and Professor Tapio Salakoski, the reviewers of the thesis, for their constructive comments and feedback to the manuscript. I have been privileged to work with many people during my studies both in Joensuu and Massey. I would like to especially name Arto Haataja, Niko Myller, Tiong Goh, Sirpa Torvinen, and Vesa Kainulainen, for fruitful co-operation related to my work. I thank my brother Harri and Justus Randolph for proof reading the manuscript. I am also grateful to all those people who have been involved in developing and running ViSCoS studies. Finally, I wish all the best to people at the EdTech research group in Joensuu. Keep up the good work!

Financial support towards this study from East Finland Graduate School in Computer Science (ECSE) is highly appreciated. I also thank Eastern Finland Virtual University Network for the support provided in the beginning of my studies.

Joensuu, May 2005

Jarkko Suhonen

Abbreviations

Abbreviation	Description	Reference
AR	action research	Allison (2002)
CD	context design	Bayer & Holtzblatt (1998)
CS	computer science	Glass et al. (2004)
DR	design research	Barab & Squire (2004)
FODEM	formative development method	Suhonen & Sutinen (2005)
ID	instructional design	Moallem (2001)
IMS	Instructions Management System	Anderson & Wason (2001)
IS	information systems	Glass et al. (2004)
JSZ	jagged study zone	Gerdt et al. (2002)
LEAP	learning process companion	Suhonen & Sutinen (2004)
LMS	learning management system	Vrasidas (2004)
MVC	model-view-controller	Butler (2001)
UKOU	United Kingdom Open University	Castro et al. (2001)
PMA	problem management assistant	Rautama et al. (2001)
PPA	problem processing assistant	Suhonen & Sutinen (2002)
SDLC	software development life cycle	Whitten et al. (2004)
SE	software engineering	Glass et al. (2004)
ViSCoS	virtual studies of computer science	Torvinen (2004)
WS	Woven Stories	Gerdt et al. (2001)
ZPD	zone of proximal development	Vygotsky (1978)

Table of Contents

1 INTRODUCTION	1
1.1 BACKGROUND	1
1.2 IMPORTANCE OF THE RESEARCH PROBLEM	2
1.3 CONTRIBUTIONS.....	3
1.4 DEFINITION OF THE MAIN TERMS	4
1.5 STRUCTURE OF THE THESIS	5
2 ACTION RESEARCH FRAMEWORK IN DEVELOPING THE FODEM METHOD	7
2.1 DEVELOPMENT AS A RESEARCH ACTIVITY	7
2.2 RESEARCH QUESTIONS.....	7
2.3 ACTION RESEARCH FRAMEWORK.....	9
2.4 METHODOLOGICAL REMARKS.....	12
3 DIGITAL LEARNING ENVIRONMENTS IN SPARSE LEARNING COMMUNITIES.....	14
3.1 DIGITAL LEARNING ENVIRONMENTS TO SUPPORT LEARNERS	14
3.2 CRITERIA FOR A FUNCTIONAL DESIGN METHOD	17
4 FORMATIVE DEVELOPMENT METHOD.....	19
4.1 GENERAL OVERVIEW	19
4.2 CONCEPTS	20
4.3 TECHNICAL ISSUES.....	28
4.4 REMARKS ON THE FEASIBILITY OF FODEM	29
5 LEAP – DIGITAL LEARNING TOOL	32
5.1 INTRODUCTION TO THE MAIN CONCEPTS	32
5.2 FUNCTIONAL LAYERS FOR IMPLEMENTING THE PEDAGOGICAL SERVICES.....	36
6 FODEM IN DIGITAL LEARNING TOOL DEVELOPMENT – THE CASE OF LEAP.....	46
6.1 OVERVIEW.....	46
6.2 THREAD 1: THE FIRST PROTOTYPE	49
6.3 THREAD 2: RE-DESIGN OF THE TECHNICAL ARCHITECTURE	73
6.4 THREAD 3: MOBILE ADAPTATION EXTENSION	84
6.5 SUMMARY AND DISCUSSION	87
7 FODEM IN ONLINE STUDY PROGRAM DEVELOPMENT – THE CASE OF VISCOS	94
7.1 BACKGROUND	94
7.2 ViSCoS DEVELOPMENT PROCESS	94
7.3 SUMMARY	104
8 DISCUSSION.....	107
8.1 RESULTS OF THE STUDY	107
8.2 EVALUATION OF THE RESEARCH RESULTS.....	112
9 CONCLUSION AND FUTURE WORK.....	115
10 REFERENCES	119

1 Introduction

1.1 Background

The purpose of this study is to construct a design method for the development of digital learning environments in sparse learning communities. A *sparse learning community* refers to a student population which is spread out over a relatively large geographic region or a long period of existence. Since learning takes often place in a limited cultural situation – geographically or temporally – a sparse learning community is also small in the number of students.

ViSCoS (Virtual Studies of Computer Science) online study program run in the Department of Computer Science, University of Joensuu is an example of a sparse learning community (Haataja et al., 2001a). In ViSCoS, students study the first year university-level computer science courses via the web (Haataja et al., 2001b; Haataja et al., 2001c). The curriculum of ViSCoS consists of three main areas – the preliminaries of ICT, basics of programming with Java, and an introduction to computer science (Torvinen, 2004). The amount of students in ViSCoS courses is typically under 100, and the students are scattered around in a large area. Altogether 94 students have completed the program between the years 2000-2004. Because ViSCoS is currently run as a continuing education project, students pay 400 euro study fee for the whole program.

Digital learning environments are technical solutions for supporting learning, teaching and studying activities. A sparsely located community needs a digital learning environment because the students live far away from each other and from the institution. A digital learning environment can range from an online study program down to a piece of digital learning material. An online study program, such as ViSCoS, consists of several smaller digital learning environments. Hence, digital learning environments can vary in complexity and comprehensiveness. As the importance of a digital learning environment increases in supporting learners, more is expected from the environment. For instance, in ViSCoS the aim has been to

provide rich digital learning environments which support students in multiple ways (Sutinen & Torvinen, 2003).

A design method behind the digital learning environment development is important for the end-result. Design methods can help designers focus their efforts on the main aspects of the development process. For instance, a solid digital learning environment design method could assist in creating innovative solutions (Design-Based Research Collective, 2003). The design method should be consistent and pragmatic to ensure the achievement of meaningful results. An important aspect of a design method from a computer science perspective is the possibility to model the development process. This would allow, for instance, the creation of a software environment to support the design. A formative development enables a gradual development of digital learning environments based on the experiences of using the environment in authentic settings.

1.2 Importance of the research problem

Three needs or challenges for a functional design method in sparse learning communities can be identified. First, the resources often limit the investments for long-standing and heavy development processes (Bork & Gunnarsdottir, 2001). Hence, fast results are needed with relatively *low investments*. A functional design method should promote cost and time efficiency (Avgeriou & Retalis, 2002).

Secondly, despite the low resources, the ad-hoc design approaches should be avoided. A design method should produce functional and efficient environments with low risk of failure (Avgeriou & Retalis, 2002). The method should also be open for the needs of different development situations. The development of an online study program can be totally different from that of a digital learning tool. When novel solutions are designed, the requirements and expectations can change during the development process (Fallman, 2003; McCracken, 2004; Moonen, 2002). Hence, a design method should be able to react to the changes.

The third challenge is the need to develop *contextual*, meaningful, digital learning environments. The method must be able to respond, for instance, to the diversity of learners, the technologies available, and to the cultural aspects of the learning context (Soloway et al., 1996). The challenge is to determine the real needs and problems of learners (Abdelraheem, 2003; Watanabe et al., 1999).

1.3 Contributions

The main contribution of this study is the construction and evaluation of *FOrmative DEvelopment Method* (FODEM) with an action research framework. FODEM has been created to meet the needs of sparse learning communities. The method has been applied in LEAP (LEARning Process companion) digital learning tool and ViSCoS development cases. The evaluation of the two cases shows that the method works in the context of sparse learning communities. FODEM has produced fast results with relatively low investments. FODEM can also be used to produce contextual digital learning environments. The method itself has proved to be a simple, yet structured, design method. Finally, the structure of FODEM can be modeled. Therefore, a corresponding technical design environment can be developed to support the use of FODEM.

A common feature in action research is that a practitioner solves a practical problem and evaluates the solution. I have been involved in defining, constructing and evaluating the FODEM method. Furthermore, I have also participated both in ViSCoS and LEAP development. The construction of FODEM started with identified needs in the context of sparse learning communities as explained in Section 1.2. At first, FODEM was applied on a general level at LEAP development. The three components, that is, *needs analysis*, *implementation* and *formative evaluation* were identified as core features of the method. It was also discovered that the development process of ViSCoS can also be modeled with FODEM too; more or less the same design principles were applied (Torvinen, 2004). At the same time, the main concepts and features of the method have gradually refined.

I have been responsible for the LEAP development, the main tasks having been the concept design and evaluation of the tool. Part of the concept design in LEAP was the *jagged study zone* (JSZ) (Gerdt et al., 2002). I was involved in creating the main ideas for JSZ that were merged later to the LEAP development. The second concrete step in the concept design of LEAP was the analysis of paper-based portfolios in the ViSCoS Programming Project Course (Suhonen & Sutinen, 2003). Some of the ideas from the analysis were included to the concept design of LEAP. I have also presented the first designs and analyzed the development of LEAP based on FODEM (Suhonen & Sutinen, 2002; Suhonen & Sutinen, 2004; Suhonen & Sutinen, 2005). I was the main author in all four papers. A mobile adaptation extension is a part of LEAP development. Three papers have been published on the

concept design of the mobile adaptation extension (Kainulainen et al., 2004; Kinshuk et al., 2003; Kinshuk et al., 2004). The main aim of articles was to analyze the suitability of different adaptation techniques to implement the mobile adaptation extension. I was the main author in the last paper. Other two are result of an equal collaboration among the authors.

In ViSCoS, I have created courses, managed the whole program and evaluated the applied learning materials and methods. I analyzed the first designs and experiences of ViSCoS together with the colleagues involved in ViSCoS development (Haataja et al., 2001a; Haataja et al., 2001b; Haataja et al., 2001c; Kareinen et al., 2001; Kareinen et al., 2002). Data mining forms one part of the ViSCoS development. I was involved in experimenting with the first ideas on using data mining in ViSCoS (Myller et al., 2002). All papers related to ViSCoS development are result of an equal collaboration among the authors.

1.4 Definition of the main terms

Learner

Learners are persons whose learning is supported with digital learning environments. I use the term “learner” to describe the persons on a general level (e.g. not specific persons). I use the term “student” when referring to a particular population of learners. The term “user” is applied mainly in Chapter 5 when describing the function of LEAP.

Instructor

An instructor is a person responsible for supporting learners in online study programs. Instructors can also take part in the development process of a digital learning environment.

Design method

A design method helps designers to develop computer software, such as digital learning environments. The aim of a design method is to depict the procedures and tasks needed during the development. A design method can also help designers focus on the most important aspects of the development process.

Computing

In this study, the computing field is defined to consist of computer science (CS), software engineering (SE) and information system science (IS) (Glass et al., 2004). The pragmatic aim of the computing field is to produce technical solutions performed with computers.

1.5 Structure of the thesis

In Chapter 2, I describe the action research framework applied to construct the FODEM. Chapter 2 also includes the research questions of the study. Finally, I discuss how the results of the action research framework can be evaluated. In Chapter 3, I analyze the characteristics and needs of sparse learning communities. The FODEM method is designed to meet the challenges identified in the chapter. Furthermore, the criteria for a functional design method in the context of sparse learning communities are defined.

In Chapter 4, I define and construct the FODEM method. There are concrete examples and visualizations of the method. In Chapters 5 and 6, I describe the development case of LEAP. Chapter 5 includes the presentation of the function and the core structure of LEAP. There are also concrete examples on how to use LEAP. Chapter 6 consists of a more detailed description of how FODEM has been used in LEAP development. In Chapter 7, I describe the development case of ViSCoS on a higher abstraction level; there is no description of the details of the development.

In Chapter 8, I conclude the results of the study. The chapter includes an analysis of the FODEM method based on the evaluation of the two development cases. I also discuss how the results of the present research are adaptable to other contexts. Finally, I illustrate how FODEM relates to other design methods in the context of sparse learning communities. In Chapter 9, I answer the research questions of the study, and I also describe the future work related to FODEM. Figure 1.1 illustrates the structure of the thesis and the relationships between the chapters.

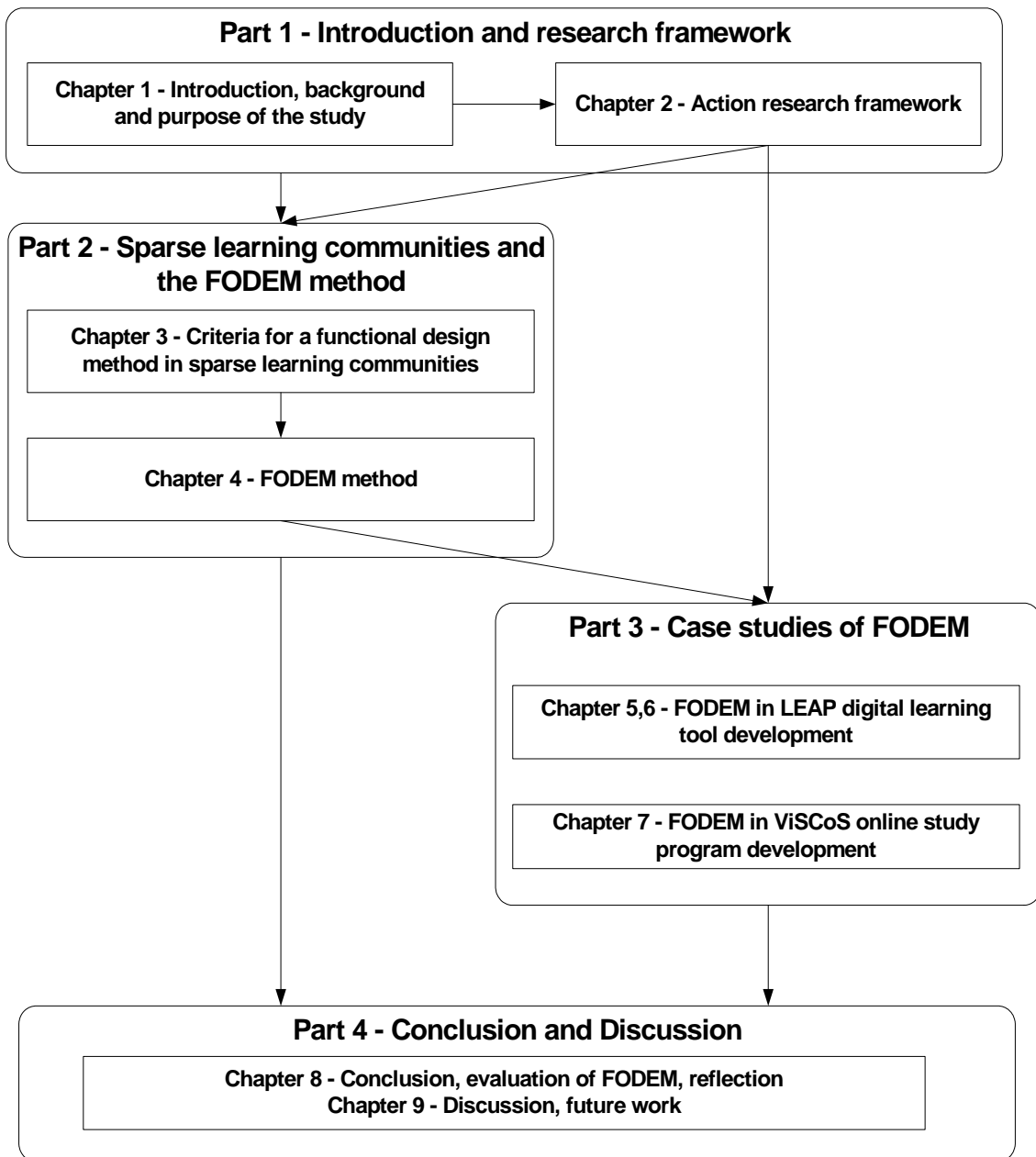


Figure 1.1: Structure of the thesis

2 Action research framework in developing the FODEM method

2.1 Development as a research activity

The aim of the current study is to develop a design method for digital learning environments in sparse learning communities. Hence, compared to other studies, the focus is not on evaluating or analyzing existing methods (Cohen & Manion, 1989; Fincher & Petre, 2004; Popkewitz, 1984). In addition, the aim is not to interpret individuals' subjective constructions of the reality (Creswell, 1994; Greening, 1997).

In the study, a pluralistic *action research framework* is used in constructing FODEM. Action research is a general method for solving practical problems and at the same time collect scientific knowledge. In computing field, action research is often used by practitioners to solve a practical software development problem (Allison, 2002; Clear, 2001; Kock et al., 1997; Rose, 2000). The cyclical nature of the method allows that designers can experiment with their design solutions (Avison et al., 1999). The following reasons can be identified for choosing this particular method. First, action (e.g. development) is interwoven with research. The aim is to construct and improve the FODEM method based on its evaluation (Clear, 2004; Conole et al., 2004). Secondly, pluralistic research methods and multiple data sources are needed to answer the different research questions (Landry & Banville, 1992; Miles & Huberman, 1994). Thirdly, the research interest was based on a concrete need for a digital learning environment design method in sparse learning communities (Baskerville, 1999; Creswell, 2003).

2.2 Research questions

The research problem in this study is to develop a design method for digital learning environments in sparse learning communities. The research problem is processed through three types of research approaches: descriptive, constructive and evaluative (Glass et al., 2004). Descriptive research approaches are often based on a literature review or describing a developed system or method. Constructive research

approaches include activities, such as formulating a process, method, model, taxonomy or an algorithm. The research results are expected to be new concepts and models. According to Glass et al., constructive research approaches are dominant in CS and SE fields. In evaluative research approaches, an existing theory, process or a method is evaluated using appropriate research methods, such as case and field studies. The aim is to explore existing concepts and systems. According to Glass et al., evaluative research approaches are more common in IS than CS and SE.

The constructive contribution of this study is the creation and definition of the FODEM method. The main concepts and features of FODEM are created based on the needs of sparse learning communities. Hence, the constructive (C) research questions are:

C. What is FODEM? What are the core features of FODEM? How can FODEM be applied?

The feasibility of FODEM is evaluated through two development cases: LEAP and ViSCoS. The two cases are evaluated to analyze the weaknesses and strengths of the method. The requirements and constraints of using FODEM are also analyzed. Finally, future development steps for the method are identified. Hence, the evaluative (E) research questions are:

E. What results has FODEM produced? What are the requirements for using the method? What are the strengths and weaknesses of the method? How can FODEM be improved?

The concept of a sparse learning community is defined through descriptive research approaches. Descriptive research questions are also used to analyze the criteria for a functional design method in the context of sparse learning communities. Finally, FODEM is related to the other design methods. Hence, the descriptive (D) research questions are:

D. What are the needs for digital learning environments in sparse learning communities? What are the criteria for a functional design method? How does FODEM relate to other design methods?

2.3 Action research framework

An action research framework is applied to develop the FODEM method. In the framework, the research questions presented above are tightly related to each other. Descriptive (D) research questions are answered by the literature reviews in Chapters 3 and 4. First, characteristics of sparse learning communities are defined in Chapter 3. Criteria for a functional method are derived from the characteristics of sparse learning communities. The literature review in Chapter 3 creates needs and requirements for the FODEM. Furthermore, FODEM is evaluated based on the criteria depicted by descriptive research questions. The literature review in Chapter 4 investigates the existing development methods in relation to FODEM.

Research questions C and E are in a dynamical relation to each other. Features, goals and main concepts of the method (constructive contribution) affect the evaluation of the method. Additionally, the results of the FODEM evaluation affect the construction of FODEM. First, FODEM is applied in LEAP and ViSCoS development. The two case studies are also used to evaluate FODEM. Figure 2.1 visualizes the action research framework for developing the FODEM method.

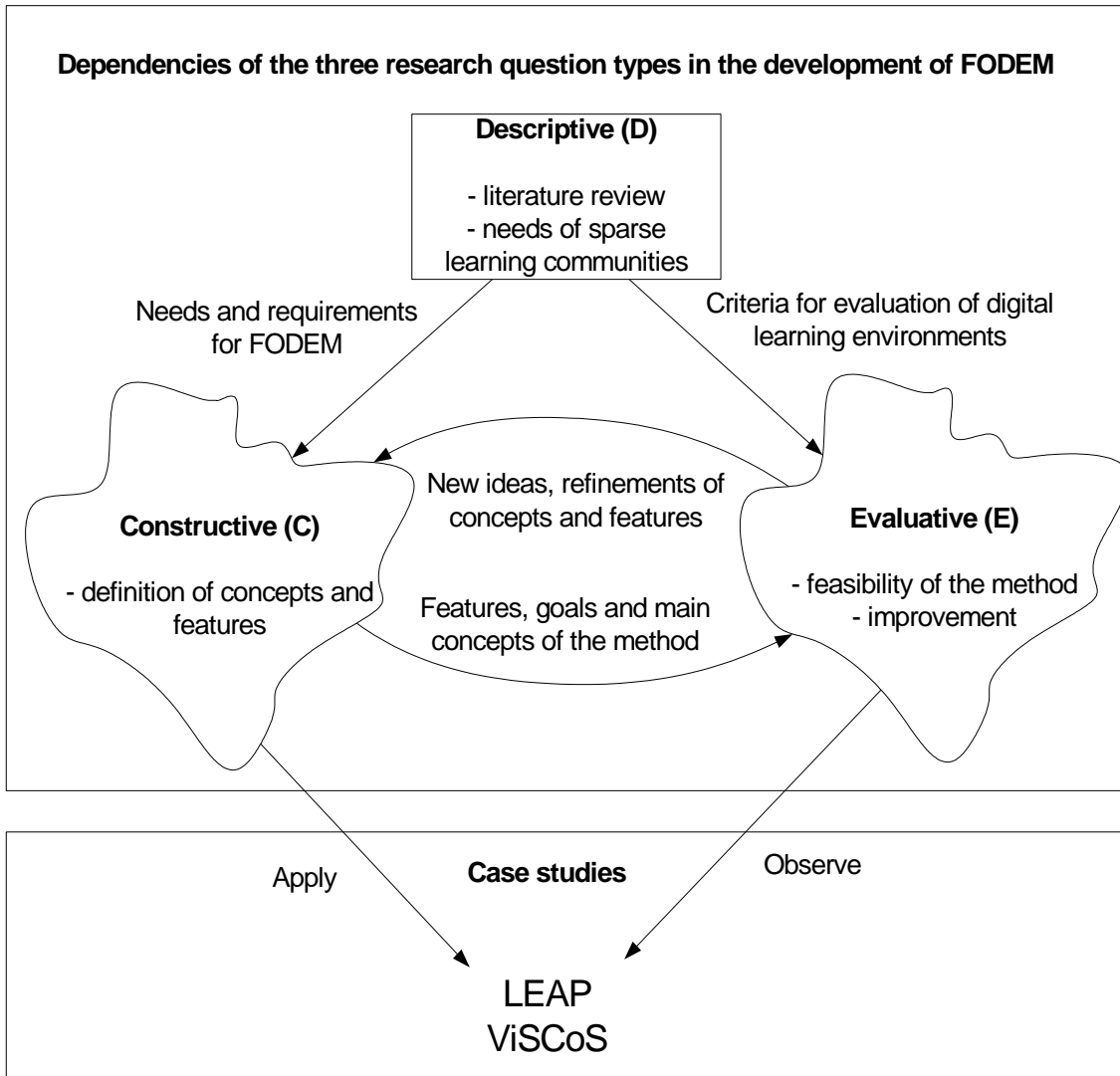


Figure 2.1: An action research framework to develop the FODEM method

The FODEM method was constructed (C) to meet the concrete needs in sparse learning communities. The concepts and features of FODEM are defined in Chapter 4. At first, FODEM only included general features. Gradually more concepts and specific features have been added based on the experiences in LEAP and ViSCoS development.

The two case studies LEAP and ViSCoS are evaluated to analyze (E) the feasibility of FODEM. A re-engineering method is applied to examine the course of the development and factors affecting the development process. Because LEAP is the main case study, it is presented as detailed and transparent as possible (Cohen et

al., 2002; Merriam, 1998). The aim is to embed the evaluation into the development context (Yin, 1994). I use reflection to draw conclusions from the results of the evaluation (Creswell, 2003).

2.3.1 Case 1: FODEM in LEAP development

LEAP is a web-based digital learning tool. It has two main functions: digital learning portfolio and creative problem solving support (Suhonen & Sutinen, 2002). In LEAP development, three parallel threads can be identified (Suhonen & Sutinen, 2005). During the first thread, the *first prototype* of the tool was implemented based on the design ideas and concepts from both theory and practice. The tool was inspired by digital learning portfolio, creative problem solving support and provocative agent concepts. Furthermore, there was a need in the ViSCoS Programming Project Course to support management of the projects (Suhonen & Sutinen, 2003). The first thread includes two studies on the use of the tool in authentic learning settings (Suhonen & Sutinen, 2004; Suhonen & Sutinen, 2005). A similar evaluation scheme was used in both studies. The scheme has two parts: analysis of how students have been using the tool and analysis of students' opinions about the tool. The goal of the two analyses is to produce concrete knowledge and ideas for the development of the tool.

In the second thread, LEAP was modified based on the results of the two studies. Totally new features were not added, but the existing features were refined. The *technical architecture of the tool was re-implemented* because of the requirements posed by the third thread. A third study was conducted within the second thread, and it was evaluated with the similar evaluation scheme as in earlier studies. The third thread in LEAP development was the *mobile adaptation extension* to adapt the functions of the tool to mobile devices (Kainulainen et al., 2004; Kinshuk et al., 2003; Kinshuk et al., 2004). The mobile adaptation extension thread is still on the concept design phase. The development process of LEAP is presented in more detail in Chapters 5 and 6.

2.3.2 Case 2: FODEM in ViSCoS development

The second case, development of ViSCoS, is not presented as comprehensively as the LEAP case. A post hoc research approach is used where the development of ViSCoS is presented based on the available documents and articles (Kareinen et al.,

2001; Kareinen et al., 2002; Meisalo et al., 2002; Meisalo et al., 2003b; Meisalo et al., 2004; Torvinen, 2004). The goal is to interpret the dynamics of the ViSCoS development process (Miles & Huberman, 1994).

Three threads can be identified in the ViSCoS development: the first designs, loop of improvements and new technical solutions. In the *first designs* thread, the first versions of the courses were created (Haataja et al., 2001a; Haataja et al., 2001b; Haataja et al., 2001c). In the second thread - *loop of improvements* - the focus was on improving the programming courses because those were the most difficult ones for the students (Torvinen, 2004). Within the loop of improvements, several studies were conducted to investigate the drop out phenomena in ViSCoS. The third thread in ViSCoS development involved *new technical solutions* to support learners. Four sub-threads can be identified: LEAP, the ethical argumentation tool Ethicsar, the Jeliot program visualization tool and data mining techniques. Ethicsar is a tool for argumentation and evaluation of ethical issues and questions (Jetsu et al., 2004). Jeliot allows novice programmers to visualize their own Java codes (Moreno, 2003; Moreno et al., 2004a). The development of data mining techniques was focused especially on processing the data related to the assignments in the courses (Myller et al., 2002). The aim of the data mining efforts is to create intelligent support for learners. The development process of ViSCoS is presented in more detail in Chapter 7.

2.4 Methodological remarks

A general methodological issue relevant to the present study is the use of the first person singular. I recognize that the use of the first person singular in scientific texts is considered to be against the idea of objectivity (Ratner, 2002). Because I have used pluralistic research approaches with mainly qualitative data analysis, reflection is a method to decrease subjective bias (Creswell, 2003). The only way I can reflect to myself is to use the first person singular. I have also used the first person singular to emphasize my own personal decisions or contributions. I use pronoun “we” to refer to the people involved in the development of LEAP or ViSCoS. I want to emphasize that the development has been a result of a tight collaboration among the persons involved in the process.

Each research question type, descriptive, constructive and evaluative, has its own criteria for evaluating the results of the research methods used. First, literature

reviews for answering the descriptive research questions can be evaluated through consistency and soundness of the literature used. When the literature review is consistent, it does not produce any contradictory or vague results. Literature review should not include any outdated literature.

The constructive research questions are answered through the creation and definition of the FODEM method. FODEM was created based on the needs and challenges of sparse learning communities. Hence, the outcomes of the constructive research questions (e.g. the FODEM method) are evaluated through usefulness and pragmatic benefits of the method in the context of sparse learning communities (Juuti, 2005).

Finally, reliability and validity are criteria for evaluative research results. *Reliability* measures the stability of the research results, that is, whether another researcher with same methods would produce similar results (Fincher & Petre, 2004). Reliability is also related to generalizability. Highly reliable results enable the researcher to draw context dependent conclusions. The main instrument for increasing the reliability is to enhance the objectivity of the research methods used. *Validity* measures how accurately the research findings describe the research situation, e.g. how well the research findings match reality (Anfara et al., 2002; Merriam, 1998). According to Fincher and Petre (2004), a research method is rarely both reliable and valid. Methods that are grounded to the reality, such as case studies, can be highly valid. However, the problem is messy data and time-consuming data analysis. Controlled studies, for instance, have more potential to produce reliable results than case studies, but the problem can be the unnatural and restrictive research situation.

3 Digital learning environments in sparse learning communities

3.1 Digital learning environments to support learners

In sparse learning communities there are often limited resources to invest into the development of digital learning environments. The design team consists of a few persons with multi-disciplinary skills and several responsibilities. The development period can also be short. The advantage of sparse learning communities is that the number of students does not restrict the pedagogical solutions used. In mass courses, for instance, the aim is often to create courses efficiently: there is no individual support for learners. Table 3.1 shows the differences between sparse and dense learning communities.

Table 3.1: Features of sparse and dense learning communities

	Sparse	Dense
Design costs per course	10 000 EUR	100 000 EUR
Number of students per course	under 100	Several hundreds up to thousands
Design team	A few persons with multi-disciplinary skills and several responsibilities	10-20 specialists
Development time	2-4 person months	1-2 years
Instructional methods	Information processing	Information delivery
Production	Tailor-made	Mass production
Feasible digital learning environment	Multipurpose digital learning tools	Re-usable learning objects

The Finnish education context gives several examples of sparse learning communities. The objective of education in Finland is to support the development of

the citizens into good and balanced members of the society (Marlow-Ferguson, 2002). The education is mainly free of charge from basic to higher education. The aim in higher education is to promote independent research and to educate the students as critical members of the society. In recent years, the pedagogical methods in Finnish school institutions have focused on learner-centered approaches with an emphasis on collaborative studying. All universities in Finland are public and receive tax based public funding. Students pay no tuition fees and receive some of the teaching materials for free. Mega-universities, such as the United Kingdom Open University (UKOU), are examples of dense learning communities. The UKOU offers a vast range of degrees with about 200,000 students (Castro et al., 2001). A full-scale design system is applied to produce learning materials and tools for courses. Several years and hundreds of thousands of euros are spent to achieve the desired end-results (Bork & Gunnarsdottir, 2001).

Digital learning environments are technical solutions for supporting learning, teaching and studying activities (Govender, 2004). A sparsely located community needs a digital learning environment because the students live far away from each other and from the institution. A digital learning environment can range from an online study program down to a piece of digital learning material. Therefore, digital learning environments are often combinations of different technical solutions. A digital learning environment can also include physical elements, such as robotics (Abdelraheem, 2003). Human participants may also have an important role in a digital learning environment.

A *digital learning content* often includes the curricula being learned. In their study, Mioduser and Nachmias (2002) found that almost 85% of the investigated educational websites supported activities based on digital learning content. In these cases, the production of the digital learning content is often equaled with quality of teaching or learning (Heydenrych, 2004; Polsani, 2003a). The interest in the educational technology field in recent years has been, for instance, to formally define the digital learning content as learning objects (Wiley, 2001).

A *Learning Management System (LMS)* is used to create, manage or run the courses in sparse learning communities. These systems are also called Virtual Learning Environments (VLE) (Armitage et al., 2004). Delivery and administration of courses, management of students and learner-support services are the core functions of LMSs (Farrel, 2001; Vrasidas, 2004). A typical LMS includes

communication and resource sharing features, such as email, chat, discussion forums and group work applications (Duffy & Cunningham, 2001). Furthermore, LMSs are capable of supporting the creation and management of digital learning content in a form of lecture notes, handouts, examples and assignments. Automated management of assignments, record keeping and monitoring of students are often integrated in an LMS. According to Vrasidas (2004), LMSs are not often re-usable and they are poorly customized to meet the needs of specific learning contexts.

Digital learning tools are applications designed to support learning or studying (Fetherston, 2001; Kommers, 2004). A digital learning tool can include features of an LMS or contain digital learning content, and it includes specific pedagogical constructs to support learning. Different types of tools can serve different purposes. Digital learning tools can be used, for instance, to help learners to define their learning objectives, collect and evaluate the learning materials, evaluate their learning processes and maintain motivation (Ehrmann, 2004; Jonassen, 2000; Kommers, 2004; Lowyck, 2002; Schroeder, 2002). Journal writing tools, for instance, help learners represent their knowledge, and visualization tools help learners express themselves visually (Vrasidas, 2004). Cognitive tools, such as a concept map drawing tool, support learners' cognitive abilities in problem solving and learning (Jonassen & Reeves, 2001; Lanzig, 2004; Stoyanova & Kommers, 2002).

MetaAHEAD digital learning tool provides metacognitive support for science students (McLoughlin & Hollingworth, 2001). MetaAHEAD includes, for instance, online discussion spaces where learners can share their experiences and gain feedback from each other. Another example of a digital learning tool is the Jeliot program visualization tool (Ben-Ari et al., 2002). Jeliot allows novice programmers to visualize their own Java code. Digital learning tools can also be used for community building. For instance, EduCities adopts the structure and operation of a real city for supporting a learner-oriented learning community (Chan et al., 2001). The EDUCOSM tool supports joint annotation and open-ended knowledge sharing, and the EDUCO tool visualizes learners' navigation activities (Kuruhila, 2003b).

An *online study program*, such as ViSCoS, can also be defined as a digital learning environment. A study program can be built on an LMS. It typically also includes digital learning contents for different purposes. Various digital learning tools can be provided to support learners. Finally, an online study program includes

human participants. In ViSCoS, instructors at the university and local tutors provide human support for learners.

3.2 Criteria for a functional design method

When looking at the features of sparse learning communities in Table 3.1, three interwoven challenges to evaluate the feasibility of a design method in the context of sparse learning communities can be identified. First, the available resources often limit the investments for long-standing and heavy development processes. For instance, the time and resource constraints do not allow comprehensive analysis and design procedures. Hence, fast results are needed with relatively *low investments*. A method that meets these requirements cannot include any heavy structures. A functional design method should promote cost and time efficiency (Avgeriou & Retalis, 2002).

Secondly, despite the low resources, the challenge is to avoid ad-hoc design approaches. The applied *design methods* should produce functional and efficient digital learning environments. Ad-hoc approaches are often applied because of opportunistic expectations about the costs and results of the development (Bork & Gunnarsdottir, 2001). The design is based on intuition and recipes; there is no solid approach for supporting the design and implementation (Lowyck, 2002). In sparse learning communities, there is no room to make expensive mistakes. Hence, a functional design method should help in decreasing the risks of a failure (Avgeriou & Retalis, 2001). The method should also be open for the needs of various development situations. The development of an online study program can be different from that of a digital learning tool. When novel solutions are developed, the requirements and expectations can change during the development process (Moonen, 2002). The original specifications may be altered or even abandoned during the process. Hence, a functional design method should be flexible for changes when the development situation or the functional specifications change.

The third challenge is the need to develop *meaningful digital learning environments* in the context of sparse learning communities. Meaningful results can be achieved through considering the contextual factors of the design situation. The design method must be able to respond, for instance, to the diversity of learners, available technologies, and to the cultural aspects of the learning context (Soloway

et al., 1996). The lack of unified theories for functional specifications of digital learning environments in sparse learning communities can also make the design situation unpredictable (Moonen, 2002). This is especially true when the whole design situation is new and the goals and expectations are loosely defined in the beginning of the development (Fallman, 2003; McCracken, 2004). Contextual factors can have an impact to the support provided to learners; the challenge is to determine the real needs and problems (Abdelraheem, 2003; Watanabe et al., 1999). In an ideal situation, there should be an individualized combination of challenge and guidance, empowerment and support (Häkkinen, 2002). The needs can also change, and the environment should be able to adapt to the changes (Kommers, 2004).

The intersection of the three challenges can be used to evaluate the feasibility of a design method in sparse learning communities (Figure 3.1). In an ideal situation, the aim is to avoid ad-hoc approaches, but still produce meaningful and fast results with reasonable investments. Figure 3.1 visualizes a design method which requires average investment, uses semi-structured design approaches and produces results that consider some of the needs in a design context. Another method might require high investments with structured design approaches and still the results do not take into account the real needs of the design context.

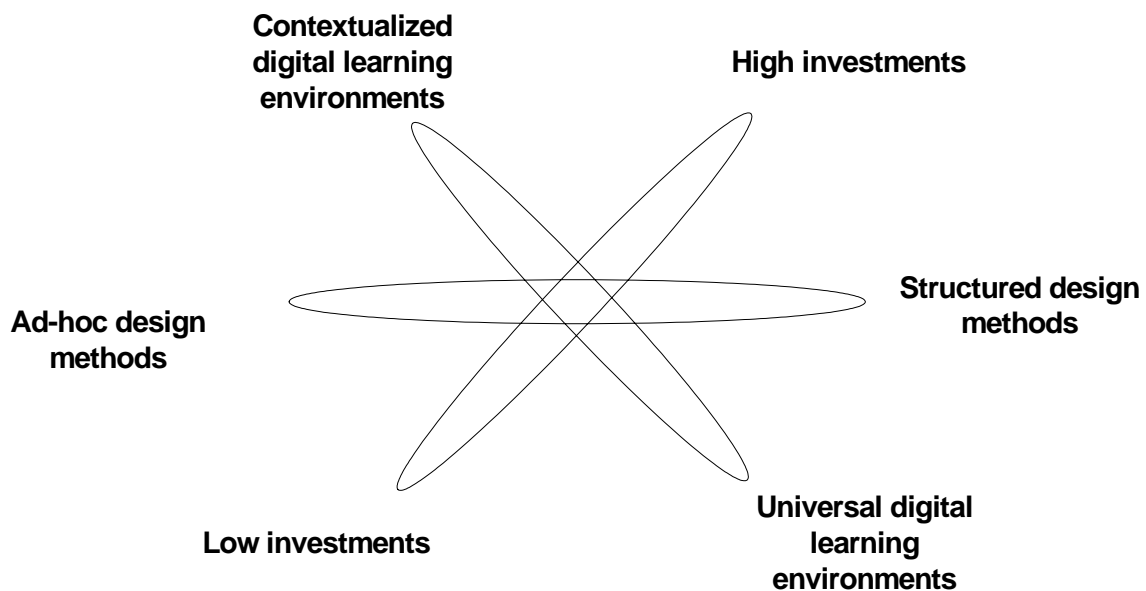


Figure 3.1: Evaluation criteria for a design method in the context of sparse learning communities

4 Formative Development Method

4.1 General overview

In FODEM, the development process consists of simultaneous *threads* with a specific goal. The term “thread” is used rather than a “stage” or “phase” because the development process consists of several threads progressing parallel. A thread has three interdependent, dialectical components: needs analysis (NA), implementation (I) and formative evaluation (FE). *Dependencies* represent the interdependent structure of threads, the interaction among components and the dependencies between components in different threads. Figure 4.1 illustrates a possible FODEM scenario. The dependency between Thread 1 and 2 shows that Thread 1 has created a need for the re-design of the technical architecture. The arrow from Thread 3 to Thread 2 illustrates that the implementation component in Thread 2 is dependent on the needs analysis component in Thread 3.

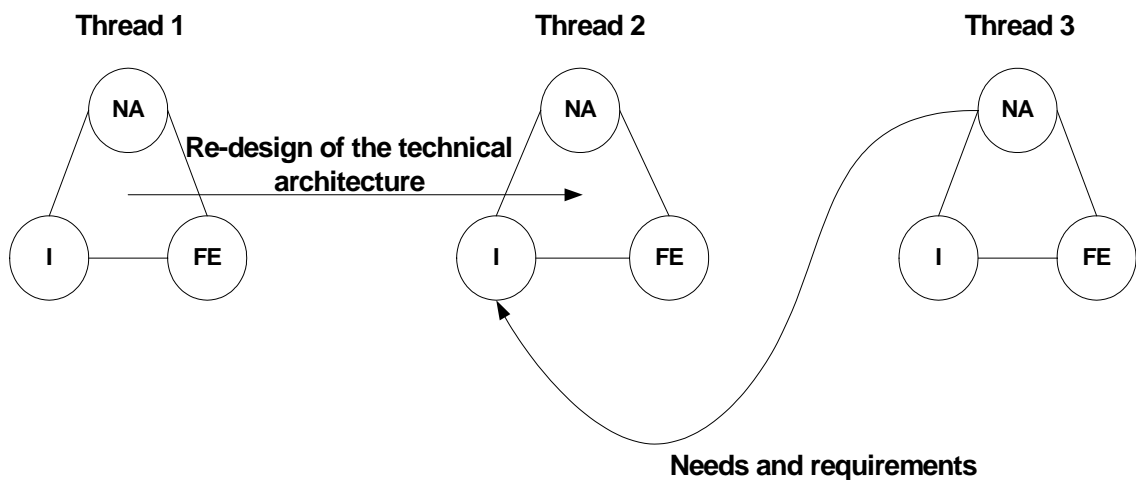


Figure 4.1: A possible FODEM scenario

4.2 Concepts

4.2.1 FODEM components

Needs analysis (NA)

In *needs analysis*, the aim is to identify the design solutions in each thread. Needs analysis includes the definition of the main concepts, roles and desired goals of the environment. Within needs analysis, the pedagogical services of the environment, requirements of the design context and possibilities of technology are also identified. A needs analysis can also include the analysis of formative evaluations in other threads.

In the needs analysis, the practical needs of the learners in a design context can be analyzed. Cultural factors are important too. If learners are used to working in a passive environment, there might be major difficulties when they are required to take a more active approach. Learning theories could be used as a source of inspiration when defining the pedagogical objectives of the environment. The design solutions can also emerge from novel ideas; sometimes there are no theories to build on. In the needs analysis component, the designers can also explore the possibilities of emerging technical solutions, such as mobile learning. However, if the technical goals are too ambitious, the problem is the available resources. The usability issues are also important. Technology should not have any negative impact on the learning situation.

Implementation (I)

The *implementation* component is used to implement the design solutions identified in needs analysis. The aim is to implement the solutions quickly to enable an experiment with learners as early as at the first stages of the development. Because the environment is developed fast, the implementation does not necessarily include all the phases of general software development, such as error checking and testing. Furthermore, some parts of the environment are not completed, such as help systems.

Naturally, the early use of the environment is not unproblematic. The designers expose their ideas and solutions under critique. They have to understand that the goal is not to produce perfect results at once, but that the final goal may be realized after a long development process. Learners have to understand that some aspects of the environment are unfinished. The feedback received about the environment can be harsh. However, if the feedback is constructively analyzed, it can be used to improve the environment. An important aspect is to make the situation transparent to the learners. If the learners are informed about the goals and threats of the experiment, there is an opportunity for getting insight about the function of the environment. Furthermore, learners can participate in the development process by expressing suggestions and development ideas.

An important aspect of early experimenting is the level of maturity of the environment. If the environment is immature, the experiment might not be fruitful. This means that the core functionalities should be working satisfactorily. There is a balance between getting fast experience and the readiness of the environment. For instance, if the environment includes too many usability problems, it may lead learners' attention to secondary issues. In the worst case, learners might not even realize the main functions of the environment. However, if learners are exposed to the development process too late, major changes to the core function of the environment are expensive and difficult to implement.

Formative evaluation (FE)

In the *formative evaluation* component, the experiment of the developed environment within the thread is evaluated. Multiple data sources ensure that a comprehensive and rich picture is gathered. In *use of the environment analysis*, learners' operations in the environment are evaluated. Activity logs, databases, and browsing traces are valuable data sources. The aim of *experience analysis* is to interpret learners' personal opinions, experiences, perceptions and feelings about the environment. Experience analysis is used to reflect the behaviour of the system based on the subjective perceptions of the learners. The goal is to reveal novel ideas for the development of the environment. Hence, attention is paid to surprises or unique experiences for getting fresh ideas. Figure 4.2 visualizes how the two analysis methods are applied in the formative evaluation component.

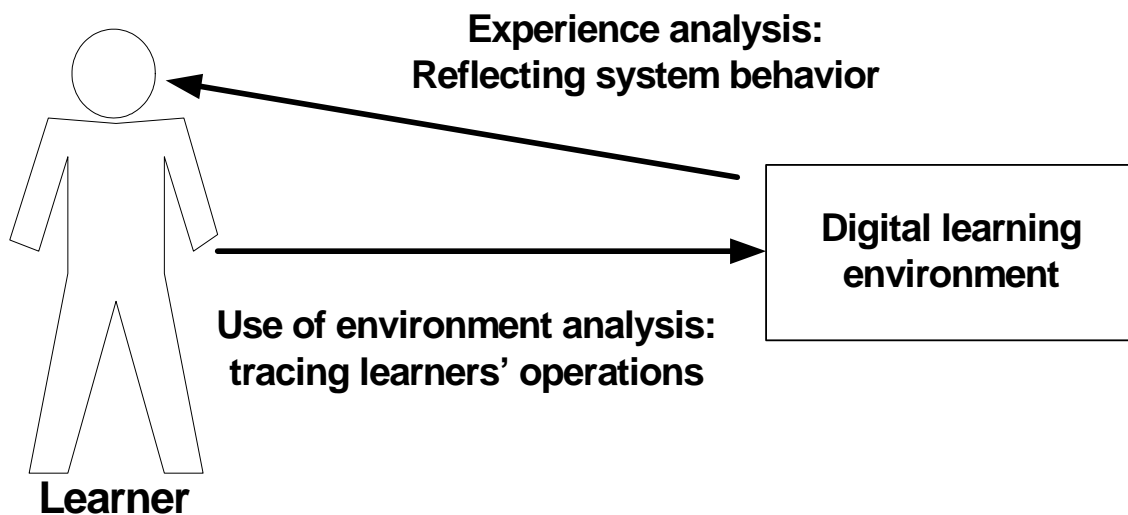


Figure 4.2: Experience and use of environment analysis in the formative evaluation component

Use of environment analysis is often related to the features of the environment. Experience analysis can include more general evaluative methods. For instance, learners can be interviewed to find out their opinions. In comment evaluation, the learners are provided with a space to express their ideas and give feedback. Content analysis can be used to evaluate any textual data related to both analyses (Cohen & Manion, 1989; Hara et al., 2000). A viable solution is to apply an emergent scheme where the content categories for the analysis derive from the data (Anfara et al., 2002; Stemler, 2001). Table 4.1 summarizes the features of the three FODEM components with examples of the main tasks, possible methods, outcomes and risks of each component.

Table 4.1: Summary of the FODEM components

	NA	I	FE
Tasks	Identify the design solutions and main concepts (Kerne, 2002).	Implement the design solutions fast to enable an early experiment with learners (Quintana et al., 2002).	Evaluate the use of environment to find out viable features (Norman, 2002).
Methods	Analysis of contextual factors (Fallman, 2003), learning theories (Gredler, 1997) and evaluation of the information received from other threads.	Fast prototyping (Bahn & Nauman, 1997).	Use of environment, and experience analysis; content analysis.
Outcome	Pedagogical and technical design principles and solutions.	Environment that is usable in authentic learning settings (Brush & Saye, 2001).	Information about the features of the environment (Heiskanen & Newman, 1997).
Risks	Incorporate the design ideas from different origins in a meaningful way (Abdelraheem, 2003).	Exposing too early to users (Wixon, 1995).	Break the structure of the environment (Kommers, 2004)

4.2.2 Threads as a dialectical structure

A *thread* is formed by intensively and dialectically interacting components. A line visualizes the interaction within the thread (Figure 4.3). A thread can also include several instances of components. For instance, a formative evaluation component can include several studies about the use of the environment (Figure 4.3).

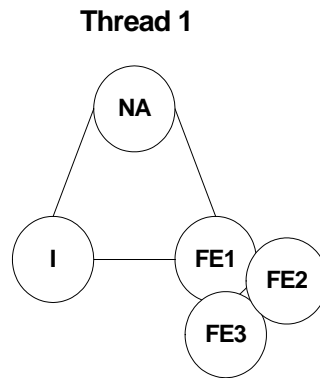


Figure 4.3: Multiple components in a thread

Threads can be activated in parallel; they are not analogous to the version-based model of system development. However, in some cases, threads can present incremental versions of the environment. Different threads can be used to develop the environment from different perspectives. For instance, a thread represents the development from learners' perspective while another thread is focused on the instructor side. The work load in the development process can also be divided among the threads. In online study program development, for instance, threads can present the development of different courses or digital learning tools related to the program.

Each thread has a certain development *theme* which aligns the goals of a thread and joins the components together. A person or persons can be assigned to a thread or a single component. Thread types can also be identified. An example of a thread type is a *cycle of development*. In a cycle of development, the components within a thread have an iterative dependency, and the development progresses according to the theme of the thread. In the cycle of development, sub-threads re-presenting the iteration cycles can be identified. Figure 4.4 illustrates the symbols used to express the threads, interactions and dependencies. Thread 1 is a cycle of development with a set of iterations. Thread 2 focuses on the specific aspects of the development process based on the experiences from the cycle of development thread.

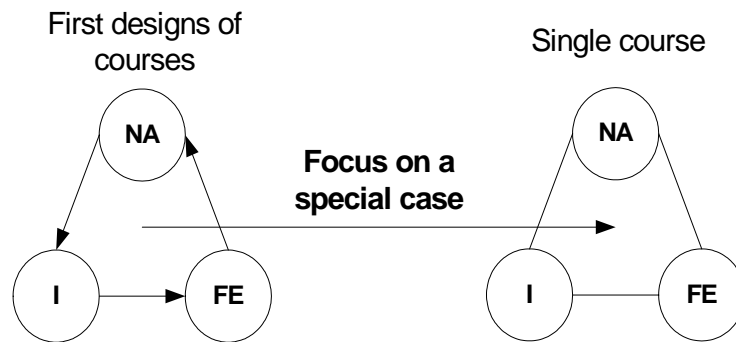


Figure 4.4: A possible FODEM scenario with two threads and a thread dependency

4.2.3 Dependencies

Dependencies represent the interdependent structure of threads, the interaction among components (cycle of development) and the relations between components in different threads. The most common dependency is between individual components in different threads. A *name* of the dependency presents the nature and meaning of the interaction. A *direction* of the dependency presents the flow of the interaction, and it is visualized with an arrow showing the direction. There can be one-directional or bi-directional dependencies. A bi-directional dependency represents a mutual relationship between two components or threads. A *thread dependency* between threads shows that a whole thread is initiated by another thread. In this case, it can be difficult to distinguish relations between single components because there are several interactions among the threads. A thread dependency often exists between a cycle of development and a normal thread because a cycle of development thread usually produces lot of information to the development process.

Figure 4.5 illustrates a one-directional dependency between formative evaluation and needs analysis components in two threads. The needs analysis in Thread 2 has produced information for the formative evaluation in Thread 1. This is the case, for instance, when a literature review or another type of analysis in needs analysis produces information to support the evaluation.

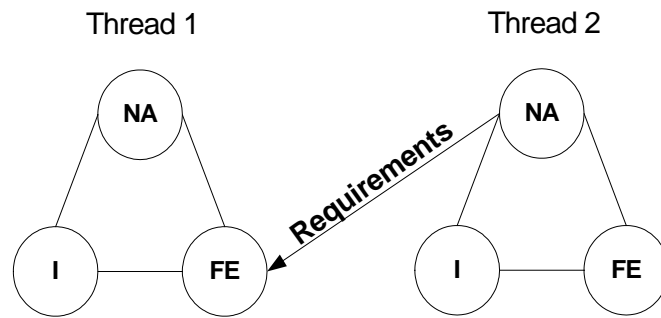


Figure 4.5: Example of a one-directional dependency

An example of a bi-directional dependency is between the formative evaluation components in different threads (Figure 4.6). Research results of the evaluations can be compared or the research methods can be altered depending on the experiences.

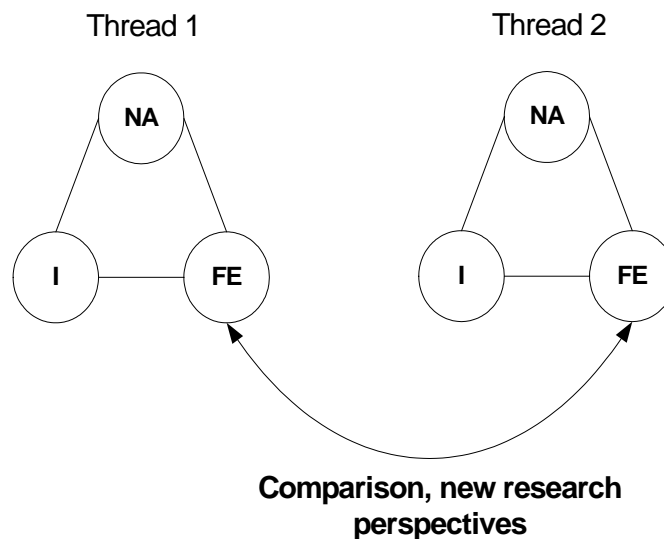


Figure 4.6: An example of a bi-directional dependency

4.2.4 Multi-layered threads and dependencies

When the development process is large enough, a thread can be divided into a set of sub-threads. In principle, many layers of threads can exist. If a sub-thread becomes large enough it can be detached into a separate development process. Figure 4.7 visualizes a multi-layered thread structure.

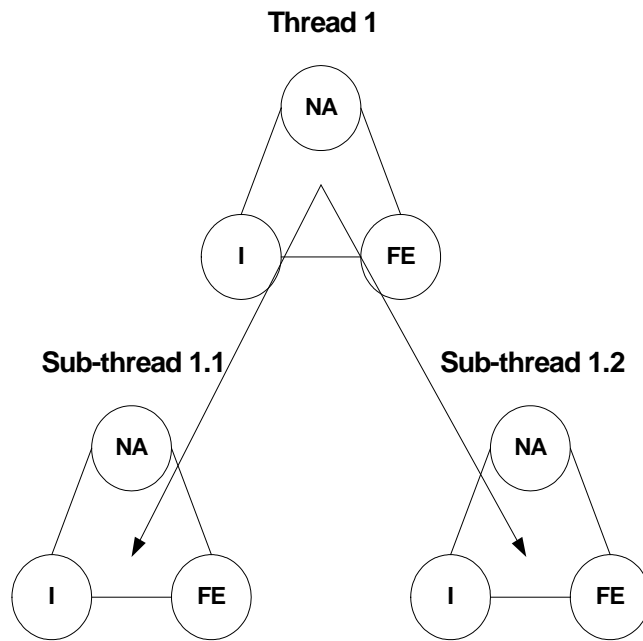


Figure 4.7: Multi-layered thread structure

The dependencies in multi-layered threads exist between different thread layers and sister sub-threads. Figure 4.8 visualizes a possible scenario of multi-layered dependencies.

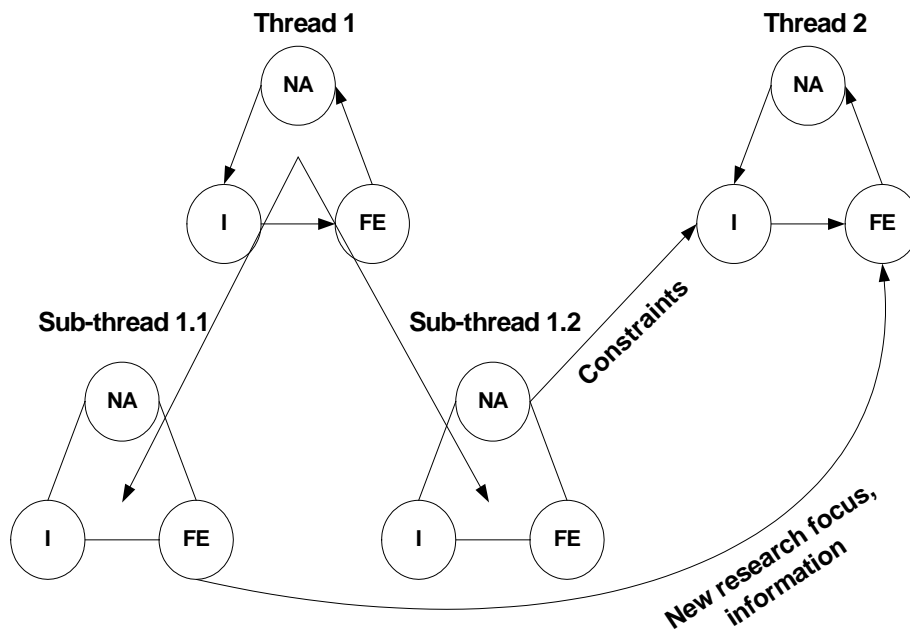


Figure 4.8: Dependencies in a multi-layered structure

4.3 Technical issues

The multithreaded structure of FODEM can be described with metadata elements to model the development process (Wiley, 2001). Table 4.2 shows possible metadata elements related to a thread. Some elements are based on the IMS (Instructional Management System) metadata description (Anderson & Wason, 2001). A similar metadata description can be created for the process, component and dependency parts of the method.

Table 4.2: Metadata description for a thread

Element	Fields	Explanation
General	- threadId - theme - description	General information about the thread. ThreadId identifies each thread uniquely
Lifecycle	- responsible_persons - creation_date - end_date	Information on the life of a thread
Type	- name	Type of a thread (i.e. loop of improvement)
Relation	- has_subthread - is_subthread - is_dependent - has_dependency	Relation of the thread to other threads
Components	- component_Id_list	List of components within the thread

The metadata description of FODEM allows that the development process can be modeled. The metadata modeling enables, for instance, in implementing software applications to support the design process. FODEM could be combined with applications, such as Woven Stories (WS), to build an environment to manage the process (Gerd et al., 2001). A WS based design environment could provide a visual representation of the development process. The environment could also include built-in procedures to help designers working with the FODEM components. The application could be used to store and distribute documents, research results, and analysis data in threads and components. Tacit knowledge related to the development process could also be stored. Finally, a web-based WS application could help designers to collaboratively work with each other's ideas and problems in different threads. The application would also allow a collaborative management of the development process.

4.4 Remarks on the feasibility of FODEM

Fast implementation and formative evaluation components allow using FODEM for gradually developing the digital learning environment. The design starts with the most important features. The development process itself refines the environment, for instance, according to needs of the learners. FODEM also enables the designers to explore with novel ideas and design solutions. In FODEM, threads and components can be conducted in various ways depending on the design context. The goal is to both introduce some guidelines and leave as much openness as possible.

FODEM can be related to existing digital learning environment design methods. The *system development life cycle* (SDLC) methods are based on de-contextualized procedures and phases (Sommerville, 1998; Whitten et al., 2004). The difference to FODEM is that SDLC methods do not include built-in procedures to consider the needs of sparse learning communities. The problem with many SDLC methods is long development times (Dennis & Wixom, 2003). Furthermore, users are often only involved at testing the final environment (Hoffer et al., 1999). Newer SDLC methods, such as fast prototyping and spiral development, incorporate the user in the development process early on (Kommers, 2004; Whitten et al., 2004). Different versions are developed iteratively, however, compared to FODEM the dependencies are rarely modeled between the versions.

The various *context design* (CD) methods emphasize the importance of identifying the problems and needs of the design context (Norman, 2002; Wixon et al., 1990). For instance, stories of experience, interviews and user observations are used to define the functions of an environment (Bayer & Holtzblatt, 1998; Bayer & Holtzblatt, 1999). Various CD approaches also emphasize the creation of many parallel ideas and concepts to identify the requirements for the system (Löwgren, 1995). Sketching, for instance, is used to represent the design ideas to users (Fallman, 2003). CD methods are applicable in needs analysis and formative evaluation components in FODEM. The main difference between CD methods and FODEM is that CD methods do not emphasize the formative evaluation process. Although CD methods focus on the users' needs, they are still meant as general guidelines for any software development.

The three methods above originate from the computing field. However, digital learning environment design methods also exist in the educational technology field.

The focus on *instructional design* (ID) is on the pedagogical side of the development process (Moallem, 2001). Contemporary ID methods stress, for instance, learner-centered approaches, and formative development (McKeanna & Laycock, 2004; Moonen, 2002). Compared to FODEM, ID methods include recommendations and rules on features of a functional digital learning environment (Jonassen, 1999; Lowyck, 2002). The rules often follow, for instance, a certain learning or instructional theory (Wilson, 1997). Therefore, the methods themselves constraint what types of environments are developed. The FODEM method is not tied to any particular theory. Many ID methods are also based on rigid approaches, which require substantial resource investments (Heydenrych, 2004; Moonen, 2002).

Design research (DR) includes a series of approaches for producing technical designs for learning (Barab & Squire, 2004). The goal in DR is not just to develop the environment, but also to create and test theoretical models of human behavior (Juuti, 2005). DR is very close to action research. Both action research and DR require the researcher to operate in a kind of a dual mode, that of research and design. Furthermore, in DR the development process is often cyclical in nature (Design-Based Research Collective, 2003). A distinctive nature of DR is to test the developed environments and theories in authentic settings (Myers, 1999). The difference to FODEM is that the aim in FODEM is not to validate or test any theories on human behaviour.

Common to all approaches presented above are *specification* (S), *implementation* (I) and *evaluation* (E) phases. The specification phase is used to determine the needs, requirements and expectations for the solutions. In the implementation phases, the solution is implemented. Finally, the evaluation phase is used to evaluate the implementation. Table 4.3 summarizes how the different ideas (marked in bold) from the sister methods relate to FODEM.

Table 4.3: Sister methods in relation to FODEM

	SDLC	CD	ID	DR
^S	De - contextualized	Users needs	Learner-centered design	Practical knowledge
^I	Fast prototyping. sequential versions, structured approaches	Sketching	SDLC methods	Embedded in use
^E	Summative, phased development	Stories of experience, actual observations	Experimenting, reflection, formative evaluation	Authentic settings

The uniqueness of FODEM compared to its sister methods is the way it models the parallel, but interdependent threads of the digital learning environment development. Furthermore, the modeling of the FODEM structure, as explained in previous section, enables the implementation of technical solutions to support the development process.

5 LEAP – digital learning tool

In this chapter, the aspects of LEAP are presented that have not changed during the development. First, a short introduction to the main concepts is given with a concrete example on how to apply LEAP. Subsequently, the pedagogical services that depict the support given to learners are presented. The pedagogical services are formed on the pedagogical, technical, application, and platform adaptation functional layers. The pedagogical, technical and a part of the application layer are presented in this chapter.

5.1 Introduction to the main concepts

5.1.1 Basic structure of LEAP

LEAP (LEArning Process companion) is a web-based client-server application. There are five basic concepts in LEAP: episode, phase, step, solution and problem solving support activities. The function of LEAP is based on a learning or problem solving *episode* (Suhonen & Sutinen, 2002). There can be several episodes activated at the same time. Typically an episode is a course, but it can also be a smaller or bigger entity, such as a task inside a course or a study program.

An episode can be divided into a set of phases. A *phase* is a distinct task or a time period inside an episode. A phase can also be a clearly defined part of an episode, like a phase in a project course. Phases are defined by the instructor and they can be added, removed or modified during the use of the tool. A phase includes an *instruction object* for distributing comments, hints or instructions. Each phase also includes a set of *user text fields* that learners are supposed to fill in relation to a phase.

Learners can add any number of *steps* to a phase. A step can be anything related to a phase, for instance, a learning event, a problem, an emotion, or a new perspective. Learners can use *step text fields* to describe the meaning of the step. Learners can make their steps public to their peers and comment on each other's

steps. Table 5.1 shows an example of an instruction object, and user and step text fields.

Table 5.1: Possible text fields in LEAP

Entry	Fields	Filled by:
Instruction object	Description, Goals, Important to Remember Note, Due Date	Instructor
User text	Goals & Plans, Progress of the Tasks, Success & Failure, Mood, Free Text	Learner
Step text	Statement, Description, Pondering, Free Text	Learner

A step can include any number of *solutions*. A solution can be an answer to a step or a new perspective. Learners can use *problem solving support activities*, namely Distance Analogies, Question Lists and BruteThink to process solutions. A more detail description of the problem solving support activities is given in section 5.2.3. Figure 5.1 illustrates the basic structure of LEAP.

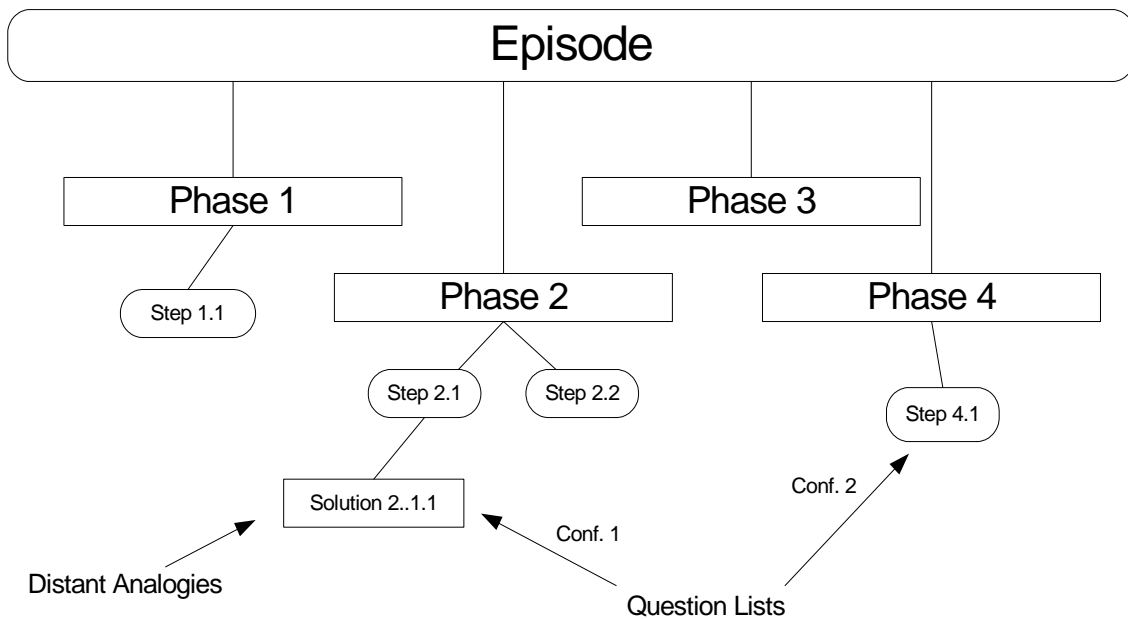


Figure 5.1: Basic structure of LEAP

5.1.2 Example of using LEAP in a Programming Project Course

In a project course, the episode can be the whole software development process. In this example, the instructor divides the course into planning, design and

implementation phases. An instruction object includes *Description*, *Goals of the phase* and *Tips and instructions* fields to direct the learners' thinking (Figure 5.2).

Intro	Project phases	History	Feedback	Own info	Help	Close LEAP
<div style="display: flex;"> <div style="background-color: #e0f2f1; padding: 5px; width: 20%;"> <p>Planning</p> <p>Status of the phase: in progress</p> <p>Own deadline: 3.2.2005</p> </div> <div style="padding: 5px; width: 80%;"> <p>Description</p> <p>In the planning phase, you should plan your project carefully. Try to define the functionality and features of the program at a higher level.</p> <p>Goals of the phase</p> <p>You should produce a project plan. The plan should include your goals (parts of the program and main functionality). The project plan could include preliminary sketches for objects, classes and methods. Project plan should not be too detailed, but it should give an overview of the different parts of the program and their relationship.</p> <p>Tips and instructions</p> <p>You can never emphasize planning phase too much. If this phase has not been done carefully, it will cost you later. You should reserve enough time and energy for planning the project.</p> </div> </div>						

Figure 5.2: Information object in a Programming Project Course

Steps are used in a project course to describe the problems, ideas or achievements related to the different phases. In Figure 5.3, a learner uses *Description* and *Meaning* fields to describe the step.

Intro	Project phases	History	Feedback	Own info	Help	Close LEAP
<p style="text-align: left;"><<Back to the phase info</p> <div style="display: flex;"> <div style="background-color: #e0f2f1; padding: 5px; width: 20%;"> <p>Feedback from the project plan</p> <p>Created: 21.12.2004 Modified: 21.12.2004</p> </div> <div style="padding: 5px; width: 80%;"> <p>Description</p> <p>We got feedback from the project plan. It said that our plan was not specific enough.</p> <p>Meaning</p> <p>We have to think how to improve the plan. I do not know whether we need to send a corrected plan to the university. We need to ask from the instructors.</p> <p style="text-align: right;"> Modify step info Remove step </p> </div> </div> <div style="background-color: #e0f2f1; padding: 5px; margin-top: 10px;"> <p>Solutions</p> <p>Add new solution</p> </div> <p style="font-size: small; margin-top: 10px;"> Topical in the course: <i>The course has started. Remember to submit a project plan in time.</i> 21.12.2004 15:28:21 </p>						

Figure 5.3: Example of a step in LEAP

Solutions in LEAP can be processed with available problem solving support activities. Figure 5.4 shows a question list related to the planning phase. Learners can use the provided questions to process the solution.

Process the step **Feedback from the project plan** from the perspective of the following questions. You can save the ideas to the text field in each question.

Planning

Can you reduce the plan?	<input type="text"/>
Does the plan include all the parts of the system?	<input type="text"/>
Have you looked at the plan from different perspective?	<input type="text"/>
Is this a good plan?	<input type="text"/>
What can you remove?	<input type="text"/>

Figure 5.4: Example of a question list in a programming project

Finally, learners can use a history function to browse all phases, steps and solutions in an episode (Figure 5.5).

Planning | **In progress** | 3.2.2005

Memory list

Free description

Steps | [Add new step](#)

[Feedback from the project plan](#) | [Add new solution](#)

Entry description: We got feedback from the project plan. It said that our plan was not specific enough.

Pondering: We have to think how to improve the plan. I do not know whether we need to send a corrected plan to the university. We need to ask from the instructors.

Created:21.12.2004
Modified:21.12.2004

Design | **Important** | -

Memory list

Free description

Steps | [Add new step](#)

[Constructing interface sketches](#) | [Add new solution](#)

Entry description: This time I made interface sketches to model the different parts of the program.

Pondering: Sketches helped me to start programming, and I was able to explain the functionality of the program more easily. Sketches also help to grasp the whole program.

Created:21.12.2004
Modified:21.12.2004

Figure 5.5: History page in LEAP

5.2 Functional layers for implementing the pedagogical services

Pedagogical services form the base of a digital learning tool; they depict the support given to learners. The pedagogical services in LEAP are the *digital learning portfolio*, *creative problem solving support*, and a *provocative agent*. Pedagogical services are implemented through a set of functional layers. Figure 5.6 depicts the layers in the form of a circle. The outermost layer, the platform adaptation layer, interacts directly with the user, while the innermost layers, the pedagogical and technical layers, are not necessarily directly visible to the user. A sector in the circle represents a pedagogical service; each service includes all the functional layers.

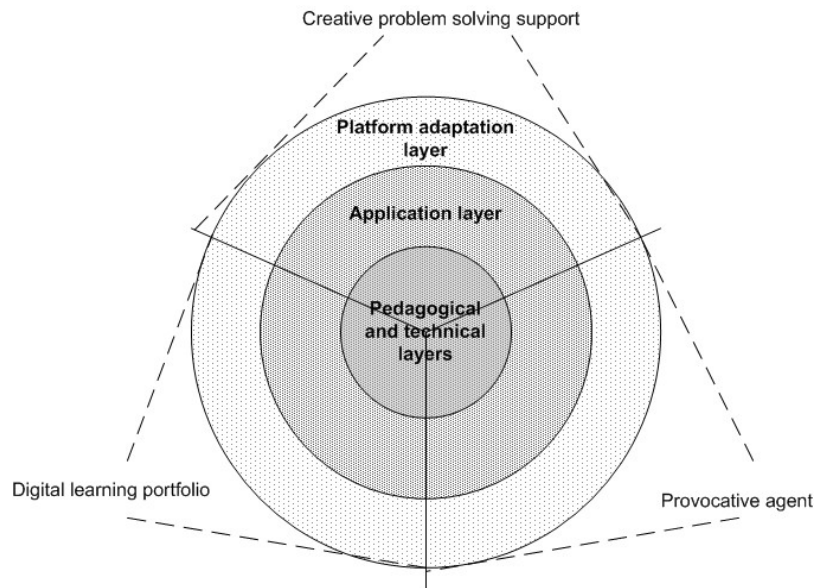


Figure 5.6: Functional layers to implement the pedagogical services in LEAP

The *pedagogical layer* depicts the pedagogical design principles and ideas for each service (Meisalo et al., 2003a). Without appropriate pedagogical solutions, a digital learning tool becomes almost useless. The *technical layer* depicts the technical design principles, requirements and constraints for the pedagogical services. The implementation of the pedagogical services should follow the technical design principles. The *application layer* is used to implement the pedagogical services based on the solutions posed by the pedagogical and technical layers. The application layer is used, for instance, for implementing the interaction with the learner; it will ultimately control how the system will be used. The *platform adaptation layer* adapts the services provided by the inner layers to the technology used by users, such as mobile devices. In the platform adaptation layer, all

requirements and constraints depicted by the inner layers must be taken into consideration. A thread in LEAP development, mobile adaptation extension, includes the first concept designs towards implementing the platform adaptation layer in LEAP.

5.2.1 The pedagogical layer in LEAP

The pedagogical layer of *digital learning portfolio* service originates both from theory and practice. In general, digital learning portfolios are applied to support learners' metacognitive processing (Fetherston, 2001; Pintrich, 2002). A common task of a digital learning portfolio is to support critical reflection (Avraamidou & Zembal-Saul, 2002; Brockbank & McGill, 1998; Chang, 2001; Herrington & Oliver, 2002). Learners can use a digital learning portfolio, for instance, to identify their learning goals, prior knowledge, and personal interests (Gathercoal et al., 2002; Masui & De Corte, 1999). Different studies have indicated positive results on using learning portfolios (Avraamidou & Zembal-Saul, 2002; McCrindle & Christensen 1995). The research has shown that many learners noted personal meaning, satisfaction and sense of accomplishment from creating individual portfolios. Furthermore, Zimmerman (2002) argues that by asking learners to record their learning experiences leads to improvement in behavior and thinking. Many examples exist on digital learning portfolios applied in a certain learning domain (Chang, 2001; Estell, 2001; Loh et al., 1998; Sugiyama et al., 2002).

The theoretical aspects of digital learning portfolios merged to the practical needs in the ViSCoS Programming Project Course. It was discovered that students needed support in managing the project. First, the programming project is a demanding task, where students need to apply their programming skills in a concrete small-scale software development. One of the learning goals of the course is to get students to understand the purpose and requirements of different tasks in the project. Important skills are management of the project, time planning and prioritization of the tasks. Secondly, the programming courses in ViSCoS prior to the Programming Project Course have only small programming tasks. In the Programming Project Course, students need to implement larger programs. Students should be able to relate this new situation to their previous experiences. They should realize what they know and what skills they have. If the project requires new knowledge and skills, students should independently learn these skills. Finally, the programming project is

open in nature. Students have a freedom to create any type of applications. Hence, a goal of the course is to get students to realize the possibilities of programming.

Table 5.2 shows how the goals of the Programming Project Course influenced the creation of the main concepts in LEAP. Relevant problems in learning software design, such as starting coding before understanding the real problems and the lack of design patterns, were also considered in the concept design of LEAP (McCracken, 2004).

Table 5.2: Needs from the Programming Project Course

Goal	Challenge	LEAP support feature
Requirements of a project	Students need to understand the meaning and relationship of the phases in the project	Basic phase-based structure of LEAP
Apply the programming skills in a concrete project	Reflection and evaluation	Text fields in phases, steps and solutions
Management of a project	Time planning, prioritization, management of the tasks and duties	Deadlines, stages of the phases
Opportunities for programming	Novel solutions in projects. Opportunities to create own applications.	Creative problem solving support activities

The first concrete support step in the Programming Project Course was a semi-structured paper-based programming portfolio created in the year 2002. The experiences in the paper-based portfolio served as a source of inspiration for the pedagogical layer of the digital learning portfolio service. The paper-based portfolio consisted of three A3 size sheet of papers. Every page had four frames, each with a theme of its own.

The paper-based version was used in the Programming Project Course from April 2002 to August 2002. Appendix E includes examples of the content extracted in paper-based portfolios from the course. The analysis of the students' contributions focused on evaluating what type of information was included in the paper-based portfolios (Suhonen & Sutinen, 2003). Furthermore, a level of contribution analysis was conducted to evaluate how much text the portfolios included. The analysis indicated that the paper-based portfolio could include valuable information about students' learning processes. The amount of low contributions indicated that there

was room for improving the concept. Either students did not understand the purpose of the portfolio or the portfolio had been totally forgotten during the course. Furthermore, only a few portfolios included metacognitive processing, such as reflection or analysis of the learning processes. It was also noticed that many students had modest goals for the project. Some of them did not even want to suggest a topic of their own. They did not relate their own weaknesses and interests to the learning goals in the course. Moreover, several students wrote that they did not get any new ideas during the course. The following ideas merged to development of LEAP

- i) A semi-structured portfolio should give relevant information about students' learning process.
- ii) The digital portfolio should help students in describing the learning process more deeply. The portfolio should also commit students to the learning process.
- iii) There was a need for methods that would spark students' imagination and creativity so that the projects would include novel ideas, not just copied ones.

The pedagogical layer of *creative problem solving support* service also originates from theory and practice. Creative problem solving support can be useful in supporting *open problem* solving (Jonassen, 1997; Ward et al., 1995). Open problem solving also requires that the problem is processed creatively from different perspectives (Brickel et al., 2002; Shneiderman, 2000).

The creative problem solving support service is related to the pedagogical design of LEAP because learning can be seen as a problem solving-process. The goal is to proceed from an undesired situation to the desired learning goal. During a learning process a learner might come up with time management or other types of problems. The problems can also be seen as challenges. Learning can include frustration, anger or other types of strong feelings. An important factor is to constructively deal with problems and emotions related to the learning process. Creative problem solving support could also be used to expand learners' thinking from a conventional track. Creative problem solving support is also related to the Programming Project Course context. A project can be seen as an open design problem. If the solution is expected to be effective and efficient, a considerable amount of problem solving must be

conducted (Deek et al., 1999). Students could also be supported to be more open towards unconventional ideas (Sutinen & Tarhio, 2001).

The concrete ideas for the pedagogical layer of the creative problem solving support service emerged from the concept of *Problem Management Assistant* (PMA). PMA concept includes tools for supporting open-ended problem solving in a web-based environment (Rautama et al., 2001). The emphasis in PMA is on managing the entire problem solving process from specification up to evaluation, rather than searching for the optimal solution. Table 5.3 includes a comparison of LEAP and PMA features.

Table 5.3: Features of PMA and LEAP

Feature	PMA	LEAP
Customized process control	X	-
Episode-phase structure	X	X
Distant thinking models	X	X
Close pages	X	X
Question lists	X	X
Animated conceptualization of the problem process	X	-
Support pages	X	X
Rating of support	X	-
Success stories	X	-
Public/private entries	-	X
BruteThink	-	X
Digital learning portfolio	-	X

The pedagogical layer of *provocative agent* service consists of two main concepts: jagged study zone (JSZ) and committing learners to the learning process. A provocative agent could, for instance, support learners in solving fuzzy problems and deal with unexpected situations (Gerdt et al., 2002). A provocative agent could also include inspiration points in a form of video clips, sounds, and figures to provide fresh ideas or provocation to learners (Nulden & Scheepers, 1999). The JSZ concept depicts that learners' need adaptive support. The provocative agent could operate on the zone of proximal development (ZPD) by providing a temporary support for learners to perform at the level just beyond their current ability (Kurahila, 2003a; Lowyck, 2002; Vygotsky, 1978).

A provocative agent could also be used to commit the learners to the learning process. Especially in an online learning situation, the instructor is transparent, intervening indirectly (Polsani, 2003b). The commitment could be achieved by so

called *slow technology* that is not necessarily easy or fast to use; rather, it can take time to learn how the technology works (Hallnäs & Redström, 2001). The user does not necessarily recognize the meaning and purpose of the slow technology instantly. Fast technology, on the contrary, is easy to use, efficient, and should produce concrete results. Figure 5.7 shows the differences between the fast and slow technology.

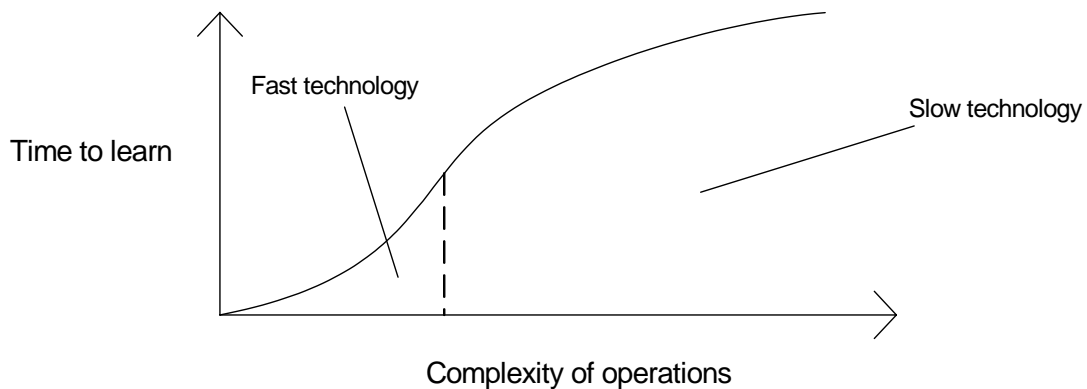


Figure 5.7: Fast vs. slow technology

Slow technology can be used, for instance, to ensure that learners are aware of the activities and meaning of the environment (Quintana et al., 2002). Figure 5.8 summarizes the concepts in the pedagogical layer of LEAP.

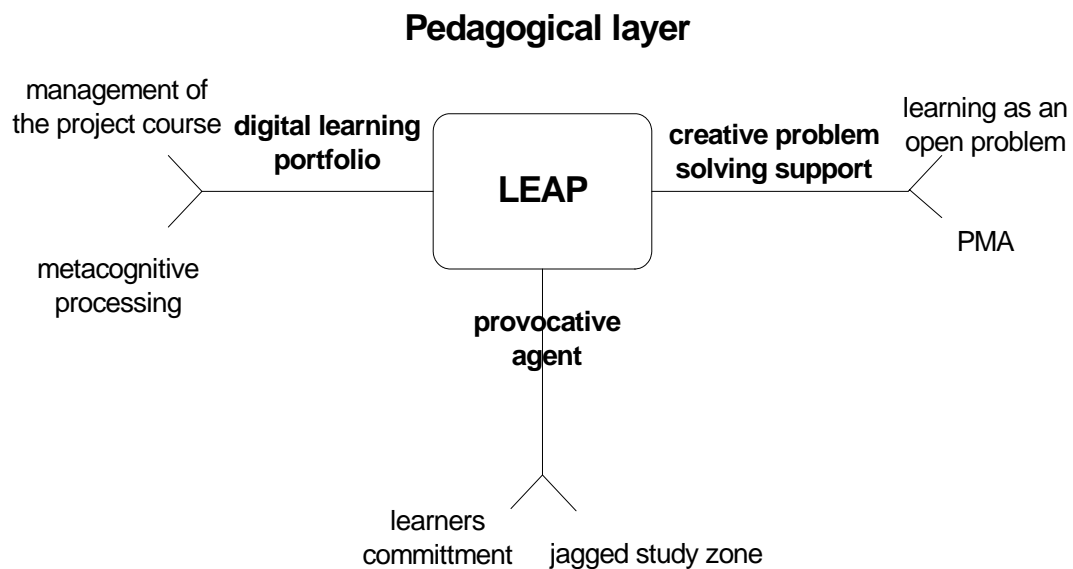


Figure 5.8: Pedagogical layer of LEAP

5.2.2 The technical layer of LEAP

The technical layer depicts the technical design principles that should be followed when implementing the pedagogical services. The *easy accessibility* principle requires that LEAP should be widely accessible (Bergman, 1999). Users should not need to install any extra plug-ins to use the tool. Hence, all LEAP versions are web-based, client-server multi-user applications (Kainulainen, 2004). LEAP should also be accessible with a diverse set of devices and platforms; the goal is platform independency (Jones et al., 1999).

The second technical design principle is an *open and extensible architecture*. *Generic tools* are usable within different learning settings and domains (Brickel et al., 2002; Pitkänen & Silander, 2004). They are not tied, for instance, to a specific learning theory or a pedagogical model. Concept mapping tools are an example of a generic tool (Stoyanova & Kommers, 2002). *Context-specific* tools are designed for a particular context, task or subject domain (Pitkänen & Silander, 2004). In many cases, one cannot use them in another domain. The advantage of context-specific tools is that they can be tuned to the requirements, constraints and needs of the context. The Jeliot family program visualization tools are an example of a context-specific tool (Ben-Ari et al., 2002). A disadvantage of a context-specific tool is the lack of re-usability. The aim in LEAP development has been to develop a tool that was not bound to a single learning context. In an ideal situation, the tool could be both open to major modifications and include features that would consider the special requirements in a certain context. Naturally, the core function of the tool poses restrictions, but inside these constraints, the tool could be modifiable towards different use scenarios. Extensible architecture also means that the tool should be open for new features.

The third technical design principle has been to *incorporate the tool into learners' everyday context* naturally and appropriately (Abdelraheem, 2003; Moonen, 2002; Schroeder, 2002; Thomas et al., 2002). If this fails, it is unlikely that the development results are successful. According to Storey et al. (2002), learners prefer convenient, flexible and easily accessible environments with simple navigation and well-designed layouts.

A concrete principle allowing the convenient use of LEAP is the light interface. The light interface has two main principles. First, the tool should be accessible and usable through a low bandwidth connection. Secondly, the interface of the tool should not increase the cognitive load of a learner. The learner should be able to concentrate on the main functionalities of the tool without spending too much time in learning to use the system (Bergman, 1999). Figure 5.9 summarizes the design principles of the technical layer in LEAP.

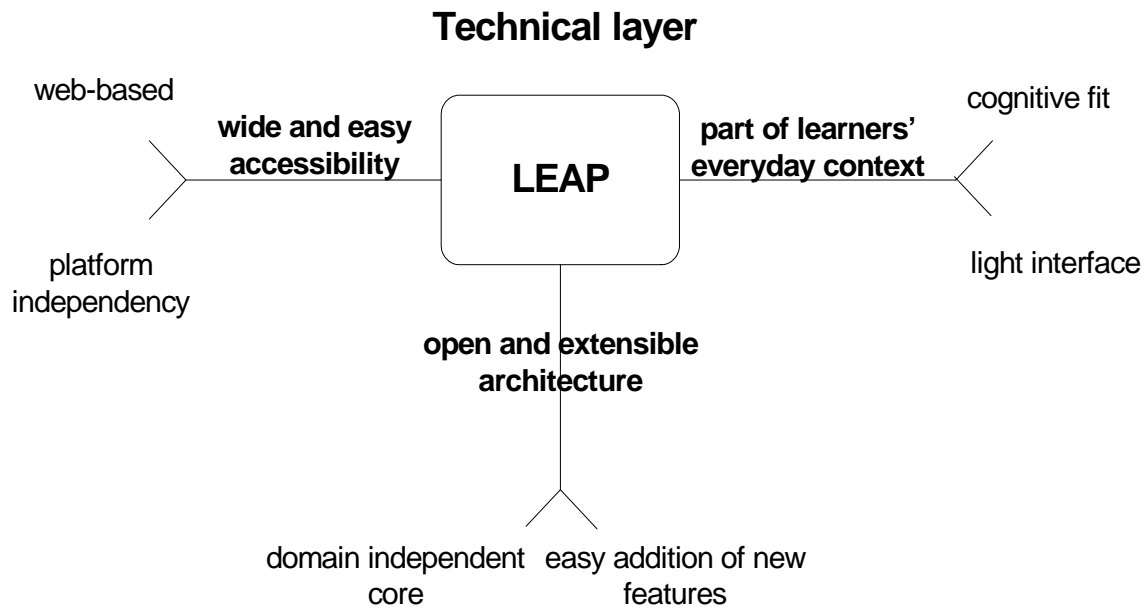


Figure 5.9: Technical layer of LEAP

5.2.3 Application layer for implementing the pedagogical services

The digital learning portfolio service is implemented through phases, steps and solutions. Phases can be used to help students in managing the learning episode. Learners are able to use user text fields in phases, step text fields and solutions, for instance, in reflecting on their learning. The text fields create a framework for the digital learning portfolio. Text field labels direct learners' attention to those aspects that are considered to be important. Labels are modifiable, making the digital learning portfolio services usable in various learning situations.

The creative problem solving support service is implemented with problem solving support activities, that is, Distant Analogies, Question Lists and BruteThink. When a new solution is added to the tool, problem solving support activities can be used to

process the solution. Each activity has a unique configuration describing the behavior of the activity. One activity can have several configuration settings in different phases. In *Distant Analogies*, the tool provides learners a random web-page to associate (Kainulainen, 2004). In order to prevent the same page for being shown again, all pages that are shown to the user are stored.

The *Question Lists* activity follows a basic problem-solving method. The goal is to get learners to ponder their problems from various perspectives. For example, in developing a new idea further, one may ask: “can you expand it?”, “can you decrease it?” or “can you separate it?” (Rautama et al., 2001). A question list category consists of a set of questions. There are ready-made question lists in the tool. Furthermore, an instructor can create new question lists. A question list category includes at least one question list. A phase can include several question list categories, and one question list category can be assigned to several phases. Hence, an instructor can create different question list categories for different phases. The question list categories can be modified, removed and added by the instructor before and during the use of the tool.

BruteThink activity gives a learner a fixed word to work with. The word is picked from a wordlist. The instructor can create several different word lists to different phases. Learners should then relate their solution to the word provided by the BruteThink. BruteThink allows a student to skip the given word and ask for another to get new insight. The provocative agent pedagogical service has not yet been implemented.

As depicted by the technical layer, the application layer of LEAP is open for modifications. The main task is to configure the pedagogical services depending on the context of the use. Both the digital learning portfolio and creative problem solving support services can be left out from the tool, if there is no need to use them in a given situation. Minor modifications can also be made to the basic functions of the tool. Table 5.4 summarizes the main administrator functions in LEAP.

Table 5.4: The main administrator functions

Function	Description
Management of students	Adding, removing and modifying student information.
Modification of interface	The interface of the tool can be modified. For instance, different graphical presentation can be created depending on the context of the use.
Creating the scenario	Deciding the number of episodes. Addition of phases and instruction objects related to phases. Deciding the active services.
Modifying problem solving support services	Problem solving support activities can be modified according to the requirements of phases. Activities can also be switched off.
Change of language and content of pages	The language of the tool can be changed. The administrator can change the content of the tool to meet the requirements of different use contexts.

6 FODEM in digital learning tool development – the case of LEAP

6.1 Overview

Three threads can be identified in the LEAP development process: (1) the first prototype, (2) a re-design of the technical architecture and (3) a mobile adaptation extension (Figure 6.1). Within the first thread, the first prototype was implemented based on the technical and pedagogical design ideas presented in Chapter 5. The first thread is also a starting point for later development. Based on the formative evaluations within Thread 1, the tool was modified in the second thread. The implementation of Thread 2 was also in accordance to the requirements from Thread 3, which includes only the needs analysis component of the mobile adaptation extension.

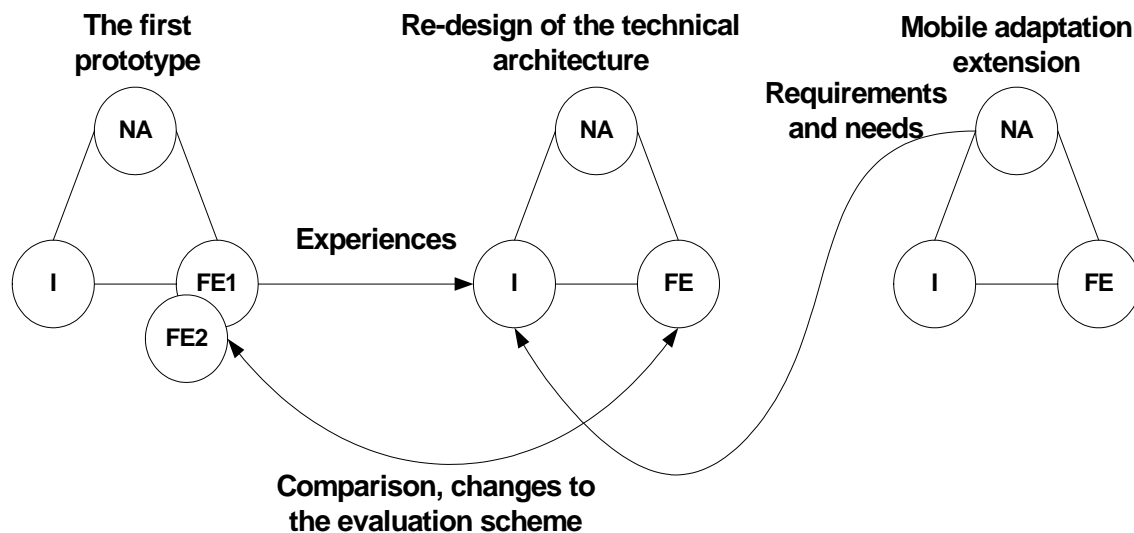


Figure 6.1: Development process of LEAP

Figure 6.2 visualizes the timeline of the LEAP development. The development started in year 2002. The last active component has been the formative evaluation component within the second thread.

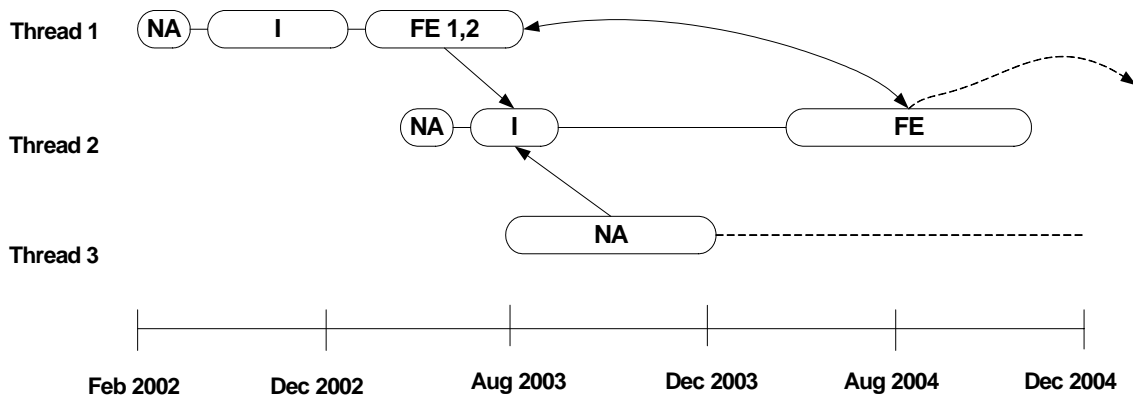


Figure 6.2: Timeline of LEAP development

The LEAP development includes three formative evaluation studies. Studies 1 and 2 are conducted in Thread 1, and Study 3 in Thread 2. In each study, the use of LEAP in authentic learning settings was evaluated. In Study 1, only the digital learning portfolio service was active. Studies 2 and 3 were conducted in a distance learning context, while Study 1 was conducted in the contact teaching course in the University of Joensuu, Finland. In Study 1, students used mainly English while in Studies 2 and 3 the language was Finnish. In the evaluation of Study 1, the examples from students' contributions are direct quotations with spelling and grammatical mistakes. Table 6.1 summarizes the study contexts in the LEAP development.

Table 6.1: Study contexts (ST=Study, TH=Thread)

ST	TH	Course	Time	No. students	Digital learning portfolio	Creative problem solving support
1	1	Problem Solving Course	January 2003 – early April 2003	19	x	-
2	1	ViSCoS Programming Project Course	mid-April 2003 – August 2003	22	x	x
3	2	ViSCoS Programming Project Course	April 2004 – August 2004	23	x	x

An evaluation scheme presented in Section 4.2.1 was used in all studies. The scheme had two main parts: the use of environment and experience analysis. The aim in the use of environment analysis was to evaluate how learners had used LEAP during the course. The aim in the experience analysis was to interpret how learners perceived or experienced the tool. Finally, both analyses aimed at revealing new development ideas for the tool. In order to understand the methods used, a

contribution concept needs to be defined. A *contribution* consists of user text fields, steps and solutions of one student. The *use of environment* analysis consist of three evaluations: general use, contribution and content. In the *general use evaluation*, numerical measures are calculated from students' contributions to LEAP. In the first part, the amounts of phases, steps and solutions were calculated. In the second part, the students' contributions were categorized according to the *amount of text*; low, medium, and high. In the *contribution evaluation*, students' contributions to LEAP were divided into three contribution categories: basic, intermediate and deep. The contribution categories show the depth of students' contributions. In the *content evaluation*, the content of students' contributions was analyzed more deeply using a content analysis method. Students' contributions were divided into content categories that emerged from the contributions.

Four different observation methods were used in the *experience analysis*. The aim of the *questionnaire* was to get information on students' opinions about the functionality and usefulness of LEAP. *Interviews* were used to find out how students perceived the tool during the course. The aim was also to reveal concrete improvement ideas and different perspectives. The interviews were analyzed by categorizing the interview data based on the themes of the interviews. *Extra assignments* were given to students who had not used the tool during the course. Students were asked to describe the reasons for not using the tool and their opinions about the features of the tool. They were also asked to provide ideas for LEAP development. Finally, in *comment evaluation* students' contributions to LEAP were browsed through for concrete development ideas to LEAP. Figure 6.2 summarizes the evaluation scheme applied in Studies 1, 2 and 3.

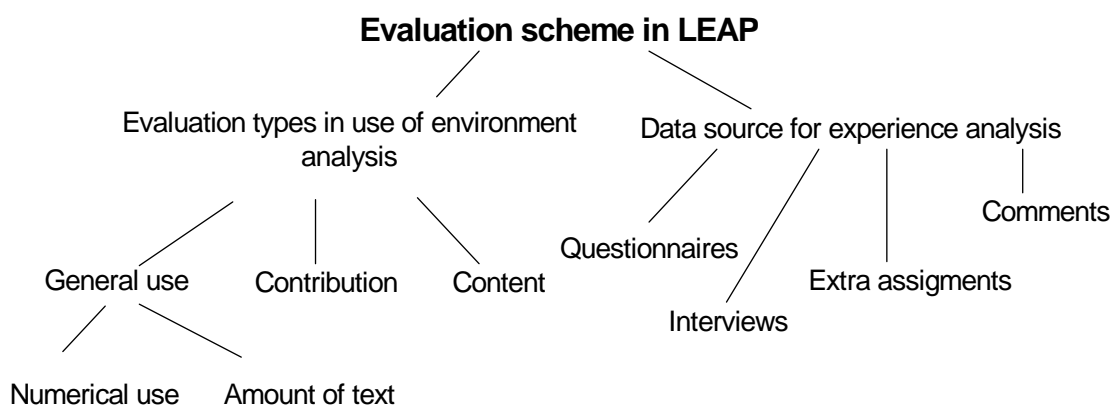


Figure 6.2: Applied evaluation scheme

Despite the differences of the two analyses, they share a common goal; to produce concrete knowledge and ideas for the development of the tool. The evaluation scheme was slightly varied in all studies. The use of environment analysis was similar in all studies (Table 6.2). The experience analysis was different in each study. This is because students were chosen differently and new evaluation methods were discovered during the course of development. In Study 1, an interview invitation (in Appendix A) was sent via email to all the students in the course, while in Studies 2 and 3 specific students were asked to take part in the interview. The students were chosen based on their contributions to LEAP. A questionnaire was sent to all students in Studies 2 and 3. Appendix B includes the questionnaire used in Study 2. Table 6.2 summarizes how the evaluation scheme was applied in Studies 1, 2 and 3. The number in parenthesis indicates the amount of contributions or students in each evaluation. For instance, in Study 1, seven students were interviewed. If there is no number, it means that all contributions are included.

Table 6.2: Summary of evaluations in LEAP development (ST=Study)

ST	Use of environment analysis			Experience analysis			
	General use evaluation	Contribution evaluation	Content evaluation	Interview evaluation	Questionnaire evaluation	Comment evaluation	Extra assignments
1	x	x	x (10)	x (7)		x	
2	x	x (10)	x (10)	x (6)	x (12)		x (3)
3	x	x	x (16)	x (3)	x (8)		

6.2 Thread 1: The first prototype

6.2.1 Needs analysis: Pedagogical and technical layers

The needs analysis in Thread 1 focused on defining the pedagogical and technical layers of LEAP presented in Section 5.2 (Suhonen & Sutinen, 2002).

6.2.2 Implementation: Application layer for implementing the pedagogical services

The pedagogical services were implemented within the first prototype with a WebMacro template framework (Kainulainen, 2004; WebMacro, 2004). WebMacro is built on MVC logic (Model-View-Controller). MVC separates the application

presentation (the view) from the application logic (the model). The model was implemented with Java classes. The view was constructed with WebMacro template files. The controller was used to control HTTP-requests and web-server tasks, such as sessions and database connections (Butler, 2001). The WebMacro architecture was chosen because it was easy to implement and simple to learn and use (Kainulainen, 2004).

WebMacro implementation enabled the division of the function of the tool into a set of pages. Each page had a certain role in the tool. Furthermore, each page was divided into content and activity. The content describes the physical structure of a page, and it consists of normal HTML elements like tables, images and forms. The *activity* is the task, function or purpose of a page. Inside one page there can be several activities. Each page had a corresponding WebMacro template file that depicts the content and activity of a page. The basic configurations of the tool could be modified by editing the WebMacro files. The structure of the pages follows the core structure of LEAP presented in Section 5.1 Table 6.3 illustrates the content and activity of the main pages in the first prototype.

Table 6.3: Content and activity of the main pages in the first prototype

Page	Content	Activity
Process	List of phases and basic information of each phase in an episode.	Learners see the entire episode consisting of phases. They can navigate to the Phase page for more information about a certain phase.
History	List of all phases, steps and solutions in an episode.	Learners can browse through all the information related to an episode.
Phase	General information related to a phase. An instruction object of the phase. User text fields to add text related to a phase. List of steps in a phase.	Learners can process the information in a phase. They can add text into the user text fields. Learners can set the status of a phase, add new steps or remove existing steps.
Step	Information related to a step: date of creation, date of modification, status of a step. Step text fields including a learner specified information related to a step. List of all solutions in a step.	Learners can remove steps or modify step text fields. They can also add new solutions to a step.
Public steps	List of all public steps in an episode. Text fields for commenting.	Learners see all the public steps in an episode. They can also send comments to other learners via e-mail.
Solution	Information related to a solution: creation date, available problem solving support activities, solution information in a single text field.	Learners can add new solutions. They can also process the solution with available problem solving support activities.

6.2.3 Formative evaluation: Studies 1 and 2

Within the formative evaluation component in Thread 1, two studies were conducted to evaluate the first prototype. In Study 1, the tool was used in the contact Problem Solving Course. In Study 2, the tool was used in the ViSCoS Programming Project Course. The goal of Study 1 was to evaluate how LEAP would perform outside the original design context. Some of the worst bugs and usability issues discovered in Study 1 were fixed for Study 2 by editing the WebMacro template files. However, major changes were not made. The main goal in Study 2 was to test how LEAP

would perform in the original design context. A contribution in Studies 1 and 2 consists of user text fields, steps and solutions of one student.

Study 1: LEAP in the Problem Solving Course

In Study 1, LEAP was divided into six phases, each covering a two weeks period of the course. Table 6.4 presents the instruction object, user and step text fields in Study 1.

Table 6.4: Text field entries in Study 1

Name	Fields	Filled by
Instruction object	Description, Goals, Important to Remember Note, Due Date	Instructor
User text	Goals & Plans, Progress of the Tasks, Success & Failure, Mood, Free Text	Learner
Step text	Statement, Description, Pondering, Free Text	Learner

Evaluation scheme in Study 1

The evaluation scheme presented in Section 6.1 was applied to evaluate LEAP. Table 6.5 summarizes the use of environment analysis in Study 1. In the *content evaluation*, the content of 10 selected contributions was analyzed more deeply. The contributions in the content evaluation were chosen according to the contribution evaluation: two basic, three intermediate and five deep contributions. Table 6.5 also shows where the results of each evaluation are presented.

Table 6.5: Use of environment analysis in Study 1

	Aim	Data	Method	Results
General use evaluation	Evaluate the use of tool in a general level	All 19 contributions	Numerical use: amount of phases, steps, public steps and solutions. Categorization of contributions according to the amount of text: low, medium and high.	Table 6.7 Table 6.11
Contribution evaluation	General picture of students' contributions to LEAP	All 19 contributions	Categorization of contributions to contribution categories: basic intermediate and deep	Table 6.12
Content evaluation	Content of students' contributions. What type of text did portfolios include?	10 contributions	Content analysis	Figure 6.3

Table 6.6 summarizes the experience analysis in Study 1. In the interview evaluation, an interview invitation was sent via email to all students in the course. The email included a sketch of the interview structure so the students could prepare for the interview. Appendix A includes the invitation (some typos are corrected later) and the interview sketch. Seven students agreed to be interviewed. The interview was an *open interview* (Cohen & Manion, 1989; Fincher & Petre, 2004). I followed the interview sketch, but the order and focus of questions changed depending on the students' responses. I conducted and recorded all the interviews in the premises of the Educational Technology Lab in the Department of Computer Science, University of Joensuu. The data was coded after the interview (Cohen et al., 2002). In the comment evaluation, the contributions were browsed through for concrete ideas to develop LEAP.

Table 6.6: Experience analysis in Study 1

	Aim	Data	Method	Results
Interview evaluation	Analyze how students perceived the tool. Find concrete improvement ideas.	Interview conducted with seven students.	Qualitative analysis of interviews	Tables 6.14, 6.15
Comment evaluation	Find out ideas on how to improve LEAP based on contributions.	All 19 contributions in LEAP	Browse through the contributions	Tables 6.16, 6.17

Results of the use of environment analysis in Study 1

General use evaluation

Table 6.7 shows the results of the numerical use evaluation. Phases indicate how many user text fields included text. *Sum* indicates the total number of phases, steps, public steps or solutions in all contributions. *Mean* specifies the average number of each variable per student. The last row *STDEV* shows the standard deviation of each variable.

Table 6.7: Numerical use data (N=19)

N=19	Phases	Steps	Public Steps	Solutions
Sum	91	302	45	26
Mean	4.8	15.9	2.4	1.4
STDEV	1.3	5.2	3.9	3.7

In average, 4.8 phases were filled in by a student, when the total number of phases was six. Furthermore, in average, every student added 15.9 steps. The number of public steps and reflections is much lower than the number of the phases and steps. *STDEV* value shows that there were variations among all the variables.

In the second part of the general use evaluation, the students' contributions were divided into three categories according to the amount of text, namely low, medium, or high. The *low* category included contributions that had minimal amount of text, for instance, most of the steps were only one or two sentences long. Many of the text fields were also empty. Table 6.8 shows a typical phase entry from the low category.

Table 6.8: Example of a low contribution

Statement	Description	Pondering
<i>“Problem”</i>	<i>“I guess our problem too wide and we need somehow limit it, because it is very philosophical question, and without any bounds it can not be solved”</i>	<i>“We will find some limits or applocation of our problem.”</i>

The *medium* category included contributions that had some long texts, but there were phases or steps that included just a few sentences. Table 6.9 shows an example of a medium contribution. Comments made by me are in brackets.

Table 6.9: Example of a medium contribution

Statement	<i>“The final representation;”</i>
Description	<i>“We have been given only limited amount of information about the final representation... so guy mentioned powerpoin... another said it should be only 15-minutes long.”</i>
Pondering	<i>“Hmm! I think somekind of graphical illustration device could be usefull to our topic. It is quite boring topic, so why not make it fun! that would be a challenge indeed.”</i>
Free Text	<i>“I suppose I could use Flash animation for implementation... I hope Erkki & Jarkko [instructors in the course] don't get defended easily... if I just could find their pictures somewhere”</i>

The *high* category included contributions that had extensive amount of text in almost all parts. Table 6.10 shows an example of a high category entry.

Table 6.10: Example of a high contribution

Free Text	<i>"It is interesting, but in this week i wrote 3 articles, and I feel i'm going to be a writer in the future. :)))"</i>
Mood	<i>"Our mood is not good, because we have an exam next week, and this week was really difficult for us. But we have many ineresting impressions, especialy some of them were got after film "Dead poet's society" [film was shown during the course], we had good time during this film, and we had some problems to think. But it wasn't so easy to write an article about this film, because you can see how people solve the problems, but it is hurd to estimate theirs way of problem decision and say how will you solve this problem."</i>
Progress of the Tasks	<i>"Hello, this week was searching week, we looked for the new ideas about different services for phone holders [this was the topic of the collaborative task]. And decided to it separately this week, everybody can use and apply his own methods and offer his ideas"</i>
Goals & Plans	<i>"Ok, my first progress was in Savonlinna, where i have been on Monday, i decided to solve my problem there by myself without any services. And i solved this problem successfully. You can read about my traveling to savonlinna in steps.</i>
Success & Failure	<i>"WE discussed about our success in our group, but now we have no any good idea how to increase our service, but our nxt step will be in our long session, we will look for any interesting information in Internet, in different forums about phone services people wishes."</i>

Table 6.11 shows the results of the amount of text evaluation.

Table 6.11: Categorization of students' contributions according to the amount of text (N=19)

Category	No. contributions
Low	4
Medium	6
High	9

Contribution evaluation

In the *contribution evaluation*, the contributions were divided into three categories: basic, intermediate and deep. The *basic* category included contributions that had mainly descriptive text about the events of the course. In a basic contribution, students did not reflect or describe about their personal feelings and perspectives. An intermediate contribution included deeper description about the

progress of the course than a basic contribution. An intermediate contribution included some text that could be characterized as metacognitive processing, such as reflection, and questioning. However, the amount of metacognitive text is still quite small. The last category, *deep*, included contributions that had thick descriptions about the events in the course. There were lot of questions, reflections, and examples of events related to the course. There was also analysis of the tasks students had encountered during the course. The deep contribution can be characterized as metacognitive processing. The contribution categories are not strict. There might be occasional deep description even in the basic category, but the overall information gained from the contributions decided the category. As seen in Table 6.12, most of the contributions in Study 1 were classified to intermediate or deep. There were three portfolios that included mainly descriptive text.

Table 6.12: Contribution evaluation in Study 1 (N=19)

Category	No. contributions
Basic	3
Intermediate	7
Deep	9

Content evaluation

In the *content evaluation*, ten contributions were analyzed for emerging themes. A *coding unit* is the smallest element in the content analysis. There were three coding units in the content evaluation of Study 1. The first unit was the user text fields in a phase. If a student had input text to six phases, there are six different units. The second unit is text fields in a step. The number of steps added by students determines the number of units. A solution entry is also a content unit. Four categories were identified. In the first category, *tasks in the course*, the theme was related to the tasks in the course. There were descriptions of meetings, problems, feelings and questions related to the tasks. Almost all of the tasks in the course were mentioned. The second category included *course-related text*. The contributions in the course-related text category did not include description about the specific tasks in the course, but they included more general descriptions about the course. These two categories can be characterized as descriptive writing where the contributions did not include reflection or metacognitive processing, but reports on events of the course (Herrington & Oliver, 2002).

The third category included text that can be characterized as *metacognitive awareness*. Issues like self-evaluation, reflection, time management and problem analysis could be found. The fourth category is about text that is unrelated to the course. It included themes that were not related directly to the content of the course. In the last category, the contributions included text about the portfolio tool itself. Table 6.13 shows what type of content each category included. There are also examples of contributions in each category.

Table 6.13: Content categories in Study 1

Name	Category	Topics	Examples
CC1	Tasks in the course	meetings, problems, progress, decisions, feelings, questions, group work issues, topic of the project, goals, tasks, plans, sessions, methods, workload, time tables, discussions, content of the project, ideas, group dynamics, presentation of the project, movie analysis, problem solving tool analysis, improvisation theatre, extra assignment	<i>“We are a bit stuck with our problem at the moment. Maybe George could help us. We have problems defining the scope of our problem. We don't know should we test any objects, or just living things, or just mammals, birds and so on.”</i>
CC2	Course-related text	description about the lectures, pondering, about the content of the course, feedback, improvement ideas for the course, who is the George?, comments about the lecturer	<i>“We had the nice lecture this day. It made a huge impression on me - some very nice methods of going out of your problem's dead end. It was really good! - a set of words, brainstorming, brainwriting...”</i>

Table 6.13: Content categories in Study 1 (cont.)

Name	Category	Topics	Examples
CC3	Metacognitive awareness	self-evaluation, improvement, reflection, time management, goals, mood, analysis, risk-taking, analyzing different ideas, management of the project tasks, expectations about the course, processing of the course content, questions, things to remember, importance of steps	<i>“Why we declined K.'s idea. A New energizer iz good idea, but very narrow, because our knowledge about different sources of energy is knowledge from the books which we were reading at school. So we can talk much time about it but we can't offer something new, many scientists were discussing about this theme, and I think they have offered already many ways of solutions, in this case our task is only to read about it and discuss why they declined some ideas.”</i>
CC4	Text unrelated to the course	trip, personal problems in life, other courses, purpose of education in general, studying, war in Iraq, courage to start an own business	<i>“We saw a very strange art exhibition in the Art musen in the Helsinki. It made an huge impression on me... it was so unusual... I can't say that I like it, but nevertheless it is a way of thinking...”</i>
CC5	Portfolio tool	meaning, use, functionality	<i>“I did not want to think about doing this portfolio stuff, I thought it is so difficult, odd, new, I don't know. I did not want to start this, but then I realized I must do it because it is required. And also, I can think it is for my own good to learn to write about my own ideas, feelings and so on. Maybe this will enhance my own thinking.”</i>

Figure 6.3 shows the *distribution of content categories* in the phases, steps and solutions of the selected ten portfolios. CC1 and CC3 categories were dominant in almost all sections. The only exception is that most of the solutions were categorized to CC5. Furthermore, *Task in the course* and *Metacognitive awareness* types of contributions comprised almost 80% of all contributions.

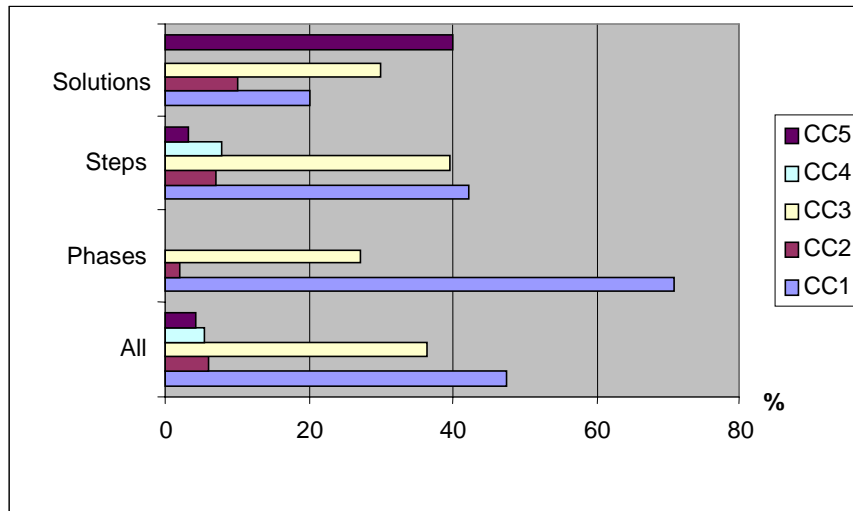


Figure 6.3: Content evaluation in Study 1

Results of the experience analysis in Study 1

Interview evaluation

In the first part of the experience analysis, the *interviews* conducted after the Problem Solving Course were evaluated using content analysis. Interviews were analyzed through the themes of the interview. Three themes were identified: the use of the tool, purpose of the tool, and development ideas. The *use of the tool* theme included students' opinions about the tool and its function during the course. The theme also included comments about problems related to the tool. The *purpose of the tool* theme included students' opinions about the purpose and usefulness of the tool. The last theme, *development ideas*, included students' suggestions on how to improve the tool. In the following categorization, the results of the analysis with each theme are presented. Details of the analysis are available (Suhonen, 2005). The first theme was the use of the tool during Study 1. Table 6.14 summarizes students' opinions and answers related to the theme.

Table 6.14: Students' opinions about the use of the tool

Issue	Opinions/answers
Meaning of the mood text field	Confusion about the meaning of the mood text field.
Public steps	Implementation of the public step activity was not good. There were, however, positive comments about the idea of the activity.
Restriction of the text fields/and static nature of the tool	Text fields in phases and steps are static, they do not change.
Static structure	Because the structure was not changing, it helped a student to write to LEAP.
Use time of the tool	- end part of the phase - after an interesting event has happened - after reviewing the earlier contributions

The second theme that emerged from the interviews was the purpose of the tool. According to students' opinions the following purposes for LEAP could be identified:

- As a source of feedback to instructors. Part of the assessment in the course;
- Starting point of thinking;
- Organization and writing of ideas during the course. Support for progress and reflection; and
- Information exchange.

The last theme identified in interviews was development ideas for LEAP. Three types of ideas could be identified. First, *cosmetic improvements*, such as interface or usability improvements. Secondly, *general improvements* to improve the current function of the tool, such as ideas to enhance the usefulness of the tool. Finally, *new features*, to add totally new functions or features to the tool. Table 6.15 presents the number of each development type identified in the interviews. Appendix D includes a list of development ideas that emerged from the interviews.

Table 6.15: Number of development idea types

Type	Number of ideas
Cosmetic improvements	5
General improvements	5
New features	2

Comment evaluation

In the second part of the experience analysis, students' contributions were *searched* for comments for improving LEAP. The comments found were categorized to the development types in a similar way as in the interview evaluation. Altogether five contributions included comments about LEAP. Table 6.16 presents the number of each development type in the comment evaluation. Appendix D includes a list of development ideas that emerged from the comments.

Table 6.16: Number of development ideas types from students' comments

Type	Number of ideas
Cosmetic improvements	6
General improvements	5
New features	2

Study 2: LEAP in the ViSCoS Programming Project Course

In Study 2, the phases in LEAP were divided into two episodes according to the submission cycle procedures in the course. Table 6.17 presents the phases in the two episodes.

Table 6.17: Phases in Study 2

Phases in the first submission cycle	
Topic selection	Documentation
Planning and project plan	Feedback and assessment
Implementation planning	Evaluation of the work and results after the first submission
Implementation/Coding	Last minute changes and panic
Testing	
Phases in the second submission cycle	
Planning of the possible changes	Evaluation of the work and results after the second submission
Implementation of the changes	Final assessment
Finishing up	

The text fields in LEAP were modified according to settings of the Programming Project Course. Table 6.18 presents the instruction object, user and step text fields in Study 2.

Table 6.18: Text field entries in Study 2

Entry	Text fields	Aim
Instruction object	Description Goals	Introduce the meaning and goals of each phase.
User text	Objectives Progress of the Tasks Success & Failure Mood Free Text	Learners could write down general issues related to a particular phase; how the tasks are progressing, possible success and failures, etc.
Step text	Statement Description Importance Related Solution Alternative	Learners are able to describe particular events during a phase.

Evaluation scheme in Study 2

The evaluation scheme in Study 2 was similar to the one in Study 1. The use of environment analysis was identical in Studies 1 and 2. The contribution and content evaluations in Study 2 included ten contributions from students who gave their permission for the study. Table 6.19 summarizes the *use of environment analysis* in Study 2.

Table 6.19: Use of environment analysis in Study 2

	Aim	Data	Method	Results
General use evaluation	Evaluate the use of the tool at a general level	23 contributions	Numerical use: number of phases, steps, public steps and solutions. Categorization of contributions according to the amount of text: low, medium and high.	Table 6.21 Table 6.23
Contribution evaluation	General picture of students' contributions to LEAP	Ten contributions	Categorization of contributions to contribution categories: basic intermediate and deep.	Table 6.24
Content evaluation	The content of students' contributions	Ten contributions	Content analysis	Figure 6.4

The *experience analysis* in Study 2 had three parts (Table 6.20). First, in the *questionnaire evaluation*, the questionnaire (Appendix B) about the use of LEAP was evaluated. Secondly, in the *interview evaluation*, the interviews conducted to five students were evaluated. The interview situation was open. The outline and main topics were specified in advance, but the flow of the interview changed depending on the responses of the interviewees. However, in Study 2 questions were not revealed to the students beforehand as in Study 1. Interviews were conducted and recorded in the premises of the Educational Technology Lab in the Department of Computer Science, University of Joensuu. Finally, *extra assignments* done by three students were evaluated. The aim was to find out different viewpoints of students' opinions about LEAP.

Table 6.20: Experience analysis in Study 2

	Aim	Data	Method	Results
Questionnaire evaluation	Students' opinions about the functionality and usefulness of LEAP	12 returned questionnaires	Numerical and qualitative analysis of questionnaires	Tables 6.26, 6.27, 6.28, 6.29
Interview evaluation	Analyze how students perceived the tool. Find concrete improvement ideas	Five interviews	Qualitative analysis of interviews	Tables 6.29, 6.30
Extra assignments	Analyze students' opinions about the tool. Concrete development ideas	Three assignments	Content analysis	Tables 6.31, 6.32

Results of the use of environment analysis in Study 2

General use evaluation

In Table 6.21, *Sum* indicates the total number of phases, steps, public steps and solutions in all contributions. *Mean* specifies the average number of each variable per student. The row *STDEV* shows the standard deviation of all variables.

Table 6.21: Numerical use data in Study 2

	Phases	Steps	Public steps	Solutions
Sum	177	40	0	31
Mean	7.4	1.7	0	1.4
STDEV	4.5	3.9	0	4.1

Table 6.21 shows that the number of steps, public steps and solutions are low compared to phases. The *Mean-Phases* column shows that, in average, one contribution included 7.4 phases with text. In the second part of the general use evaluation, the contributions were divided into three categories according to the amount of text: low, medium and high. The division was made with the same rules as in Study 1. As seen in Table 6.22, most of the contributions were categorized as low or medium.

Table 6.22: Categorization of students' contributions according to the amount of text (N=23)

Category	No. contributions
Low	12
Medium	6
High	5

Contribution evaluation

In the contribution evaluation, the contributions were divided into three contribution categories as described in Study 1: basic, intermediate, and deep. Table 6.23 shows the number of contributions in each category. The evaluation shows that most of the contributions can be classified to basic and intermediate. Two contributions were categorized as deep.

Table 6.23: Contribution evaluation in Study 2 (N=10)

Category	No. contributions
Basic	3
Intermediate	5
Deep	2

Content evaluation

In the content evaluation, the same ten portfolios as in the contribution evaluation were analyzed more deeply. Contributions were analyzed to find the

content categories to describe the content of students' text. Four content categories were identified as shown in Table 6.24.

Table 6.24: Content categories in Study 2

Name	Category	Issues	Examples
CC1	Project work description	topic, meaning of the topic, plans, problems, difficulties, goals, information search, features of the software, progress, outside difficulties, solutions	<i>“At first I had major problem on what [keyboard] button should include each function (or are buttons the only solution). But now I have managed to choose a good button and I'm very pleased with the work. Procedures inside the game are also clear.”</i>
CC2	Metacognitive awareness	mood, success, difficulties of tasks, time management, failures, goals, plans, feelings, things that have been learned, evaluation, problems, solutions, important questions, reasons for not progressing	<i>“I had no time to make the program as big as I was planned. Still, when one runs the program for couple of minutes, one sees clearly that is follows the survival of real species; most beautiful and clever creature will survive while the harmful features fade away. These features, however, stay partly in the genomes of the population.”</i>
CC3	Meaning and description of the phases and tasks	starting point, goals, questions, tasks, description of phase, success in phase, evaluation of the importance	<i>“I tested the program during the whole project after every new solution. But there were some problems that I could not fix.”</i>
CC4	Other	portfolio feedback, testing	<i>“This portfolio means only extra work for me. This is the problem. Portfolio is complaining when my goal is to do all the phases at once. Red notices are every where and there are past deadlines at every page, which have already expired ages ago”.</i>

Figure 6.4 shows the distribution of content categories among phases, steps and solutions within the ten portfolios. The categories CC1 and CC3 were dominant in all situations except in phases. Over 50% of all entries were in the category CC1.

Furthermore, approximately 20% of all entries belonged to the metacognitive awareness category.

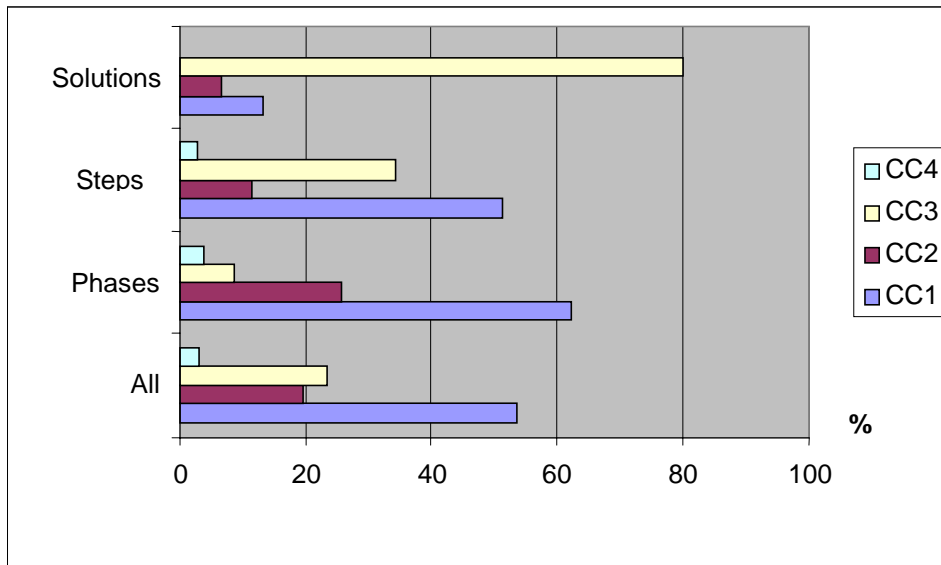


Figure 6.4: Content evaluation in Study 2

Results of the experience analysis in Study 2

Questionnaire evaluation

The questionnaire (Appendix B) in the Study 2 consisted of four parts: general use of the tool, frequency of using the tool, statements about the use of LEAP, and an empty page for open comments. Table 6.25 shows the results of the first part. The most interesting figures have been emphasized. Altogether 12 students out of 23 returned the questionnaire. Students had a scale of 1-5 to evaluate the statements: 1 = strongly disagree, 2 = disagree, 3=undecided, 4 = agree, 5 = strongly agree.

Table 6.25: Statements about the general use of the tool (N=12)

Statement	Number of answers				
	1	2	3	4	5
LEAP was easy to use	2	2	1	6	1
It was effortless to start using the tool	1	1	4	4	2
Given instructions were clear	1	1	4	3	3
I felt that LEAP was useful for me during the course	7	2	3	-	-
The structure of the tool was clear	2	3	6	1	-

The second part of the questionnaire evaluated the regularity of the use of the tool during the course. The answer scale was: 1=daily, 2=weekly, 3=once or twice a month, 4=a few times during the course, and 5 = never. Table 6.26 shows the results of the second part.

Table 6.26: Regularity of the use of the tool (N=12)

Statement	Number of answers				
	1	2	3	4	5
I logged on to the tool	1	2	2	7	-
I added steps	-	1	2	3	6
I used the Distant Analogies activity	-	-	1	4	7
I used the Question Lists activity	-	-	-	3	9
I used the BruteThink activity	-	-	1	2	9

In the third part of the questionnaire, the students evaluated a set of statements about LEAP. Table 6.27 shows the results of the third part.

Table 6.27: Statements about the use of LEAP

Statement	Number of answers		
	Agree	No opinion	Disagree
Distant Analogies, Question Lists and BruteThink activities were useful for defining new ideas	-	8	4
It was easy for me to write about the Programming Project Course with the tool	1	-	11
I cannot find any use for LEAP	5	4	3
LEAP helped me understand the phases of the project	4	2	6
LEAP helped me organize my work	1	2	9
I used LEAP to solve problems during the course	1	1	10

The third part of the questionnaire included a line for commenting the statements in Table 6.27. The fourth part of the questionnaire was an open page for free comments. Appendix D includes a summary of students' comments from the third part of the questionnaire in Study 2.

Interview evaluation

In the second part of the experience analysis, the *interviews* conducted in Study 2 were evaluated using content analysis. The interviews were analyzed through the themes of the interview. Table 6.28 shows the topics covered in the interview. The questions were altered during the interview depending on students' answers.

Table 6.28: Topics covered in the interview

Topic	Questions and themes
Earlier use of learning portfolios	Have you previously used in school, or during your studies, a portfolio or diary type of working method?
Use of LEAP during the course	How often did you use LEAP during the course? Why did you not use it?
Usefulness of LEAP	Did you feel that LEAP was useful for you? Would you use the tool for longer periods of time? Individual question followed depending on the answers.
Problem solving support activities	Did you try the problem solving support activities in LEAP during the course? What is your opinion about those activities?
Areas of improvement	What were the main weaknesses of the tool?
New features	How would you develop the tool further?

The original six topics were reduced to two themes in the analysis: use of the tool and usefulness of the tool. Areas of improvement and new features themes were dropped out because students did not have any usable comments on these topics. In the following categorization, the results of the analysis with the final themes are presented.

Three issues were identified inside the *use of the tool* theme: earlier use of portfolios, use of LEAP during the course and problem solving support activities. Table 6.29 shows the results of the evaluation. Details are available (Suhonen, 2005).

Table 6.29: Students' answers to the use the tool theme

Issue	Students' answers/opinions
Earlier use of portfolios	<ul style="list-style-type: none">- some earlier experience with a different kind of portfolio- no previous experience
Use of LEAP during the course	<ul style="list-style-type: none">- the tool was used almost every week- text was added in larger parts- tool was used a few times per month when the student was working with the project- tool was used because it was mandatory- tool was not used because the project did not progress- no time to use the tool
Problem solving support activities	<ul style="list-style-type: none">- were used a couple of times during the course- were not used because they were difficult to find- did not bring any extra value- were available, but they were not used

The second theme in the evaluation was the *usefulness of the tool*. Three issues were identified within the theme: general usefulness, support in managing the project, and LEAP in longer periods. Table 6.30 shows the results of the evaluation.

Table 6.30: Students' opinions about the usefulness of the tool theme

Issue	Students' opinions
General usefulness	<ul style="list-style-type: none">- the tool was not considered to be beneficial or useful- to see what to do at different time of the course.
Help to manage the project	<ul style="list-style-type: none">- helped a little bit, but it was not radical- phases in the tool were peculiar.
LEAP in longer periods	<ul style="list-style-type: none">- not energy to use the tool during the whole ViSCoS.- it might have helped to see whether a student had learned something during a longer period of time

Extra assignments

The third part of the experience analysis in Study 2 was the evaluation of the *extra assignment*. Extra assignments were given to those students who did not use LEAP at all during the course. The students were instructed to write a two to three pages report on LEAP. The assignment included two questions:

1. Test every function of the tool. Write a short report about the test. In your report you can discuss issues, such as main shortcomings of the tool, usability problems, functions that are not logical and annoying features, etc.

2. Imagine yourself as responsible for the future development of the tool. Write a report on things that you would change. In your report, you can include issues, such as ideas about new features or improvements of current features. You can also reflect the following question: How would you improve the tool so that it would be useful to you?

Two themes were identified in students' assignments: function of the tool and development ideas for LEAP. The function of the tool included students' opinions about the features and function of the tool. The development ideas theme included students' ideas or suggestions to improve LEAP. Table 6.31 summarizes the issues found in the extra assignment that belonged to the *function of the tool* theme. Details are available (Suhonen, 2005).

Table 6.31: Students' opinions about the function of the tool theme

Issue	Students' opinions
Text fields in phases	- the mood text field does not relate to anything
Usability and interface problems	- problem solving activities were difficult to reach - the tool looked boring
Problem solving support activities	- activities were not useful because student felt that he could solve all the problems inside his head.
Public steps	- not interested in someone else's problems

Students' *development ideas* were divided into three categories as in Study 1: cosmetic improvements, general improvements and new features. Table 6.32 presents the amount of each development type found in students' extra assignments. Appendix D includes a summary of students' answers related to the development idea theme.

Table 6.32: Number development ideas types from students' comments

Type	Number of ideas
Cosmetic improvements	9
General improvements	6
New features	1

6.3 Thread 2: Re-design of the technical architecture

6.3.1 Needs analysis: Analysis of an instructor questionnaire

The needs analysis in the second thread included the analysis of a questionnaire sent to the two instructors who had worked earlier in the Programming Project Course. The goal of the questionnaire was to get more insight and fresh ideas on how to incorporate LEAP into the course. In the first part of the questionnaire, the instructors were asked to evaluate the most difficult aspects of the course for students. We also wanted to know about any frequently appearing problems. The following challenges and problems could be identified from the instructors' answers

- Understanding the documentation and using the document;
- Lack of planning;
- Use of API documentation; and
- Selection of too ambitious topics.

In the second part of the questionnaire, the instructors were asked to express their opinions on what kind of technical support would solve the problems related to the course. The following ideas emerged

- A tutorial application for documentation and planning;
- Tracking of the project plan, progress of the project and timetables throughout the project;
- Support planning with concepts maps or other graphical tools;
- Search tool for API documentation and error messages; and
- Intelligent categorization and suggestion of topics.

The results of the questionnaire did not affect the implementation in Thread 2. As with the interviews and extra assignments in Study 2, comments from instructors can serve as a source of inspiration for future threads in the LEAP development. Moreover, ideas from the questionnaires could be used as a starting point for the development of totally new technical support for the course.

6.3.2 Implementation: New technical architecture

Within Thread 2, a new version of LEAP was implemented. The new implementation was based on Xalan-Java XSLT engine and Java Servlet API (Kainulainen, 2004). Instead of WebMacro template pages, the content for LEAP

was produced with XSL-pages. The interface and content of the tool could be customized by editing the XSL style sheet files. Table 6.33 shows the content of the XSL style sheet files within the new architecture.

Table 6.33: XSL style sheet files

File	Content
Pma	Digital learning portfolio activities and the core structure of the tool.
Distant	Content and function of the Distant Analogies activity.
Questionlist	Content and function of the Question Lists activity.
Brute	Content and function of the BruteThink activity.

Appendix C includes the *Distant* XSL style sheet file. The XML based implementation was chosen because the needs analysis component in Thread 3 depicted that it was preferable for the mobile adaptation extension. Furthermore, small changes were implemented to the function of the tool because the studies within Thread 1 revealed concrete development ideas. Especially Study 2 showed that there was room for improvements. Table 6.34 summarizes the main changes made to the function of LEAP within the second thread. The changes were mainly made to the content and activities of single LEAP pages. We did not make any changes to the core structure of the tool.

Table 6.34: Main changes made within the second thread

Target	Change
History page	- learners can access the phase information directly from the history page. Learners can add, access, and remove steps and solutions through the history page. - modifications to the interface to separate the history page from the process page
Solution activity	- removal and modification of added solutions.
Number of steps in a phase	- an information field was added to the process page to show how many steps learners have added to a phase.
Number of solutions in a step.	- an information object was added to the phase page to show how many solutions user had added to a step.

6.3.3 Formative evaluation: Study 3

Within the formative evaluation component in Thread 3, Study 3 was conducted to evaluate the new version of the tool. The study context was similar to the context in Study 2. The procedures, goals and arrangements of the course were almost identical to those in Study 2.

Study 3: LEAP in the ViSCoS Programming Project Course

In Study 3, some modifications were made to the basic configuration of the tool. The most important change was that the public step activity was removed from the tool. Both studies in Thread 1 indicated that the implementation of the public step activity was not working as planned. For instance, in Study 2 students did not add any public steps. Since the tool is aimed at supporting individual learners, the public step was not considered to be relevant for the evaluation. Small changes were also made to the content of some pages. Problem solving support activities were not used extensively in Study 2. Furthermore, some of the students indicated that LEAP was not enough grounded to the course activities. Hence, problem solving support activities were modified to fit better into the course. For instance, new question lists aimed to the particular phases were added. Table 6.35 includes the phases and Table 6.36 includes the instruction object, user and step text fields in Study 3.

Table 6.35: Phases in Study 3

Phase	Meaning
Topic selection	Selection of the topic for the project
Focus of the problem space	The original topic is focused
Planning and project plan	General planning of the class and method structures.
Start of programming	Start of the implementation
Implementation	Implementation of the project
Testing	Testing of the program
Documentation	Documentation of the project
Finishing up and submission	Last moment implementation and submission of the project
Feedback and evaluation of the project	Process the feedback from the instructors and evaluate the project
Other issues	This phase can include issues that do not fit into other phases

Table 6.36: Text field entries in Study 3

Entry	Text fields	Aim
Instruction object	Description Goals of the Phase Important to Notice Instructions	Distribution of the information related to the phases; goals, important issues and specific instructions.
User text	Memory List Free text	Students could write down general issues related to phases; memory list for issues that needs to be remembered and free text.
Step text	Name Description Meaning	Describe the meaning of steps.

Evaluation scheme in Study 3

The evaluation scheme in Study 3 was similar to the one in earlier studies. The use of environment analysis was identical to Studies 1 and 2. Altogether 22 contributions were in the tool after the course ended. The content evaluation in Study 3 did not include all the contributions because four contributions were extremely low. Furthermore, only step and solution entries were analyzed in the content evaluation because of low amount of text in the phases. During the evaluation, it was also noticed that steps and solutions could not fit into the same categories. Hence, different categories were created for steps and solutions in the content evaluation. Table 6.37 summarizes the use of environment analysis in Study 3.

Table 6.37: Use of environment analysis in Study 3

	Aim	Data	Method	Results
General use evaluation	Evaluate the use of tool at a general level	22 contributions	Numerical use data: amount of phases, steps and solutions. Categorization of contributions according to the amount of text: low, medium and high.	Table 6.40 Table 6.41
Contribution evaluation	General picture of students' contributions in LEAP	22 contributions	Categorization of contributions to contribution categories; basic intermediate and deep.	Table 6.42
Content evaluation	Content of students' contributions.	16 contributions	Content analysis	Figures 6.5, 6.6

Table 6.38 summarizes the experience analysis in Study 3. In the questionnaire evaluation, the questionnaire relating to the use of LEAP during the course was evaluated. In the interview evaluation, interviews conducted with three students were evaluated. The students who had not yet answered the questionnaire were asked to participate in the interview.

Table 6.38: Experience analysis in Study 3

	Aim	Data	Method	Results
Questionnaire evaluation	Students' opinions about the functionality and usefulness of LEAP	8 questionnaires	Numerical and qualitative analysis of questionnaires	Tables 6.45, 6.46, 6.47
Interview evaluation	Analyze how students perceived the tool. Find concrete improvement ideas.	Three interviews	Qualitative analysis of interviews	Table 6.49

Results of the use of the environment analysis in Study 3

General use evaluation

Table 6.39 shows the results of the numerical use evaluation. *Sum* indicates the total number of phase text fields including text, steps, and solutions in all contributions. *Mean* specifies the average number of each variable per student. *STDEV*, the last row, shows the standard deviation of each variable.

Table 6.39: Numerical use data in Study 3

N=22	Phases	Steps	Solutions
Sum	21	214	177
Mean	1.0	9.7	8.1
STDEV	2.2	7.1	8.3

The number of steps and solutions is higher compared to the phases. Standard deviation is also high indicating that there is a lot of variation. The low number of phases is partly explained by a bug when modifying the user text fields. The database of LEAP included many “NULL” phase entries. This indicated that students had added text, but during the later use of the tool the text was removed from the database. Similar bug was also mentioned by a student in Study 2. In the second part of the general use evaluation, the amount of text in students’ contributions was evaluated as explained in earlier studies. Table 6.40 shows the results of the evaluation.

Table 6.40: Categorization of students’ contributions according to the amount of text
(N=22)

Category	No. contributions
Low	10
Medium	8
High	4

Contribution evaluation

In the contribution evaluation, the contributions in LEAP are divided into three categories, namely, basic, intermediate, and deep as explained in earlier evaluations. Table 6.41 shows the number of contributions in each category. The evaluation shows that most of the contributions were classified into basic and intermediate.

Table 6.41: Contribution evaluation in Study 3 (N=22)

Category	No. contributions
Basic	6
Intermediate	13
Deep	3

Content evaluation

In the content evaluation, the content of 16 contributions was analyzed in more detail. Different categories were created for steps and solutions. Table 6.42 shows the step categories and examples of content.

Table 6.42: Step categories in Study 3

Name	Category	Issues
SC1	Specific text related to the topic of the program	Bugs in the program, things to do, solutions, new solutions to problems
SC2	General project work description	Feedback, work hours, pair or individual work, improvement, programming order, procedures and tasks in phases, problems, plans, changes, designs, progress of the project
SC3	Metacognitive awareness	Pondering, time problems, ideas about the topic, evaluation of the work, importance of tasks in the project, questions, difficulties, scope of the project, ideas, processing of the feedback, evaluation of the difficulties, success, decisions, prioritisation, goals, satisfaction/dissatisfaction
SC4	Other	Testing, portfolio tool issues

Figure 6.5 shows how the step categories were distributed among the step contributions in Study 3.

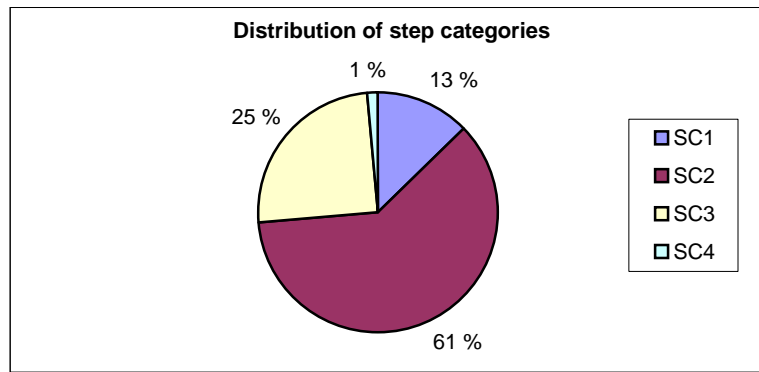


Figure 6.5: Content evaluation of steps in Study 3

The category SC2 is dominant. Most of the steps described the progress of the project on a general level. One fourth of steps included metacognitive processing. A similar categorization was done for solutions. Table 6.43 shows the solution categories in Study 3.

Table 6.43: Solution categories in Study 3

Name	Category	Explanation
SOC1	Direct answer	Solution included a direct answer to question/problem posed in a step.
SOC2	New perspective	Comments or new perspectives to a step.
SOC3	Problem solving support activity	Student has used problem solving support activities to process the step

Figure 6.6 shows the distribution of solution categories among solution entries in Study 3.

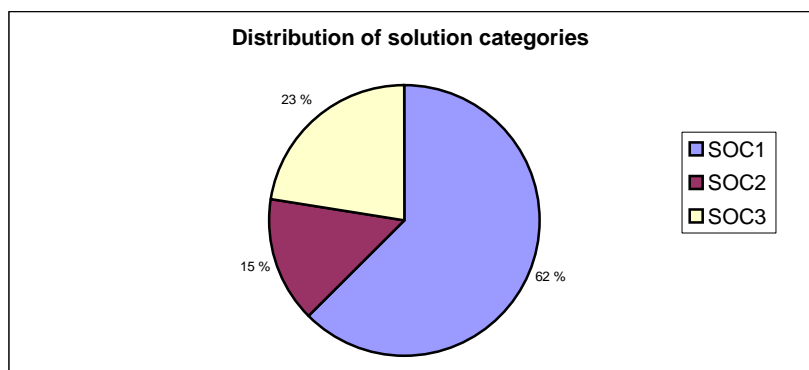


Figure 6.6: Content evaluation of solutions in Study 3

The category SC1 was the dominant one in Study 3. Most of the solutions included text that was a direct solution to a step. Figure 6.5 also shows that 23% of solutions were processed with problem solving support activities.

Results of the experience analysis in Study 3

Questionnaire evaluation

The questionnaire in Study 3 was almost similar to the one in Study 2. The first part included questions about the general use of the tool (Table 6.44). Altogether 8 students out of 22 returned the questionnaire. The most interesting figures have been emphasized in bold font. The answer scale was: 1=strongly disagree, 2=disagree, 3=undecided, 4=agree, 5=strongly agree.

Table 6.44: Statements about the general use of the tool (N=8)

Statement	Number of answers				
	1	2	3	4	5
LEAP is easy to use	-	-	2	4	2
It was not difficult to start using the tool	-	1	5	2	-
Given instructions were clear	-	1	3	3	1
I felt that LEAP was useful for me during the course	4	2	2	-	-
The structure of the tool was clear	1	2	2	3	-
LEAP helped me to manage the project	4	3	1	-	-
The purpose of LEAP become clear during the course	1	2	4	1	-

The second part estimated how regularly students had used the tool. The answer scale was: 1=daily, 2=weekly, 3=a few times per month, 4=a few times during the course, 5=not once. Table 6.45 shows the results of the second part.

Table 6.45: Regularity of the use of the tool (N=8)

Statement	Number of answers				
	1	2	3	4	5
I logged into the tool	-	-	4	4	-
I added steps	-	-	4	3	1
I added solutions	-	1	2	4	1
I used the Distant Analogies activity	-	-	1	1	6
I used the Question Lists activity	-	1	-	1	6
I used the BruteThink activity	-	-	1	-	7

In the third part of the questionnaire, the students evaluated a set of statements about LEAP. Table 6.46 shows the results of the third part.

Table 6.46: Statements about the use of LEAP

Statement	Number of answers		
	Agree	No opinion	Disagree
Distant Analogies, Question Lists and BruteThink activities were useful for defining new ideas	-	3	5
It was easy for me to write about the Programming Project Course into the tool	-	2	6
I cannot find any use for LEAP	5	3	-
LEAP helped me understand the phases of the project	1	3	4
LEAP helped me organize my work	-	2	6
I used LEAP to solve problems during the course	-	2	6

In the third part of the questionnaire, the students had an open space for comments. Furthermore, the fourth part of the questionnaire included an empty page for open comments related to the tool. Appendix D includes a summary of all the comments found in the third and fourth part of the questionnaire.

Interview evaluation

The interview topics in Study 3 differed slightly from the ones in Study 2. Table 6.47 shows the topics and questions that were covered during the interview. The interview was an open interview where the questions were altered depending on the students' answers.

Table 6.47: Interview topics and questions in Study 3

Topic	Question
Use of LEAP during the course	How often did you use LEAP during the course? Why did you not use it? Did you try the problem solving support activities in LEAP during the course? Which part of the course did you use the tool? Did you read the earlier steps later in the course?
Usefulness of LEAP	Did you felt that LEAP was useful for you? What is your opinion of the problem solving support activities?
Programming Project Course	Were there any major problems during the course? Did you get enough support during the course?
Areas of improvement	What were the main weaknesses of the tool? Where there any problems?
New features	How would you develop the tool further?

The original six topics were reduced into two themes after a preliminary overview analysis: the function and purpose of the tool and problems in the Programming Project Course. Areas of improvement and new features themes were dropped out because the students did not have any usable comments on those themes. Table 6.48 includes the students' opinions related to the function and purpose of the tool theme.

Table 6.48: Students' opinions about the function and purpose of the tool

Issue	Opinions
Frequency of the use	<ul style="list-style-type: none"> - as much as the student had to - all the entries were added at the end of the course - a couple of times during the course
Addition and reading of steps	<ul style="list-style-type: none"> - during the whole course - check whether all the important things are in the tool - mostly in the end of the course
Problem solving support activities	<ul style="list-style-type: none"> - activities were not used because the student did everything by himself
Purpose and usefulness of the tool	<ul style="list-style-type: none"> - did not understand the meaning of the tool - the tool was extra task in the project, but nothing was gained from it - there was no advantage - purpose of the problem solving activities were unclear

The second theme in the students' responses was related to the problems in the Programming Project Course. The following types of answers were identified;

- No main problems during the course;
- The biggest problem was the lack of time; and
- The problems in the project were familiar.

6.4 Thread 3: Mobile adaptation extension

The third thread, the mobile adaptation extension, has been active from the start of LEAP development. The platform adaptation layer adapts the technical, pedagogical and application layers to various platforms and end-devices (Jones et al., 1999). The mobile adaptation extension is the starting point for implementing the platform adaptation layer to LEAP.

6.4.1 Needs analysis: Requirements for the mobile adaptation extension

The mobile adaptation extension would enrich the function of LEAP. With mobile devices students can compose and read carry-on notes, which brings LEAP closer to the actual learning situation (Kinshuk et al., 2003). The tool would be available whenever needed. For instance, if students discover fresh ideas related to their learning process, they can instantly write them down to LEAP (Kinshuk et al., 2004).

The WebMacro implementation in Thread 1 was not suitable for the mobile adaptation extension. The promising option was to use XML/XSLT based architecture. WebMacro templates had their own advantages, such as more simplified syntax. However, we noticed that WebMacro did not meet the needs of the mobile adaptation extension. XML/XSLT implementation allowed the adaptation of the tool for many platforms. XML-based implementation is also highly extensible; in theory there is no limit to implementing new features to the tool. XML/XSLT based solutions also have ready-made tools for database connections and operations (Kainulainen, 2004). Hence, a viable technical solution for the mobile adaptation extension was the XML/XSLT architecture.

Another important aspect is the needs for the actual adaptation process. The challenges in mobile devices are the small screen size and difficulties in data input (Luchini et al., 2003). Existing research suggests that reading or browsing a piece of text with a smaller screen is not a problem (Jones et al., 1999). Simple navigation

procedures and menu selections can also be performed easily, if the users are familiar with the functions. However, because LEAP includes more complex tasks than just reading or browsing a piece of text, there is a need for creating the mobile adaptation extension. The main challenge is that LEAP includes both static content and interactive activities. As presented in Chapter 5, LEAP can be divided into a set of pages. Furthermore, the function of each page can be divided into content and activity. The content describes the physical structure of each page. The content in LEAP consists of normal HTML elements, like tables, images and forms. Activity depicts the task or purpose of a page. The activity part implements, for instance, the interaction with the learner. The mobile adaptation extension should consider both of these features.

The pages of LEAP can be described by two dimensions according to their function and stability. Both of these dimensions have two values. The function dimension separates the content and activity. The content includes the physical structure of a page. The activity is used to implement users' active involvement. The two values for the stability dimension are static and dynamic. Static pages do not include any complicated structures and they do not change while using the tool. Furthermore, the pages include only one or two input or output objects. Only a few pages in LEAP are dynamic, but these are the main challenge for the mobile adaptation extension. Dynamic pages include several objects, both input and output, and the content of the pages changes during the use of the tool. The combination of dimensions results in four categories representing the needs for the mobile adaptation extension. Table 6.49 gives an example of a LEAP page that fits under a certain category. However, the categories overlap with each other and they are not mutually exclusive; one page in LEAP can have features from all categories.

Table 6.49: Features of LEAP divided into four categories

<p>Content-Static Content of the tool do not change the use of LEAP. No interactivity with the learner. <i>Example:</i> Text field templates in phases</p>	<p>Activity-Static The learner is required to interact with static objects. <i>Example:</i> Text is added to the static text fields.</p>
<p>Content-Dynamic Content of the tool changes during the use, but there is no interactivity with the learner. <i>Example:</i> History page</p>	<p>Activity-Dynamic Learner interacts extensively with the tool and the content of the pages depends on the use context. <i>Example:</i> Problem solving support activities.</p>

Three content adaptation techniques can be identified for implementing the mobile adaptation extension: re-authoring, transcoding and structure-aware models (Chen et al., 2001b; Hwang et al., 2003; Kinshuk & Goh, 2003; Monash et al., 1999; Phan et al., 2002; Trevor et al., 2001). Table 6.50 summarizes the analysis of how the three content adaptation techniques would fulfill the needs depicted in Table 6.49 (Kinshuk et al., 2004). In Table 6.50, “++” indicates that a technique can meet all the necessary needs. Furthermore, “++” indicates that a technique is preferable from the viewpoint of the category. Moreover, “+” indicates that a technique can meet some of the needs inside the category. The technique is also suitable, but not preferable for implementation. Finally, “-“ indicates that the method neither can satisfy the needs of the dimensions nor it is suitable for implementing certain type of pages.

Table 6.50: Content adaptation techniques to suit the needs of the mobile adaptation extension

	Re-authoring	Transcoding	Structure-aware models
Content-static	++	+	-
Content-dynamic	-	++	+
Activity-static	-	+	+
Activity-dynamic	-	-	+

As seen in Table 6.50, none of the techniques can completely satisfy the needs of the mobile adaptation extension. The main weakness of the content adaptation techniques was that they could not account for specific needs of the activities in LEAP. The techniques are too general in describing the function of certain tasks in the tool. For instance, problem solving support activities can be modified to meet the needs of a certain use context. In that context, an activity might have a certain task, like producing novel ideas. The adaptation process should weight the importance of tasks. Furthermore, the content techniques are not suitable for a dynamic adaptation. Therefore, a viable solution for the mobile adaptation extension would be to combine different techniques such as Multiple Representation Mobile Adaptation for dynamic adaptation, task adaptation for adapting the activities and content adaptation for adapting the content (Eisenstein et al., 2000; Kinshuk & Goh, 2003; Kinshuk et al., 2003).

6.5 Summary and discussion

6.5.1 Results of the LEAP development

Three threads were identified in the development of LEAP: the first prototype, re-design of the technical architecture and the mobile adaptation extension. The first thread was a starting point for the development. The design ideas of LEAP were based both on identified needs in the ViSCoS Programming Project Course and theoretical perspectives. Two pedagogical services were implemented: digital learning portfolio and creative problem solving support. Furthermore, third pedagogical service, the provocative agent, was identified. An important aspect in the first thread was to test the tool in authentic learning settings as soon as possible. Studies 1 and 2 evaluated the use of the tool in different learning contexts.

The main focus in the second thread was to implement the new technical architecture for the tool. LEAP was also modified based on the formative evaluations in the first thread. The main indication from Thread 1 was that the tool should be simplified. Especially Study 2 revealed that there was still problems with the function of the tool. A similar situation also happened in Jeliot development, where the first Jeliot version was too complicated for novice users (Moreno et al., 2004b). Within the second thread, Study 3 was conducted to evaluate LEAP in a similar context of use as in Study 2. The main result of Study 3 was that LEAP should be more grounded to the course context. The students felt that the tool was somehow outside of the course. The mobile adaptation extension thread included only the needs analysis component. The needs analysis, however, affected the implementation component in Thread 2. Figure 6.7 visualizes the development of LEAP.

The development of LEAP has been done with low resources. I have been involved mainly in concept design and arranging and evaluation the conducted research studies. Vesa Kainulainen, a MSc. student was responsible for the implementation components in Threads 1 and 2. Furthermore, the development process has been rather fast (Figure 6.2). The development started in the beginning of year the 2002, and the Studies 1 and 2 were conducted between January 2003 and

August 2003. The implementation within the second thread was done in May-September 2003. Study 3 was conducted in May 2004-August 2004.

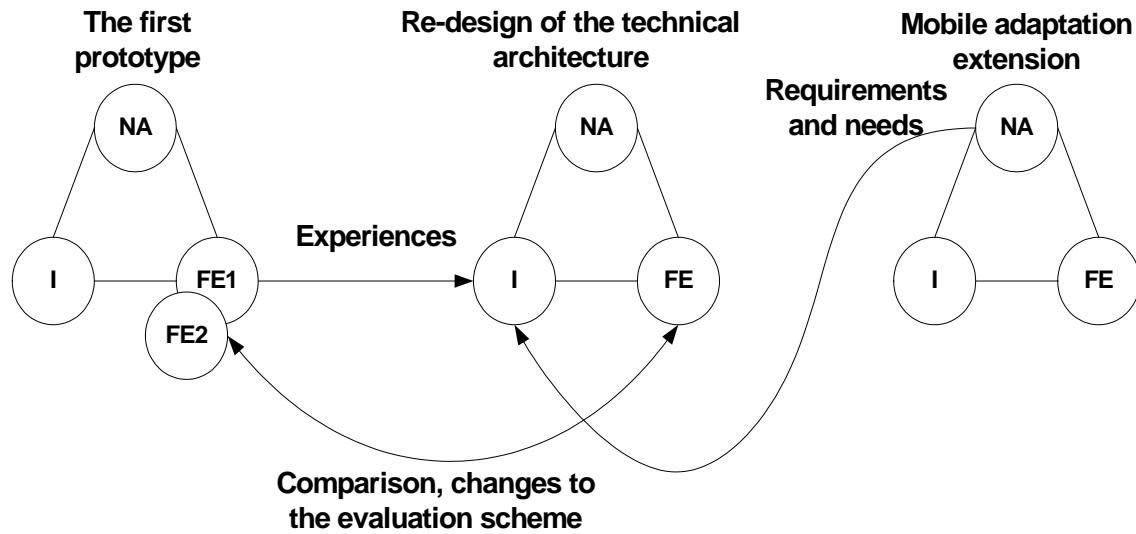


Figure 6.7: LEAP development

Studies 1, 2 and 3 can be compared because a similar evaluation scheme was used. The first part of the scheme was the use of environment analysis. The general use evaluation in the use of environment analysis was similar in all studies. In the numerical use evaluation, the number of phases, steps, public steps and solutions in students' contributions was evaluated. Table 6.51 summarizes the results of the three studies (results in Tables 6.7, 6.21, 6.39). The first column *Phase (%)* indicates how many percentages of phase text fields, in average, included text in one contribution. The other columns show the mean value of each variable per student in all studies. As seen in Table 6.51, the students used LEAP more actively in Study 1 than other studies. However, the low number of phases in Study 3 is explained by the bug in the program.

Table 6.51: Amount of phases, steps, public steps and solutions

Study	Phase (%)	Steps	Public steps	Solutions
1	80	15.9	2.4	1.4
2	53	1.7	0	1.4
3	10	9.7	-	8.1

In the second part of the general use evaluation, the contributions in LEAP were divided into three categories according to the amount of text: low, medium and high.

Figure 6.8 shows the distribution of the number of text categories in all studies (results in Tables 6.11, 6.22 and 6.40). One can clearly see the differences between Study 1 and the other studies. The results are completely opposite. Moreover, Studies 2 and 3 are almost identical.

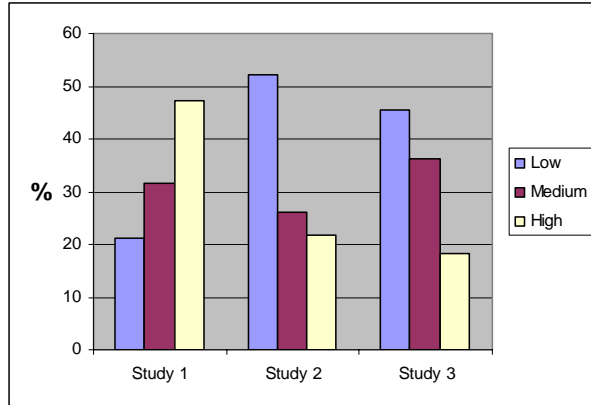


Figure 6.8: Amount of text in students' contributions

The second part of the use of environment analysis was the contribution evaluation. Students' contributions were divided into three categories: basic, intermediate, and deep. Figure 6.9 shows the results of the contribution evaluation in all three studies (results in Tables 6.12, 6.23 and 6.41). The results are similar to the numerical use and amount of text evaluations. It should be noticed, however, that in Study 2 all contributions were not evaluated. Altogether, the use of the environment analysis revealed that students used LEAP more extensively in Study 1 than in other studies.

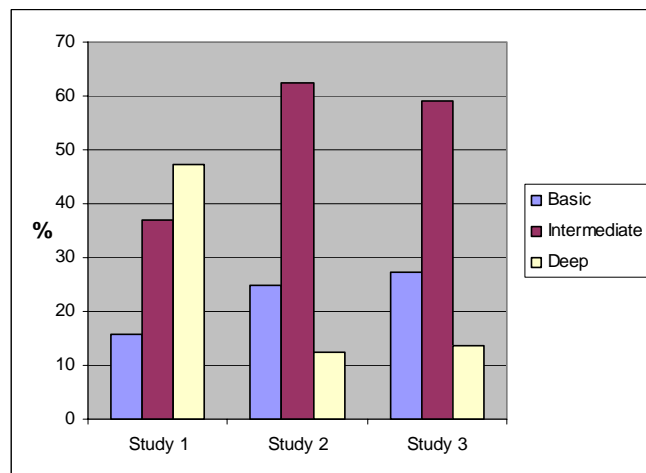


Figure 6.9: Contribution evaluation

The third part of the use of environment analysis was the content evaluation. The contributions were divided into the content categories. Table 6.52 shows the two dominant content categories in each study (results in Figures 6.3, 6.4 and 6.5). It should be noticed that Study 3 includes the distribution of steps.

Table 6.52: Dominant content categories in studies

Category	Percentage
Study 1: Tasks in the course	47.7
Study 1: Metacognitive awareness	36.5
Study 2: Project related text	53.7
Study 2: Meaning of the phases	23.5
Study 3: Project work	61.0
Study 3: Metacognitive awareness	21.0

Table 6.52 confirms the results of the earlier evaluations. In Study 1, the LEAP contributions included text that was expected from students' contributions, e.g. text that can be characterized as metacognitive awareness. Study 1 revealed that students had used the tool at least satisfactorily. In Study 2, the two dominant categories did not include text that was expected from the contributions. Finally, the results of the Study 3 were better compared to Study 2, but still the amount of metacognitive type of contributions was quite low.

One factor that could explain the different results is the high impact of the digital learning portfolio on the grade in Study 1. In Studies 2 and 3, LEAP was mandatory, but it did not affect the grade. The second factor is the maturity of the students. In Study 1, students were normal university students (age 20-30), but in Studies 2 and 3, the students were high school students (age 16-18). Furthermore, most of the students in Study 1 had an earlier experience with a portfolio (a portfolio method had been used in earlier courses), while the interview evaluation in Study 2 revealed that students did not have much experience on learning portfolios (Table 6.29). Naturally, the contact and distance learning context affects the situation. In Study 1, most of the students apparently used the tool through the premises of the university. In Studies 2 and 3, the students were using the tool mainly from their homes. It seems that the tool did not provide enough support, so the students would be ready to use it through a modem connection. Although this was considered during the development, it seems that there is room for improvements. ViSCoS students in Studies 2 and 3 also mentioned about the usability problems of the tool. It seems that

if learners have some obstacles in using the tool, more concrete support or direct advance the tool should provide. This can be clearly seen from the comments in Studies 2 and 3 that the tool should be more focused to the course.

The experience analysis in all studies revealed both positive and negative comments towards the tool. An important aspect from the design method perspective is the number of concrete development ideas to improve the tool. In Figure 6.10, three evaluations in the experience analysis are compared in relation to the number of development idea types. In Figure 6.10, *IE1* refers to the results of the interview evaluation in Study 1 (Table 6.15), *CE1* refers to the comment evaluation in Study 1 (Table 6.16) and *EA2* refers to the extra assignment evaluation in Study 2 (Table 6.32). Figure 6.10 shows that the interview and comment evaluations from Study 1 provided slightly more usable development ideas compared to the extra assignment evaluation in Study 2. However, the extra assignment evaluation was the only evaluation in Studies 2 and 3 that produced concrete development ideas. Some of the ideas (mainly cosmetic and a few general) from Studies 1 and 2 were implemented within the Thread 2. The differences between the interviews in Study 1 compared to other studies are partly explained by the age difference between the students. The interview atmosphere in Study 1 was clearly more open and relaxed. Students were eager to share their ideas. It was also a good idea to send the interview questions beforehand to the interviewees. In Studies 2 and 3, the interview situation was not so natural. I almost had to squeeze the information out of the students. Clearly some of the students were nervous; maybe the situation was new and frightening for them.

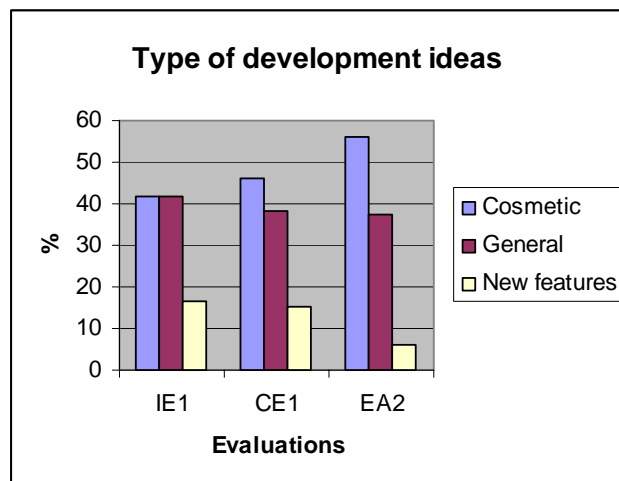


Figure 6.10: Type of development ideas

In Study 1, some students thought that the role of LEAP was to act as an instant feedback channel to the instructors. There was also some confusion about the meaning of the text fields (Table 6.14). The interview evaluation in Study 1 also revealed contradictory opinions. One student thought that the restrictive nature of the tool prohibited his writing. However, another student saw that the static structure helped in concentrating to the important issues. The questionnaires and interviews in Studies 2 and 3 did not provide so many concrete ideas. Furthermore, the general message from the questionnaires and interviews was that LEAP was easy to use. However, some students also thought that LEAP was not useful for them (Tables 6.25, 6.27, 6.30, 6.44, 6.46). The questionnaires also revealed that the students who answered the questionnaire had not been using the tool regularly (Tables 6.26, 6.45). A recurring comment in all studies was that the idea of LEAP was good, but the implementation did not work.

The most important result was that the problem solving support activities were hardly used during Studies 2 and 3. Furthermore, students' opinions towards these activities were mostly negative. One aspect that emerged from the interview evaluation in Studies 2 and 3 was that the students had added larger parts of text in one occasion (Table 6.48). Hence, the tool was not used as it was supposed to be genuine reflection or metacognitive processing. Finally, a couple of students were looking for functions that were more incorporated to the course context. Altogether, the results from the use experience analysis indicated that there is need for improvements in LEAP, especially in problem solving support activities. However, the interesting aspect is how to improve the tool in a situation where some students indicate that they do not need any extra help during the course (interview evaluation in Study 3).

6.5.2 Evaluation of the results

The critical factor in the evaluation scheme used in LEAP development is how accurately the used methods map the reality and pragmatic results (Fincher & Petre, 2004). The clear advantage of the evaluation scheme in LEAP development was that mixed data is used (Cohen et al., 2002). The aim of the use of environment analysis was to evaluate how students have been using the tool. The general use evaluation was the most reliable evaluation in the use of environment analysis. The number of phases, steps, public steps and solutions, and categorization of the amount of text are quite exact measures on evaluating the use of LEAP. The general use evaluations in

different studies were easily to be compared with each other. The results in the contribution and content evaluation are dependent on my interpretation on students' contributions to LEAP. However, the overall picture is important. For instance, it does not make almost any difference if the amount of metacognitive awareness category in content evaluation is 20 or 30%. The main measurement is how the category relates to other categories. Altogether, the use of environment analysis provided an exact enough picture about the use of the tool.

The aim of the experience analysis was to interpret how students perceived the function of the tool. Furthermore, the goal was to produce development ideas for LEAP. The clear results from the experience analysis in Studies 2 and 3 were that LEAP did not fulfill all the expectations. Furthermore, the interviews in Study 1 and extra assignments in Study 2 produced concrete ideas for LEAP. The interview in Study 1 was especially fruitful as explained earlier in the chapter. Hence, the experience analysis also produced pragmatic results.

7 FODEM in online study program development – the case of ViSCoS

7.1 Background

ViSCoS (Virtual Studies of Computer Science) is an online study program for first-year university-level courses in computer science (Haataja et al., 2001a). The ViSCoS curriculum consists of three main areas – the preliminaries of ICT, basics of programming with Java, and introduction to computer science. Altogether, ViSCoS courses require about 600 hours of work (Torvinen, 2004). The ViSCoS program is divided into three half-year long semesters: the goal is to finish the studies in a one and a half years time. If students pass the program with marks high enough (grade 2 out of 3), they are free to enter the university as computer science majors in the Department of Computer Science, University of Joensuu.

Originally, ViSCoS was offered for high school students in the surrounding area of Joensuu. The program was totally free and it was open for anyone who was interested. The first students started to study ViSCoS in September 2000. Subsequently, the program has expanded to other parts of Finland with a close cooperation with the university and partner institutions. Four groups of students have started the program between the years 2000-2004. Altogether 94 students have completed the program during this time. Almost all of the graduated students have earned a free entry to the university.

7.2 ViSCoS development process

The development of ViSCoS has been a continuous process. There have been constant changes, from the whole curriculum down to small parts of particular courses. It is actually difficult to clearly identify the different threads. Changes have been made rapidly and with various focuses. In the beginning, the focus was on the most important aspects of the program. Gradually, the structure of ViSCoS and contents of course materials have been improved. Three threads can be identified in

the ViSCoS development: the first designs, loop of improvements, and new technical solutions to support learners.

7.2.1 Thread 1: The first designs

CANDLE scheme as a basis for needs analysis

Within the first thread, the first versions of the ViSCoS courses were designed, implemented and evaluated. The design started in 1999 through preliminary discussions with interested high schools. The implementation of the first courses started in May 2000. The first group of students began their studies in September 2000. There were several interrelated factors that restricted our design (Haataja et al., 2001a). First, the high school students were used to an instructional setting different from that of the university. It was acknowledged that the ViSCoS curriculum would be a real challenge for most of the students. Secondly, the opportunities of the web were limited for compensating a missing teacher contact. Thirdly, the implementation of complete digital learning materials covering the whole curriculum was too much to compose within the time given. Finally, synchronous communication or even delivery of asynchronous video lectures was out of the question because of the low bandwidths available at students' homes. The CANDLE scheme was created to support the implementation of ViSCoS courses (Haataja et al., 2001b; Sutinen & Torvinen, 2003).

Because time was limited, the first design goal was to get the studies running at some level. The effort was directed to those aspects of the program that had to work properly. It was acknowledged that some aspects of the courses were not completely ready, but this did not prohibit the running of the first courses. The second goal was to design the ViSCoS program from the viewpoint of high school students. Timetables, course arrangements and learning materials were designed from this perspective. Almost all teaching in ViSCoS is done via the net. Students can study locally, at home or through the partner institution. One of the major design principles was the equality between students; no matter where students were situated, they could take part in the program.

Human contacts were also considered to be important part of ViSCoS. For instance, we decided that every assignment submitted by the students was assessed together with constructive feedback. Each high school (and later partner institutions)

had a tutor who helped students in troubled situations (Sutinen & Torvinen, 2003). Tutors were also contact people between the university and students. The third emphasis in CANDLE is the light interface. Therefore, the aim in ViSCoS was to keep the user interfaces of the digital learning materials simple. Furthermore, ViSCoS courses should be accessible with low bandwidth internet connections.

Finally, the emphasis in ViSCoS has been on creating meaningful digital learning environments. Digital learning materials and tools should support and motivate learners in their studies. Exercises and examples, for instance, were designed to motivate the students. A visuals-first approach was applied in creating the exercises. Visually appealing examples and exercises were used, especially in the programming courses (Kareinen et al., 2001; Kareinen et al., 2002).

Implementation of the first courses

Within the first thread, the curriculum of ViSCoS included nine courses (Haataja et al., 2001a). Table 7.1 shows the content, scope (in European Credit Transfer System (ECTS), credit points (cps)) and schedule of the courses within the first thread (Torvinen, 2004).

Table 7.1: Content, scope and schedule of ViSCoS courses in Thread 1

Course	Content	ECTS	Schedule
Computers, Operating Systems and Networks	Introduction to ICT and computing	1.5	Semester 1
Practical Use of Computers, Essential Applications and Network Services.	Practical skills of using word processing and spreadsheet applications, basics of Unix	1.5	Semester 1
Programming I	Algorithmical thinking. Basic structures of programming with Java	3	Semester 1
Programming II	Introduction to object-oriented programming (objects, graphical programming)	4.5	Semester 2
Hardware, Computer Architecture and Operating Systems	An overview of architecture of computers, parsers, system software, and databases	3	Semester 2
Programming Project	Software design, implementation, testing and documenting	3	Semesters 2 and 3
Introduction to the Ethics of Computing	General knowledge in the ethics of computing	1.5	Semester 3
Design of Algorithms	Introduction to basic issues in computer science; algorithms, computation, and data structures.	1.5	Semester 3
Research Fields of Computer Science	Introduction to a selection of research fields in computer science	3	Semester 3

Four types of learning methods were used in the courses: assignment-based, collaborative study, essay, and project. *Assignment-based* courses had a similar structure; each course was divided into 6 to 12 learning units. Each learning unit covered a one to two week period of time during the course. Each unit consisted of

- digital and printed learning materials;
- optional warm-up exercises; and
- 4-6 assignments.

In the *assignment-based* courses, the emphasis was on assignments and a learning-by-doing approach. During the courses, the students were engaged in diverse, mainly individual, learning assignments. One course was normally built upon 40 to 50 assignments distributed among the learning units. Digital learning materials helped students grasp the theory, which was tightly linked with the

assignments. The main idea behind the assignments was to get the students to solve relevant, motivating, and even open problems. The goal was to spark students' own interest to support meaningful learning.

Visually appealing game-like examples and assignments were created especially for the Programming I and II courses. The aim was to support learning of theoretical aspects through animations and event-driven keyboard and mouse programming approaches (Kareinen et al., 2001). Java-applets, for instance, were used as visual examples. The visuals-first approach was extensively utilized in the Programming II course. Because the content of the course was object-oriented programming, the assignments in the course included programming tasks of graphically oriented applets.

During the *collaborative study* method, the students made a short study in groups of two to six, utilizing the collaborative tools provided by the digital learning environments. A study is a small investigation on a given subject. There were no ready-made learning materials; the students used the internet and the library on their own to compose the studies. Each study was assessed by the instructors and peers. In the *essay* method, students wrote an individual essay on a given subject. For instance, in the Introduction to the Ethics Computing course students wrote an essay on the ethical aspects of ICT.

In the *project* method, the students did project-work. For instance, in the Programming Project Course the students designed, implemented, tested and documented small applications individually or in pairs. The goal of the Programming Project Course was to familiarize students with a small-scale software development process. Table 7.2 summarizes the working methods created for ViSCoS courses within the first thread. These methods have stayed similar throughout the ViSCoS development.

Table 7.2: Summary of working methods in ViSCoS courses

	Assignment-based	Essay	Project	Study
Computers, Operating Systems and Networks	X			
Practical Use of Computers, Essential Applications and Network Services.	X			
Programming I	X			
Programming II	X			
Programming Project			X	
Design of Algorithms	X			
Introduction to the Ethics of Computing		X		
Research Fields of Computer Science				X
Hardware, Computer Architecture and Operating Systems	X			

Within the first thread, digital learning materials were created mainly for assignment-based courses. One of the design solutions was to link the digital material with printed material, e.g. course books. This meant that we did not have to produce all the materials in the digital format. In the assignment-based courses, digital learning materials consisted of the content of the course, examples and clarifying explanations in the form of graphics and animations, assignments, and visualizations. In other types of courses, the digital learning materials were mainly instructions. Within Thread 1, it was also recognized that there was a need for activating digital learning tools. However, concrete steps on creating and using the tools started within other threads.

Formative evaluation

The first versions of the courses were run between 2000 and 2001. While the first courses were running, feedback was collected from all parties involved. The experiences with Thread 1 were evaluated through a formative evaluation scheme (Sutinen & Torvinen, 2003). The research was focused on discovering the reasons for difficulties in studies. The evaluation indicated that Programming I and II courses seemed to be most difficult for students because most drop outs occurred in these courses (Torvinen, 2004).

7.2.2 Thread 2: Loop of improvements

Needs analysis

Experiences in Thread 1 led to the modifications in Thread 2. The focus on the courses was changed and two courses were merged into a single course. The course structures were also modified to focus on the most difficult issues: extra examples and exercises were created. The supportive efforts in Thread 2 were aimed at Programming I and II courses because experiences in Thread 1 indicated that these courses were the most difficult ones for students. It was also decided that studying the drop out phenomena would be the main focus within Thread 2.

Implementation

Concrete changes were made to the whole program as well as to individual courses. The amount of courses was dropped to eight. Computers, Operating Systems and Networks and Practical Use of Computers, Essential Applications and Network Services courses were merged to the course Introduction to ICT and Computing. There were also changes to the scope and schedule of the courses. Thread 1 indicated that semester 2 was too demanding for the students. Programming II course was too heavy; there was a need to shorten the content of the course. Hence, the Programming II course was reduced from 4.5 cps to 3 cps, and because the curriculum of ViSCoS had to remain 22.5 cps, the Introduction to Algorithms course was expanded from 1.5 cps to 3 cps. Some contents from the Programming II course were moved to the Introduction to Algorithms course. The schedule of the two courses was also changed. The Research Fields of Computer Science course was moved from semester 3 to semester 2 while the Hardware, Computer Architecture and Operating System course was moved from semester 2 to semester 3. These changes evened out the workload between the two semesters. Table 7.3 presents the curriculum of ViSCoS within Thread 2 (Torvinen, 2004).

Table 7.3: Scope and schedule of the ViSCoS courses within Thread 2

Course	ECTS	Schedule
Introduction to ICT and Computing	3	Semester 1
Programming I	3	Semester 1
Programming II	3	Semester 2
Research Fields of Computer Science	3	Semester 2
Programming Project	3	Semesters 2 and 3
Hardware, Computer Architecture and Operating Systems	3	Semester 3
Introduction to the Ethics of Computing	1.5	Semester 3
Introduction to Algorithms	3	Semester 3

Formative evaluation

The formative evaluation within Thread 2 focused especially on the programming courses. The aim was to reveal the reasons for dropping out of the program, and to improve the practices and learning materials (Meisalo et al., 2003b). Several studies were conducted in years 2000 - 2003 to study the drop out phenomena. The most recent studies, for instance, focused on analyzing the profile of the assignments in relation with their difficulty (Meisalo et al., 2004). The research method used has been action research. The aim was to both improve the courses and study the drop out phenomena. These two aims are in dynamic relation. New information about the drop out phenomena affected the design of the courses. On the other hand, new designs enabled to investigate which solutions might be appropriate for decreasing the number of drop outs in ViSCoS.

Pluralistic methods were used to study the drop out phenomena, such as questionnaires, interviews, log files, analysis of exams and submitted exercises, and analysis of student and tutor feedback (Torvinen, 2004). The main reasons for dropping out were lack of time, difficult exercises and failing the exams. Many students felt that it was difficult to study independently. They wished more support on the difficult topics of programming, such as arrays, methods, applets, and animations (Meisalo et al., 2002). The drop out research has also revealed that drop out rates in ViSCoS are at a similar level compared to other distance learning contexts (Meisalo et al., 2004). The drop out rate ranges in ViSCoS from zero up to 43% depending on the course (Sutinen & Torvinen, 2003).

Course arrangements and materials were changed according to the results of the drop out research. Based on the feedback given by students, the curriculum was modified so that there is more time for difficult topics. For instance, the amount of learning units in Programming I was increased from 10 weeks to 12 weeks within Thread 2 (Meisalo et al., 2004). At the same time, optional exercises and easier examples were created to most difficult topics, such as arrays, loops and methods (Sutinen & Torvinen, 2003). The Jeliot 2000 visualization tool (Ben-Bassat Levy et al., 2001), Flash animations and interactive applets were also created and used.

7.2.3 Thread 3: New technical solutions

The focus in Thread 3 is on creating new technical solutions to support learners. Thread 3 consists of four sub-threads. Three digital learning tools have either originated from ViSCoS or a ViSCoS context has played an important role in the development of the tool. Chapters 5 and 6 have already described the development of LEAP and how it related to ViSCoS. Other digital learning tools in Thread 3 are Ethicsar, ethical argumentation, and the Jeliot 3 visualization tool. The development of the tools affects the development of ViSCoS and vice versa. When the tools improve, it will have some affect on the support given to the students. The fourth sub-thread is the use of data mining methods for processing student data from ViSCoS courses. The data mining methods were focused, especially, on analyzing the data related to the assignments in the courses.

Ethicsar

The Ethicsar - ethical argumentation tool - was developed for the Ethics of Computing Course to support collaborative discussion and analysis of ethical problems (Jetsu et al., 2004). Interaction in Ethicsar is based on analyzing and commenting on peers' ethical opinions through visual means. Instructors can add ethical cases to the tool. Students can create answers to the cases and prepare a drawing line for the argumentation. They can also add their own cases. Students and instructors can observe how the discussion and different opinions have been created during the course of ethical cases. Extensive studies have not yet been conducted on the first user experiences with the tool. However, there are already some ideas on how to develop the tool further. The current Ethicsar has statistical information on the ethical arguments, such as the means and standard deviation (Jetsu et al., 2004).

This could be enhanced by showing animations or graphs of the argumentation process and differences among the answers.

Jeliot Family visualization tools

The Jeliot Family is a group of program visualization tools (Moreno & Myller, 2003). The development of Jeliot started in the beginning of the 1990s and several versions have been developed. Two recent versions of the tool, Jeliot 2000 and Jeliot 3 have been used in ViSCoS courses. Jeliot 3 is especially designed for supporting novice learners in learning procedural and object-oriented programming (Moreno et al., 2004b). Jeliot 3 visualizes the execution of Java programs by showing the current state of the program (e.g. methods, variables and objects) and animations of the expression evaluations (Moreno et al., 2004a). The students in ViSCoS have been using Jeliot 2000 and Jeliot 3 in assignments. They were also instructed to use Jeliot when they encounter difficulties during the courses. The development process of Jeliot and ViSCoS are intensively related to each other. The user experiences of Jeliot in ViSCoS have been directed to development of the program. A recent study with Jeliot in ViSCoS analyzed, for instance, the improvement ideas given by the students (Kannusmäki et al., 2004).

Data mining to process student data in ViSCoS

The third sub-thread within Thread 3 is data mining for processing student data. Online study programs, such as ViSCoS, include information about students' progress. The assignments, exam marks and activity logs are all possible data that can be processed with data mining techniques. Furthermore, students' skills and backgrounds, such as age or learning history can be utilized to support students individually. Some examples of using data mining in educational context exist, but it seems that it is not yet common (Chen et al., 2001a; Ma et al., 2000; Silva & Viera, 2002).

The assignments play an important role in ViSCoS. A typical course includes 40-50 assignments. Data mining techniques could be applied to process the data for several purposes. First, the students could be classified according to their individual characteristics. Secondly, a general overview of the student population and their success inside a course could be analyzed. This could help, for instance, to predict

those students who are likely to drop out from the course. Finally, information could be collected from several courses.

The first trials on using data mining methods in ViSCoS were conducted in year 2002 (Myller et al., 2002). The aim was to see how a set of standard data mining methods could be used with programming assignment data. We tried to form heterogeneous groups and predict students' performance in exams. The first trials did not produce reasonable results, but it was a beginning for the future efforts. The second concrete step was a construction of a model for processing exercise data in programming courses. The hybrid model combines both data mining and machine learning techniques in constructing a Bayesian network for describing the students' learning process. The goal of the model is to classify students so that individual guidance could be implemented.

7.3 Summary

Three threads in ViSCoS development were identified. Table 7.4 summarizes the components in each thread. The sub-threads in Thread 3 are presented in Figure 7.1.

Table 7.4: ViSCoS development

Th	Theme	NA	I	FE
1	First designs	Prioritization of the tasks, identifying the contextual factors of the high school students	First implementation of the program and courses	Analyze the feedback from all the participants
2	Loop of improvements	Identify the most difficult aspects of the programming courses	Formative development of the programming course	Drop out phenomenon
3	New technical solutions	Needs and requirements from the experience in ViSCoS	Digital learning tools, data mining to process student data	Improvement of the technical solutions and ViSCoS

As already noted, the three threads in ViSCoS development have several dependencies among each other. Thread 3 is divided into four sub-threads. Figure 7.1 visualizes threads, components and dependencies in ViSCoS development.

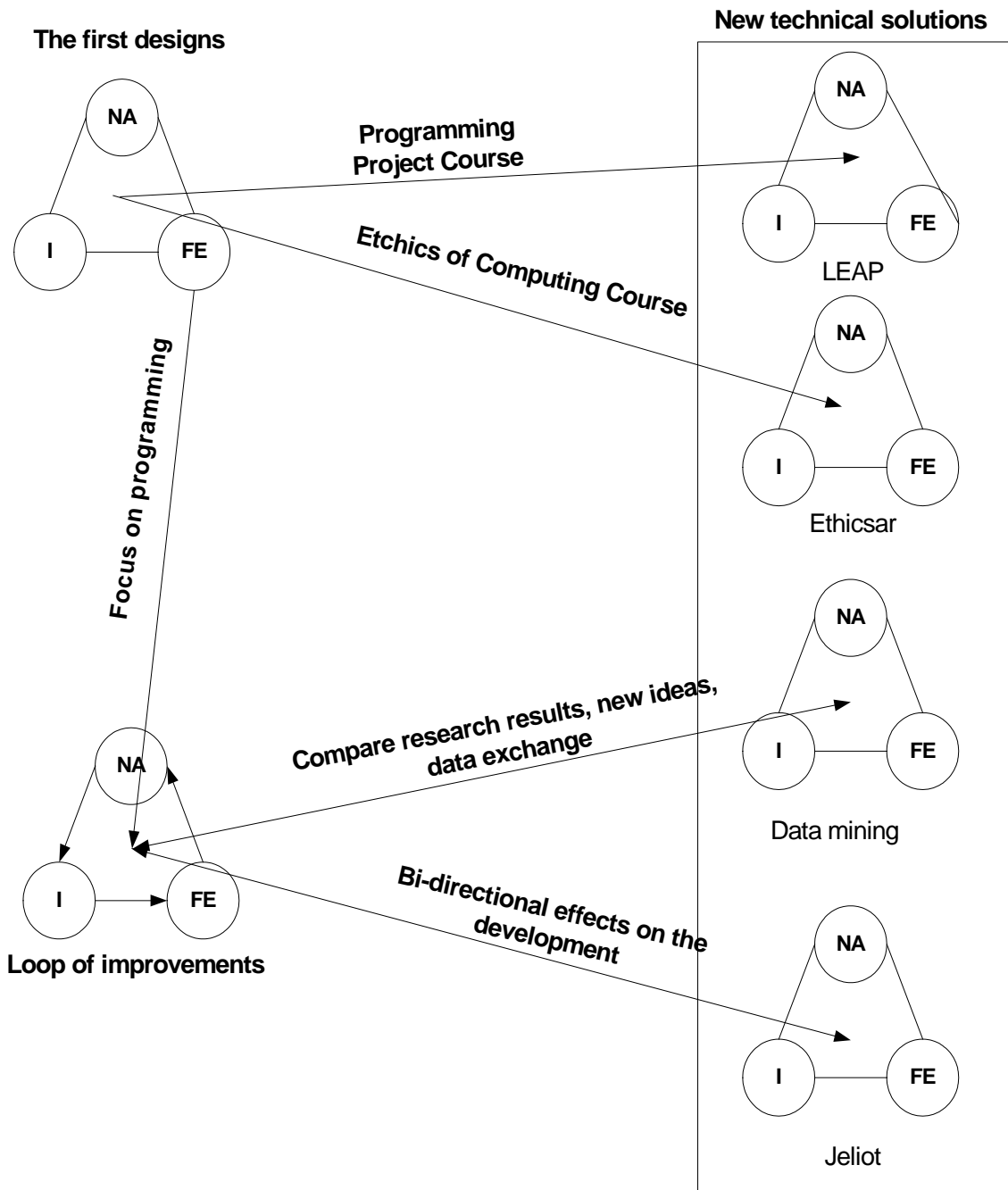


Figure 7.1: Visualization of the ViSCoS development process

As seen in Figure 7.1, there are several dependencies between Thread 3 and other threads. The Jeliot sub-thread is interesting. The development of Jeliot has started independently and years before ViSCoS. However, Jeliot has been a part of ViSCoS already from the beginning. These two separate development processes have joined together. Currently, both of these processes affect each other. When

there are new features and development in Jeliot, it can be used more efficiently in ViSCoS. On the other hand, when Jeliot is used in ViSCoS, the tool can be improved. Furthermore, as seen in Chapter 6, LEAP development can be divided into a set of sub-threads.

The first ViSCoS courses were running after three months design and implementation in September 2000. During the loop of improvements thread, courses and the overall program has been improved according to the results of formative evaluation. Fast changes were made when there was a need. Figure 7.2 summarizes the timeline of ViSCoS development.

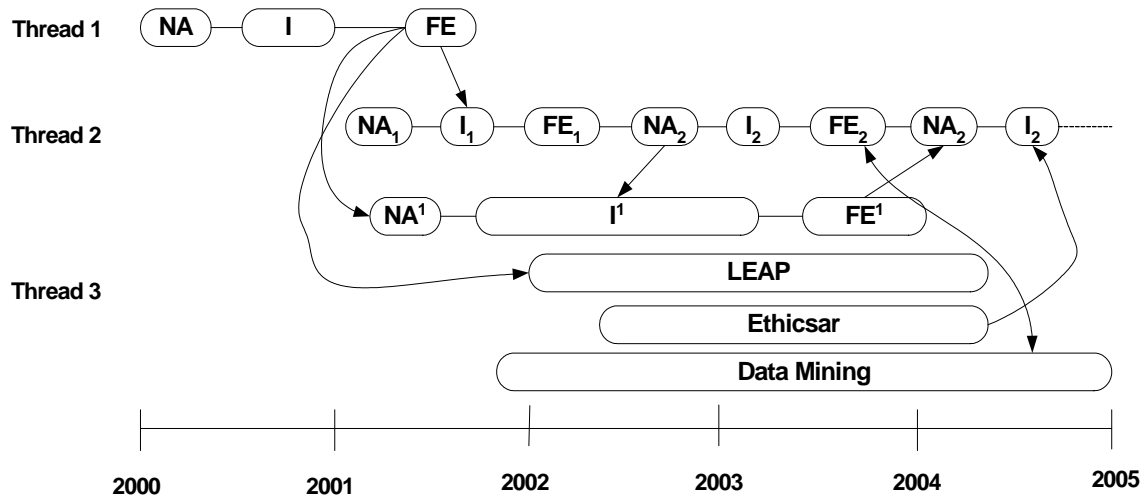


Figure 7.2: Timeline of ViSCoS development

8 Discussion

8.1 Results of the study

The action research framework presented in Chapter 2 was used to answer the research questions in the study. The first result is the construction of the FODEM method for developing digital learning environments in sparse learning communities. FODEM consists of three main components: (1) needs analysis, (2) implementation, and (3) formative evaluation. First, in needs analysis both theory and practice are used to define the functional specifications for the environment. Secondly, in implementation, fast prototyping is emphasized to allow testing the design ideas in authentic learning settings. Thirdly, formative evaluation is used to evaluate the use of the environment. In FODEM, the development process consists of simultaneous threads where the components interact extensively. Furthermore, the dependencies are used to represent the interdependent structure of threads, the interaction among components and the dependency between components in different threads. The aim in the dependencies is to represent the most important relationships. FODEM also enables different development processes to be interwoven together as seen with ViSCoS and Jeliot. A thread within a development process can be a part of another development process.

FODEM has been developed gradually following the action research framework. We started from the practical needs in sparse learning communities. At first, the general concepts of the method were applied. The specified features and concepts of the method were developed when applying FODEM to design LEAP and ViSCoS. It is likely that FODEM will be further modified and reshaped via the action research process. However, the thread-based approach will probably stay as a core feature of the method.

FODEM was successfully applied to develop the LEAP digital learning tool and digital learning environments in the ViSCoS online study program. The development of LEAP is still in the early stages. During the LEAP development the

core function of the tool was not changed, but modifications were made to the technical implementations. The LEAP development progressed in discrete steps; changes were not made continually but they were implemented as distinct units. Clear threads can be identified with a common theme. Three formative evaluation studies have indicated both positive and negative aspects of LEAP. The evaluation scheme used in the studies allowed me to compare the results of the studies. The main supposition from the results is that the ideas behind LEAP are viable, but the implementation needs refinement. The results indicate that the public steps and problem solving support activities need most improvement. Furthermore, the tool could be more incorporated to the context of use. Finally, the evaluation scheme produced concrete development ideas even for the core function of the tool. The interviews conducted in Studies 2 and 3 were not totally successful. The inadequate handling of difficult interviews can cause problems and affect the results (Cohen et al., 2002). I was not able to direct the interview situation so that more usable data would have been found. If I had been able to create a more relaxed atmosphere, the interview might have produced more useful information.

The ViSCoS case included many parts; the development did not proceed in clear steps, but changes were made in smaller parts. The development of ViSCoS has been mainly focused on improving the programming courses. Thread 3 in ViSCoS development represents the nature of FODEM. Different sub-threads can be identified with multiple dependencies. The ViSCoS program has been running for four years; it has been proven to be a flexible solution for studying first year computer science courses via the net. Altogether 94 students have completed the whole study program, most of them while still studying in high school.

According to the results of the cases, FODEM is applicable in two different digital learning environment development situations. In Section 3.2, the intersection of three variables, costs of development, applied design approaches, and meaningful results in the context of sparse learning communities, were defined as the criteria for evaluating the feasibility of a design method in sparse learning communities. Both ViSCoS and LEAP have been developed with reasonably low investments. The method has also enabled a quick start of the studies. Especially in the ViSCoS development, formative evaluation with pluralistic research methods has helped to fit the course arrangements and digital learning materials to students' needs and requirements (Sutinen & Torvinen, 2003). For instance, courses are implemented in

a light way enabling the students to use modem connections in their studies. A formative development process with drop out research focus has identified weaknesses in the program. The accumulation of experience from courses gradually widens the knowledge about the phenomena.

Furthermore, ViSCoS development has created “spin-offs” for designing new technical solutions for supporting first year CS online learning. Finally, the development process in FODEM follows a simple, yet structured, design method. The formative evaluation component ensures that the development process is both dynamic and focused. Figure 8.1 visualizes how FODEM, based on the results of LEAP and ViSCoS development, is located in relation to the design method criteria in sparse learning communities. The method requires some investments and provides semi-structured support for the development. ViSCoS results show that FODEM can produce contextualized digital learning environments.

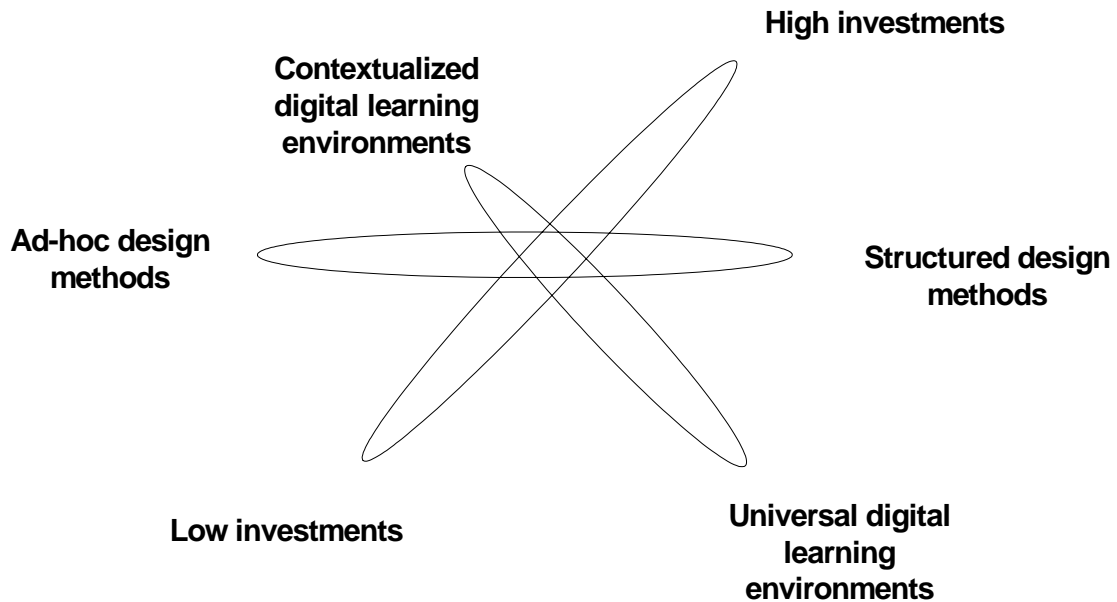


Figure 8.1: FODEM method related to the criteria in sparse learning communities

Figure 8.2 shows a characterization of Moonen’s three space design (from educational technology field) method in relation to the criteria for design methods in sparse learning communities (Moonen, 2002). Compared to FODEM, Moonen’s method is more complex and requires higher resource investments. Moonen’s method is also sequential in nature.

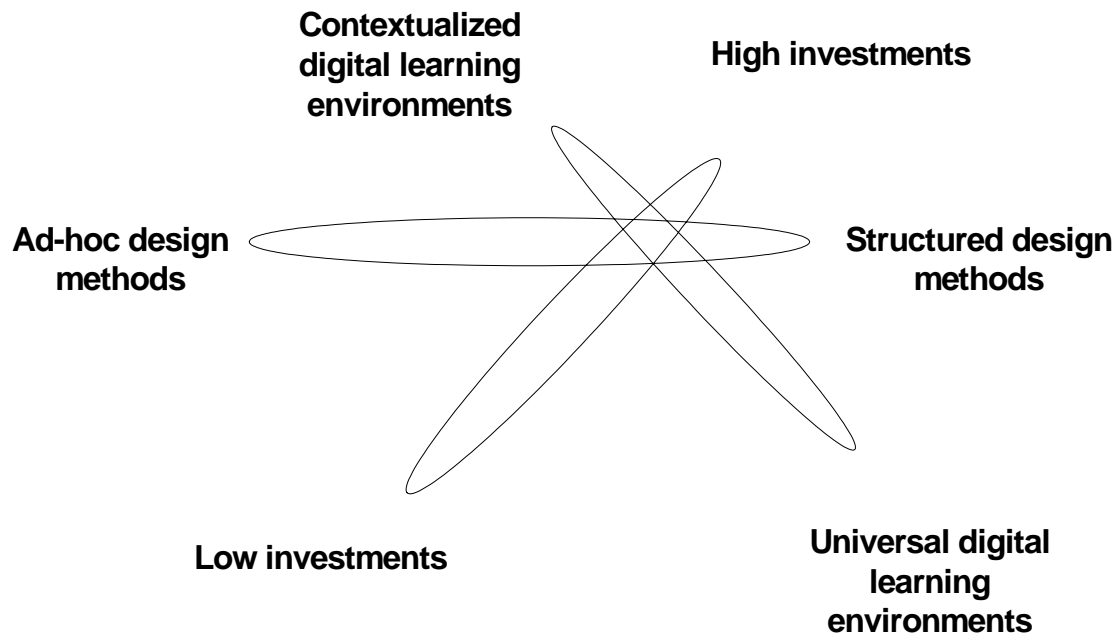


Figure 8.2: Moonen's three space design method related to the criteria for design methods in sparse learning communities

Figure 8.3 illustrates how structured software development methods relate to the evaluation criteria. The structured software methods require high investments, and the design methods are highly structured. The aim in structured software methods is more on producing universal results. Figures 8.1, 8.2 and 8.3 indicate that FODEM is more feasible compared to the two other methods in the context of sparse learning communities.

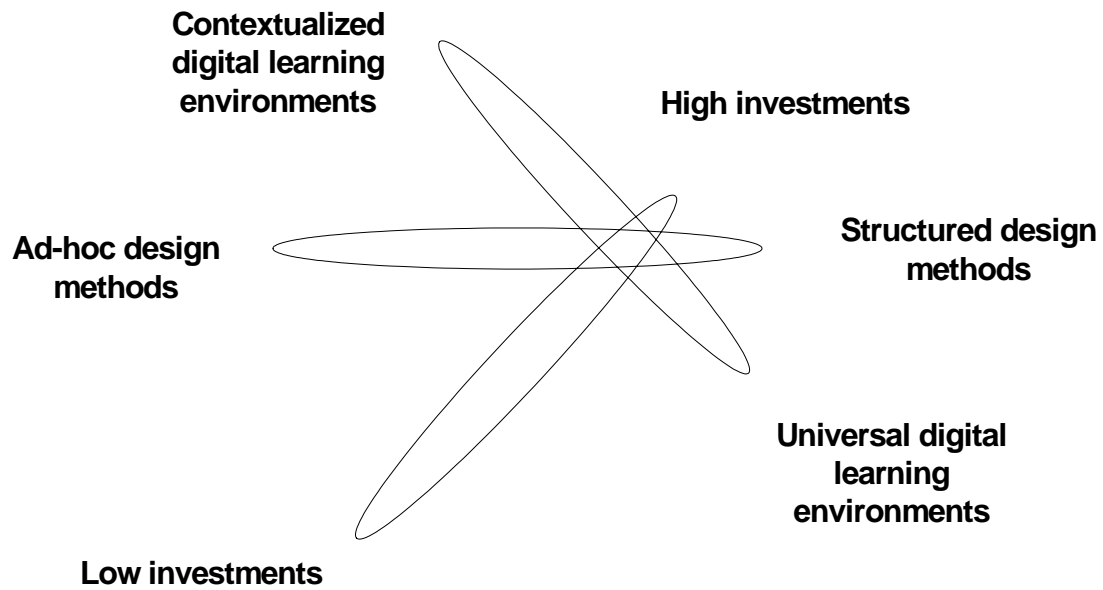


Figure 8.3: Structured software development methods in relation to the criteria for design methods in sparse learning communities

The other research questions in the study relate to the strengths and weaknesses of FODEM and the requirements for using the method. A strength of FODEM is the multithreaded structure that can be modeled as explained in Section 4.3. Modeling enables, for instance, a corresponding technical design environment to be created to support FODEM development process. Modeling also allows that the design process can be re-used. A designer could follow the component and thread structure of a similar situation. The different FODEM development processes can also be compared. FODEM is open and flexible in nature; threads and components can be created depending on the design context. Figure 8.1 depicts the other strengths of the method.

Although the generality of FODEM allows it to be applied in different situations, it is also a weakness of the method. The method itself does not offer any clear phases or steps that should be followed in a certain development situation. FODEM can be used in a structured way, but the method itself does not include clear instructions on how to use it. When the two development cases LEAP and ViSCoS are compared with each other, the results in ViSCoS development have been better. This indicates that FODEM is more efficient in the online study program development when the development situation is broader and more complex. For

instance, in a digital learning tool development a more focused version-based approach could produce better results.

One clear requirement for the use of FODEM can be identified in LEAP development. When the development ideas are tested in the early stages of development, the feedback received from the learners can be direct (as seen in Appendix D). It is important to block out any noise and try to figure out the real reasons for the problems when using the environment. Another requirement is that the development context should be open for testing the designs. Furthermore, there should be room for failure; it is an opportunity to learn from the mistakes and improve the function of the environment. According to Hydenrych (2004), FODEM types of methods might not work in situations where the development context does not allow innovations and there is no readiness for changes. Currently, it is also difficult to clearly specify the costs of threads and components in the development. This will depend on many factors, starting from the size of the development. Individual components can include procedures that have totally different costs. For instance, Fincher and Petre (2004) calculated that the costs of experimental studies are much higher than questionnaires. Hence, the costs are related to the development context. It is even difficult to assess the workload needed for components. The needs of a certain development context depend on what components are emphasized.

8.2 Evaluation of the research results

Different factors can be used to evaluate the research results of the descriptive, constructive and evaluative research questions as explained in Section 2.4. The descriptive research questions were answered through the literature reviews in Chapters 3 and 4. The literature in the study originates from three fields: education, educational technology and computing. The educational and educational technology references are related to establishing criteria for a functional design method for digital learning environments in sparse learning communities. The educational technology literature is also related to the existing development methods for digital learning environments. The main focus in the literature review was in currently published articles in scientific journals and conference proceedings. I selected those journal series that have been published for a longer time. Furthermore, I found the main references by going systematically through the latest proceedings of the following conferences: International Conference on Computers in Education,

International Conference on Advanced Learning Technologies and World Conference on Hypermedia, Multimedia and Telecommunications.

The computing literature is related to the software development methods. The computing literature is very diverse from conference proceedings to books. I found most of the computing literature in the digital libraries of IEEE (Institute of Electrical and Electronics Engineers) and ACM (Association for Computer Machinery). I consider these references to be rigorous and sound. Different sources are used in different parts of the thesis, but in some chapters all origins are represented. Altogether, the main weakness of the literature reviews is that they are not totally comprehensive. For instance, the literature review on different design methods could be deeper. A more comprehensive review of all the sources for all the topics would be beyond the scope of this study. The challenge was to combine the literature from different sources in a meaningful way.

The constructive research results are evaluated through the usefulness and pragmatic benefit of FODEM in sparse learning communities. Chapters 5, 6 and 7 describe that the method was successfully applied in two different development cases. The method produced useful and pragmatic results. Furthermore, Figure 8.1 illustrates how FODEM relates to the requirements in sparse learning community context.

The evaluative research questions were answered through two case studies of FODEM. The critical factors in the evaluative research methods are validity and reliability. As recommended by Baskerville and Wood-Harper (1996) and Creswell (2003), I presented the development process of LEAP as detailed and transparent as possible to show that the conclusions about the process make sense. I also described the methods used in different parts of the development, e.g. interview and questionnaire protocols and data analysis procedures (Steeple, 2004). The richness of data and multiple data sources were also used to increase the validity (Cohen et al., 2002). Because the ViSCoS case is presented on a higher abstraction level, the validity of the results can be questioned. It is possible that my perceptions about ViSCoS development are not accurate enough; the results do not present exact information about the processes and the decisions made during the development. However, the ViSCoS development case was based on published articles, for instance, about the drop out research. Furthermore, I have been personally involved

in the ViSCoS development. Therefore, in my opinion, the ViSCoS development process is presented realistically.

The two cases are used to increase the reliability of the FODEM evaluation (Miles & Huberman, 1994). However, because the two development cases are related to each other, the generalization is problematic. In fact, the development context is almost similar. FODEM was applied with the same group of people working in the same institution. The two cases show that FODEM works and it has been produced satisfactory results in this context. FODEM can be justified based on the two cases, but there is no guarantee that it also works in other contexts. Beside the development contexts similar to ViSCoS and LEAP, FODEM could also be applied, for instance, for small courses at rural schools.

If FODEM is used in other contexts, the requirements of the method explained earlier in the chapter should be considered. For instance, the use of environment with real learners may be expensive in some educational cultures. The constraints of FODEM give some indications on what kind of attitude is needed when using the method. For instance, the FODEM process needs an atmosphere where un-polished design ideas are tested. If the constraints and requirements are not met, FODEM might not be usable. It should also be noted that FODEM is developed for digital learning environments in sparse learning communities. Hence, the generalization of the research results to other contexts is left to the reader.

9 Conclusion and future work

The action research framework for developing the FODEM method consisted of three types of research questions. The descriptive research questions were

D. What are the needs for digital learning environments in sparse learning communities? What are the criteria for a functional design method? How does FODEM relate to other design methods?

In Chapter 3, the needs for digital learning environments in sparse learning communities are identified. The identified needs created the criteria for a functional design method in sparse learning communities. The intersection of the three variables: development costs, structure of the design approach and consideration of contextual factors can be used to evaluate the feasibility of a design method in the context of sparse learning communities. In Chapter 4, the literature review on sister design methods reveal that FODEM differed from the existing methods. The uniqueness of FODEM compared to its sister methods is the way how it models the parallel, but interdependent threads of the digital learning environment development. Furthermore, modeling of the FODEM enables to implement technical solutions to support the development process.

The constructive contribution of the study was the construction and definition of the FODEM method. The constructive research questions were;

C. What is FODEM? What are the core features of FODEM? How can FODEM be applied?

FODEM was defined in Chapter 4. The main concepts are process, components, threads and dependencies. FODEM is an action-oriented method where the development progresses through a formative process. The metadata description of FODEM allowed modeling of the development process. Finally, FODEM was successfully applied in two development cases as explained in Chapters 5, 6 and 7.

An important part of the action research framework was the evaluation of the FODEM. Evaluative research questions were;

E. What results has FODEM produced? What are the requirements for using the method? What are the strengths and weaknesses of the method? How can FODEM be improved?

The results of FODEM can be seen in Chapters 5, 6 and 7. The requirements of using the method, and its strengths and weaknesses were analyzed in Chapter 8. The main strength of FODEM is its dynamic nature. However, the method is still focused on the development of digital learning environments in sparse learning communities.

The question about improving FODEM has not yet been answered. Naturally, one of the main aims in FODEM development is to improve the weaknesses discussed in Chapter 8. However, it is important that the main features of the method, such as open and flexible nature, are not jeopardized. If specific steps or phases are created to enhance FODEM in a certain development situation, it will reduce the general usability of the method. The action research framework allows that the method can be refined when new experience is gathered about the use of the method in different settings. The components can change, or more defined sub-components will be created. Furthermore, the role and meaning of dependencies may also change. At this point the goal of the method is to represent the most important aspects of a development process. It might be feasible to go into details when modeling the dynamic relations between the components, threads and development processes. LEAP case showed that the occurring problems should be fixed quickly. As seen in Table 6.34 the changes made to LEAP between Thread 1 and 2 were minimal. This can be seen also in the results of the third evaluation.

A natural, new development step would be to create a full metadata schema based on the thread metadata description in Section 4.3. When the idea is developed further on, the application using the metadata description could offer tools for managing and analyzing all the data related to the development process. This is especially important in large digital learning environment development cases.

The metadata scheme of FODEM could also be used to analyze the costs of the development. The cost analysis could reveal, for instance, how various data analysis

methods in formative evaluation affect the total costs. This would help designers to decide what kinds of methods are economically suitable. The analysis could be enhanced, for instance, through information on how useful are the results that different study methods produce. This would help in deciding which methods produce the best results in certain development situations. It might be, for instance, that in the early stages of development the evaluation should produce concrete development ideas. In the later stages of development, the evaluation could concentrate on analyzing the real effects of the environment.

Because FODEM has been tightly linked to LEAP and ViSCoS development, they are also related to each other in future. The results of the LEAP development indicated that the current implementation of problem solving support activities and public steps are not working as planned. Intelligent features might remove some of the problems. It might be reasonable to implement the activities so that they are hidden from the students. For instance, intelligent question lists could be activated when the tool recognizes a need to provoke learners thinking. In the current LEAP implementation, different question list configurations can be created for different phases. Every time students add a new solution to a step in a phase, they are presented with the same questions. The simplest form of intelligence would be to assign new questions in a phase depending on the students' actions. The community of practice and peer support are also important aspects of problem solving that are not yet considered in LEAP (Shneiderman, 2000). Another possible future thread is the implementation of the provocative agent service defined in the pedagogical layer of LEAP. The provocative agent could change the tool into a more dynamic one. Currently, the core structure of the tool is a tree. The division among episodes, phases, steps and solutions is conventional and restrictive. The text fields in phases, steps and solutions are also static. A functional digital learning portfolio should be dynamic; it might be useful to have different text fields when new steps are added. The third immediate development step is the graphical support. Learners should be able to either add graphical files or draw inside the tool.

The ViSCoS program almost ended in 2004, but new financial resources enabled it to continue. There was also a change in the student population. ViSCoS is currently offered via the Open University of Joensuu to other partner institutions than high schools. ViSCoS has established a static position, but there is a constant need for improvements and new supportive solutions. Threads 2 and 3 presented in

Chapter 7 continue in the future. For instance, the drop out rate in the program is still quite high. It is also possible that new threads are formed when new ideas emerge. A concrete, already active, thread is the expansion of ViSCoS to other cultural contexts, such as developing countries. The first steps have already been taken through creation of the ViSCoS programming courses in the Tanzanian context. The Programming I course is also partly implemented in English, but it is too early to conceptualize the expansion of the program as a whole thread. Another future thread is the creation of *ViSCoS Mobile* where learners could use mobile devices in their studies.

The main outcome of the study is the FODEM method for developing digital learning environments in sparse learning communities. The action research framework behind FODEM development has enabled gradual development of the method. FODEM has produced fast results with low investments. FODEM can also be used to produce contextual digital learning environments. The model itself has proven to be a simple, yet structured, method for designing digital learning environments for sparse learning communities.

10 References

- Abdelraheem, A.Y. (2003). Computerized learning environments: Problems, design challenges and future promises. *The Journal of Interactive Online Learning*, **2**, 2, 1-9.
- Allison, A.J. (2002). An action research approach to the design and implementation of an on-line course in applied mechanics. In S. Banks, P. Goodyear, V. Hodgson, D. McConnell (Eds.), *Networked Learning 2002: A Researched Based Conference on E-Learning in Higher Education and Lifelong Learning*. Sheffield, United Kingdom: University of Sheffield, 289-297.
- Anderson, T., Wason, T. (2001). IMS metadata best practice and implementation guide: final specification, version 1.1. *IMS Global Learning Consortium, INC*. Retrieved, March 12, 2005, from <http://www.imsproject.org/metadata/mdbestv1p1.html>.
- Anfara, V.A., Brown, K.M., Mangione, T.L. (2002). Qualitative analysis on stage: Making the research process more public. *Educational Researcher*, **31**, 7, 28-38.
- Armitage, S., Bryson, M., Creanor, L., Higgison, C., Jenkins, M., Ringan, N., Newland, B., Prescott, D., Yip, H. (2004). Supporting learning technology: Relationships with research and theory. *Proceedings of the Networked Learning 2004*, April 5-7, Lancaster University, United Kingdom.
- Avgeriou, P., Retalis, S. (2002). Modelling web-based instructional systems. *Journal of Information Technology Education*, **1**, 1.
- Avison, D., Lau, F., Myers, M., Nielsen, P.A. (1999). Action research. *Communications of the ACM*, **42**, 1, 94-97.
- Avraamidou, L., Zembal-Saul, C. (2002). Making the case for the use of web-based portfolios in support of learning to teach. *The Journal of Interactive Open Learning*, **1**, 2, 1-19.

- Bahn, D.L., Naumann, J.D. (1997). Evolution versus construction: Distinguishing user review of software prototypes from conventional review of software specifications. *Proceedings of the ACM Conference on Computer Personnel Research (SIGCPR'97)*, April 3-5, San Francisco, CA, USA. New York: ACM Press, 240-245.
- Barab, S., Squire, K. (2004). Design-based research: Putting a stake in the ground. *The Journal of the Learning Sciences*, **13**, 1, 1-14.
- Baskerville, R.L. (1999). Investigating information systems with action research. *Communications of the Association for Information Systems*, **2**, 1999.
- Baskerville, R.L., Wood-Harper, A.T. (1996). A critical perspective on action research as a method for information system research. *Journal of Information Technology*, **11**, 235-246.
- Bayer, H., Holtzblatt, C. (1998). *Contextual Design*. San Francisco, CA: Morgan Kaufmann Publishers.
- Bayer, H., Holtzblatt, C. (1999). Contextual design. *Interactions*, Jan/Feb, 32-42.
- Ben-Ari, M., Myller, N., Sutinen, E., Tarhio, J. (2002). Perspectives on program animation with Jeliot. *Software Visualization: International Seminar*. Dagstuhl Castle, Germany. Lecture Notes in Computer Science 2269, Springer Verlag, 31-45.
- Ben-Bassat Levy, R., Ben-Ari, M., Uronen, P. (2001). An extended experiment with Jeliot 2000. *Proceedings of the First Program Visualization Workshop*, July 7-8, Porvoo, Finland. University of Joensuu, Finland, 131-140.
- Bergman, T. (1999). Networking for the self-directed learner in the digital age. In P. Linnakylä, M. Kankaanranta, J. Bopry (Eds.), *Portfolios on the Web - Portfolioita Verkossa*. Jyväskylä, Finland: Institute for Educational Research, 16-35.
- Bork, A., Gunnarsdottir, S. (2001). *Tutorial Distance Learning: Rebuilding Our Educational System*. New York: Kluwer Academic/Plenum Publishers.
- Brickel, D., Harper, B., Ferry, B. (2002). Developing on-line tools to support learners in problem-solving activities. *Proceedings of the International Conference on Computers in Education (ICCE2002)*, December 3-6, Auckland, New Zealand. Los Alamitos, CA: IEEE Computer Society, 1058-1062.

- Brockbank, A., McGill, I. (1998). *Facilitating Reflective Learning in Higher Education*. Buckingham, England: Society for Research into Higher Education & Open University Press.
- Brush, T., Saye, J. (2001). The use of embedded scaffolds with hypermedia-supported student-centered learning. *Journal of Educational Multimedia and Hypermedia*, **10**, 4, 333-356.
- Butler, M.H. (2001). *Current technologies for device independence*. Hewlett-Packard Publications, Publishing Systems and Solutions Laboratory, HP Laboratories Bristol. Retrieved January 25, 2005 from <http://www.hpl.hp.com/techreports/2001/HPL-2001-83.pdf>.
- Castro, M., López-Rey, A., Pérez-Molina, C.M., Colmenar, A., de Mora, C., Yeves, F., Carpio, J., Peire, J., Daniel, J.S. (2001). Examples of distance learning projects in the European Community. *IEEE Transactions on Education*, **44**, 4, 406-410.
- Chan, T.W., Hue, C.-W., Chou, C.-Y., Tzeng, O.J.L. (2001). Four spaces of network learning models. *Computers & Education*, **37**, 141-161.
- Chang, C.-C. (2001). A study on the evaluation and effectiveness analysis of web-based learning portfolio (WBLP). *British Journal of Educational Technology*, **32**, 4, 435-458.
- Chen, C.F., Lin, M.H., Tzeng, T.-S., Yeh, C.-J., Wang, T.P. (2001a). The application of data mining to on-line education and training. *Proceedings of the International Conference on Computers in Education (ICCE/Schoolnet2001)*, November 12-15, Seoul, Korea, 1350-1353.
- Chen, J., Zhou, B., Shi, J., Zhang, H., Fengwu, Q. (2001b). Function-based object model towards website adaptation. *Proceedings of the 10th World Wide Web Conference*, May 1-5, Hong Kong. New York: ACM Press, 587-596.
- Clear, T. (2001). Research paradigms and the nature of meaning and truth. *SIGCSE Bulletin*, **33**, 2, 9-10.
- Clear, T. (2004). Critical enquiry in computer science education. In S. Fincher, M. Petre (Eds.), *Computer Science Education Research*. London: RoutledgeFalmer, 101-126.

Cohen, L., Manion, L. (1989). *Research Methods in Education (2nd Ed.)*. London: Routledge.

Cohen, L., Manion, L., Morrison, K. (2002). *Research Methods in Education (5th Ed.)*. London and New York: RoutledgeFalmer.

Conole, G., Oliver, M., Istroff, K., Ravenscroft, A. (2004). Addressing methodological issues in e-learning research. *Proceedings of the Networked Learning 2004*, April 5-7, Lancaster University, United Kingdom.

Creswell, J.W. (1994). *Research Design: Qualitative and Quantitative Approaches*. Thousand Oaks, CA: Sage.

Creswell, J.W. (2003). *Research Design: Qualitative, Quantitative, and Mixed Method Approaches (2nd Ed.)*. London: Sage Publications.

Deek, F.P., Turoff, M., McHugh, J. (1999). A common model for problem solving and program development. *IEEE Transactions on Education*, **42**, 4, 331-336.

Dennis, A., Wixom, B.H. (2003). *Systems Analysis & Design (2nd Ed.)*. John Wiley & Sons.

Design-Based Research Collective. (2003). Design-based research: An emerging paradigm for educational inquiry. *Educational Researcher*, **32**, 1, 5-8.

Duffy, T.M., Cunningham, D.J. (2001). Constructivism: Implications for the design and delivery of instruction. In D.H. Jonassen (Ed.), *Handbook of Research for Educational Communications and Technology*. The Association for Educational Communications and Technology, 170-198.

Ehrmann, S.C. (2004). Asking the right question: What does research tell us about technology and higher education. Annenberg/CPB. Retrieved, December 28, 2004 from <http://www.learner.org/edtech/rscheval/rightquestion.html>.

Eisenstein, J., Vanderdonckt, J., Puerta, A. (2000). Adapting to mobile contexts with user-interface modeling. *Proceedings of the 3rd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'00)*, December 7-8, Monterey, California, USA. Los Alamitos, CA: IEEE Computer Society, 83-94.

- Estell, J.K. (2001). IPP: A web-based interactive programming portfolio. *Proceedings of the Thirty-Second SIGCSE Technical Symposium on Computer Science Education (SIGCSE 2001)*, February 21-25, Charlotte, NC, USA. New York: ACM Press, 149-153.
- Fallman, D. (2003). Design-oriented Human-Computer Interaction. *Proceedings of the Conference on Human Factors in Computing Systems (CHI2003)*, April 5-10, Ft. Lauderdale, Florida, USA. New York: ACM Press, 225-232.
- Farrel, G.M. (2001). The changing faces of virtual education. Commonwealth of Learning. Retrieved, March 29, 2005, from <http://www.col.org/virtualed/>.
- Fetherston, T. (2001). Pedagogical challenges for the World Wide Web. *Educational Technology Review*, **9**, 1.
- Fincher, S., Petre, M. (2004). The field and endeavour. In S. Fincher, M. Petre (Eds.), *Computer Science Education Research*. London: RoutledgeFalmer, 1-81.
- Gathercoal, P., Love, D., Bryde, B., McKean, G. (2002). On implementing web-based electronic portfolios. *Educause Quarterly*, **25**, 2, 29-37.
- Gerdt, P., Kommers, P., Looi, C.-K., Sutinen, E. (2001). Woven Stories as a cognitive tool. In M. Beynon, C. L. Nehaniv, K. Dautenhahn (Eds.), *Cognitive Technology: Instruments of Mind, 4th International Conference (CT2001)*, August 6-9, Warwick, United Kingdom. Lecture Notes in Computer Science 2117, Springer-Verlag, 233-247.
- Gerdt, P., Kurhila, J., Meisalo, V., Suhonen, J., Sutinen, E. (2002). Rambling through the wilds: A concept of Jagged Study Zones. *Proceedings of the World Conference on Educational Multimedia, Hypermedia & Telecommunications (ED-MEDIA2002)*, June 26-29, Denver, Colorado, USA. Association for the Advancement of Computing in Education, 596-601.
- Glass, R.L., Ramesh, V., Vessey, I. (2004). An analysis of research in computing disciplines. *Communications of the ACM*, **47**, 6, 89-94.
- Govender, D. (2004). Learning support: Using information communication technologies (ICTS) to scaffold for success. In J. Heydenrych, D. Govender (Eds.), *Distance Learning Technologies in Context - Selected Papers on Agent, Process and*

Product at the University of South Africa. University of South Africa, Pretoria: Unisa Press, 45-54.

Gredler, M.E. (1997). *Learning and Instruction – Theory into Practice (3rd Ed.)*. Upper Saddle River, New Jersey: Prentice-Hall.

Greening, T. (1997). Paradigms for educational research in computer science. *Proceedings of the Second Australasian Conference on Computer Science Education*, July 2-4, Melbourne, Australia, 47-51.

Haataja, A., Kontkanen, S., Suhonen, J., Sutinen, E. (2001a). Teaching university-level computer science to high school students over the web. *Proceedings of the World Conference on Hypermedia, Multimedia and Telecommunications (ED-MEDIA2001)*, June 25-30, Tampere, Finland. Association for the Advancement of Computing in Education, 630-635.

Haataja, A., Suhonen, J., Sutinen, E. (2001b). How to learn introductory programming over web. *Informatica*, **25**, 2, 165-171.

Haataja, A., Suhonen, J., Sutinen, E., Torvinen, S. (2001c). High schools students learning computer science over the web. *Interactive Multimedia Electronic Journal of Computer Enhanced Learning (IMEJ)*, **3**, 2. Available in URL: <http://imej.wfu.edu/articles/2001/2/index.asp>.

Hallnäs, L., Redström, J. (2001). Slow technology - designing for reflection. *Personal and Ubiquitous Computing*, **5**, 3, 201-212.

Hara, N., Bonk, C. J., Angeli, C. M. (2000). Content analysis of on-line discussion in an applied educational psychology course. *Instructional Science*, **28**, 115-152.

Heiskanen, A., Newman, M. (1997). Bridging the gap between information systems research and practice: The reflective practitioner as a researcher. *Proceedings of the 18th International Conference on Information Systems*, December 14-17, Atlanta, Georgia, USA. New York: ACM Press, 121-132.

Herrington, J., Oliver, R. (2002). Designing for reflection in online courses. *Proceedings of the Higher Education Research and Development Society of Australasia (HERDSA2002)*, July 7-10, Perth, Australia, 312-319.

- Heydenrych, J. (2004). Rethinking learning development: Empowerment and change through collaboration. In J. Heydenrych, D. Govender (Eds.), *Distance Learning Technologies in Context - Selected Papers on Agent, Process and Product at the University of South Africa*. University of South Africa, Pretoria: Unisa Press, 5-22.
- Hoffer, J.A., George, J.F., Valaich, J.S. (1999). *Modern Systems Analysis and Design (2nd Ed.)*. Addison Wesley Longman.
- Hwang, Y., Kim, J., Seo, E. (2003). Structure-aware web transcoding for mobile devices. *IEEE Internet Computing*, September-October, Los Alamitos: IEEE Computer Society, 14-21.
- Häkkinen, P. (2002). Challenges for design of computer-based learning environments. *British Journal of Educational Technology*, **33**, 4, 461-469.
- Jetsu, I., Meisalo, V., Myller, N., Sutinen, E. (2004). Ethical argumentation with Ethicsar. *Proceedings of the IADIS International Conference on Web Based Communities*, March 24-26, Lisbon, Portugal, 255-261.
- Jonassen, D.H. (1997). Instructional design models for well-structured and ill-structured problem-solving learning outcomes. *Educational Technology: Research and Development*, **45**, 1, 65-95.
- Jonassen, D. H. (1999). Designing constructivist learning environments. In C. M. Reigeluth (Ed.), *Instructional Design Theories and Models: A New Paradigm of Instructional Theory (Vol. II)*. New Jersey: Lawrence Erlbaum Associates, 215-239.
- Jonassen, D.H. (2000). *Computers as Mindtools for Schools (2nd Ed.)*. Upper Saddle River, New Jersey: Prentice-Hall.
- Jonassen, D.H., Reeves, T.C. (2001). Learning with technology: Using computers as cognitive tools. In D.H. Jonassen (Ed.), *Handbook of Research for Educational Communications and Technology*. The Association for Educational Communications and Technology, 1171-1195.
- Jones, M., Marsden, G., Mohd-Nasir, N., Boone, K., Buchanan, G. (1999). Improving web interaction on small displays. *Proceedings of the 8th International World Wide Web Conference*, May 11-14, Toronto, Canada.

Juuti, K. (2005). *Physics Teaching and Learning Design Research Approach*. Academic Dissertation. University of Helsinki, Faculty of Behavioural Sciences, Department of Applied Sciences of Education, Finland: Yliopistopaino.

Kainulainen, V. (2004). *Computer-Aided Creative Solving - Experiments with PPA*. Masters Thesis. University of Joensuu, Department of Computer Science, Finland.

Kainulainen, V., Suhonen, J., Sutinen, E., Goh, T., Kinshuk. (2004). Mobile digital portfolio extension. *Proceedings of the 2nd IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE2004)*, March 23-25, JungLi, Taiwan. Los Alamitos, CA: IEEE Computer Society, 98-102.

Kannusmäki, O., Moreno, A., Myller, N., Sutinen, E. (2004). What a novice wants: Students using program visualization in distance programming course. In A. Korhonen (Ed.), *Proceedings of the Third Program Visualization Workshop*, Research Report CS-RR-407, Department of Computer Science, University of Warwick, United Kingdom, 126-133.

Kareinen, A., Myller, N., Suhonen, J., Sutinen, E. (2002). Visuals first approach in an introductory web-based programming course. *Proceedings of the Second Program Visualization Workshop*, June 27-28, Aarhus, Denmark, 17-22.

Kareinen, A., Suhonen, J., Sutinen, E. (2001). Graphical and event-driven approach to teach introductory programming. *Informatika*, **38**, 2, 25-38.

Kerne, A. (2002). Concept-context-design: A creative model for the development of interactivity. *Proceedings of the Fourth Conference on Creativity & Cognition (C&C'02)*, October 14-16, Loughborough, Leicester, United Kingdom. New York: ACM Press, 192-199.

Kinshuk, Goh T. T. (2003). Mobile adaptation with multiple representation approach as educational pedagogy. In W. Uhr, W. Esswein, E. Schoop (Eds.) *Wirtschaftsinformatik 2003 - Medien - Märkte - Mobilität*, September 17-19, Dresden, Germany. Heidelberg, Germany: Physica-Verlag, 747-763.

Kinshuk, Suhonen, J., Sutinen, E. (2004). Mobile adaptation extension of a digital learning tool. *Proceedings of the International Conference on Computers in Education (ICCE2004)*, November 30-December 3, Melbourne, Australia.

Melbourne, Australia: Common Ground Publishing and RMIT University, 1581-1590.

Kinshuk, Suhonen, J., Sutinen, E., Goh, T. (2003). Mobile technologies in support of distance learning. *Asian Journal of Distance Education*, **1**, 1, 60-68.

Kock, N.F., McQueen, R.J., Scott, J.L. (1997). Can action research be made more rigorous in a positivist sense? The contribution of an iterative approach. *Journal of Systems and Information Technology*, **1**, 1, 1-24.

Kommers, P.A.M. (2004). Concepts in the world of the mind. In P.A.M. Kommers (Ed.), *Conceptual Support Systems for Learning - Imagining the Unknown*. Amsterdam, The Netherlands: IOS Press, 3-30.

Kurhila, J. (2003a). *Considering Individual Differences in Computer-Supported Special and Elementary Education*. Doctoral Dissertation. University of Helsinki, Department of Computer Science, Faculty of Science, Finland: Helsinki University Printing House.

Kurhila, J. (2003b). Tools to support on-line communities for learning. *Proceedings of the 3rd IEEE International Conference on Advanced Learning Technologies (ICALT'03)*, July 9-11, Athens, Greece. Los Alamitos, CA: IEEE Computer Society, 518-519.

Landry, M., Banville, C. (1992). A disciplined methodological pluralism. *Accounting, Management and Information Technologies*, **2**, 2, 77-97.

Lanzig, J. (2004). Everything you wanted to know about ... Concept mapping. In P.A.M. Kommers (Ed.), *Conceptual Support Systems for Learning - Imagining the Unknown*. Amsterdam, The Netherlands: IOS Press, 47-72.

Loh, B., Radinsky, J., Russel, E., Gomez, L.M., Reiser, B.R., Edelson, D.C. (1998). The progress portfolio: Designing reflective tools for a classroom context. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'98)*, April 18-23, Los Angeles, California, USA. New York: ACM Press, 627-634.

- Lowyck, J. (2002). Pedagogical design. In A.A. Adelsberger, B. Collins, J.M. Pawlowski (Eds.), *Handbook on Information Technologies for Education and Training*. Springer-Verlag, 199-217.
- Luchini, K., Quintana, C., Soloway, E. (2003). Pocket PiCoMap: A case study in designing and assessing a handheld concept mapping tool for learners. *Proceedings of the Conference on Human Factors in Computing Systems (CHI2003)*, April 5-10, Ft. Lauderdale, Florida, USA. New York: ACM Press, 321-328.
- Löwgren, J. (1995). Applying design methodology to software development. *Proceedings of the Conference on Designing Interactive Systems: Processes, Practices, Methods & Techniques (DIS'95)*, August 23-25, Ann Arbor, MI, USA. New York: ACM Press, 87-94.
- Ma, Y., Liu, S.L.B., Yu, P., Wong, C. (2000). Targeting the right students using data mining. *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 20-23, Boston, Massachusetts, USA. New York: ACM Press, 457-464.
- Marlow-Ferguson, R. (Ed.). (2002). *World Education Encyclopedia – A Survey of Educational Systems Worldwide (2nd Ed.)*. Gale Group, USA.
- Masui, C., DeCorte, E. (1999). Enhancing learning and problem solving skills: Orienting and self-judging, two powerful and trainable learning tools. *Learning and Instruction*, **9**, 517-542.
- McCracken, W.M. (2004). Research on learning to design software. In S. Fincher, M. Petre (Eds.), *Computer Science Education Research*. London: RoutledgeFalmer, 155-173.
- McCrinkle, A.A, Christensen, C.A. (1995). The impact of learning journals on metacognitive and cognitive processes and learning performance. *Learning and Instruction*, **5**, 167-185.
- McKenna, P., Laycock, B. (2004). Constructivist or insructivist: Pedagogical concepts practically appield to a computer learning environment. *Proceedings of the 9th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE2004)*, June 28-30, Leeds, United Kingdom. New York: ACM Press, 166-170.

- McLoughlin, C., Hollingworth, R. (2001). Computer supported problem solving: Enhancing thinking skills in science. *Proceedings of the World Conference on Educational Multimedia, Hypermedia & Telecommunication (ED-MEDIA2001)*, June 25-30, Tampere, Finland. Association for the Advancement of Computing in Education, 1293-1298.
- Meisalo, V., Sutinen, E., Tarhio, J. (2003a). *Modernit Oppimisympäristöt (2nd Ed.)*. Helsinki, Finland: Tietosanoma Oy.
- Meisalo, V., Sutinen, E., Torvinen, S. (2002). How to improve a virtual programming course. *Proceedings of 32nd ASEE/IEEE Frontiers in Education Conference (FIE2002)*, November 6-9, Boston, MA, USA.
- Meisalo, V., Sutinen, E., Torvinen, S. (2003b). Choosing appropriate methods for evaluating and improving the learning process in distance programming courses. *Proceedings of 33rd ASEE/IEEE Frontiers in Education Conference (FIE2003)*, November 5-8, Boulder, CO, USA.
- Meisalo, V., Sutinen, E., Torvinen, S. (2004). Classification of exercises in a virtual programming course. *Proceedings of the 34th Frontiers in Education Conference (FIE2004)*, October 20-23, Savannah, Georgia, USA.
- Merriam, S.B. (1998). *Qualitative Research and Case Study Applications in Education*. San Francisco: Jossey-Bass Publishers.
- Miles, M.B., Huberman, A.M. (1994). *Qualitative Data Analysis: An Expanded Sourcebook (2nd. Ed.)*. London: Sage Publications.
- Mioduser, D., Nachmias, R. (2002). WWW in education. In A.A Adelsberger, B. Collins, J.M. Pawlowski (Eds.), *Handbook on Information Technologies for Education and Training*. Springer-Verlag, 23-44.
- Moallem, M. (2001). Applying constructivist and objectivist learning theories in the design of a web-based course: Implications for practice. *Educational Technology & Society*, **4**, 3, 113-125.
- Monash, R., Smith, J.R., Li, C.-S. (1999). Adapting multimedia internet content for universal access. *IEEE Transactions on Multimedia*, **1**, 1, 104-114.

- Moonen, J. (2002). Design methodology. In A.A Adelsberger, B. Collins, J.M. Pawlowski (Eds.), *Handbook on Information Technologies for Education and Training*. Springer-Verlag, 153-180.
- Moreno, A., Myller, N. (2003). Producing an educationally effective and usable tool for learning - The case of the Jeliot Family. *Proceedings of the International Conference on Networked e-learning for European Universities*, November 23-25, Granada, Spain.
- Moreno, A., Myller, N., Sutinen, E. (2004a). Collaborative program visualization with Woven Stories and Jeliot 3. *Proceedings of the IADIS International Conference on Web Based Communities*, March 24-26, Lisbon, Portugal, 482-485.
- Moreno, A., Myller, N., Sutinen, E., Ben-Ari, M. (2004b). Visualizing programs with Jeliot 3. *Proceedings of the International Working Conference on Advanced Visual Interfaces (AVI2004)*, May 25-28, Gallipoli, Italy. New York: ACM Press, 373-376.
- Myers, M.D. (1999). Investigating information systems with ethnographic research. *Communications of the Association for Information Systems*, **2**, 1-20.
- Myller, N., Suhonen, J., Sutinen, E. (2002). Using data mining for improving web-based course design. *Proceedings of the International Conference on Computers in Education (ICCE2002)*, December 3-6, Auckland, New Zealand. Los Alamitos, CA: IEEE Computer Society, 959-963.
- Norman, A.D. (2002). *The Design of Everyday Things*. New York: Basic Books.
- Nulden, U., Scheepers, H. (1999). Interactive multimedia and problem based learning: Experiencing project failure. *Journal of Educational Multimedia and Hypermedia*, **8**, 2, 189-215.
- Phan, T., Zorpas, G., Bagrodia, R. (2002). An extensible and scalable content adaptation pipeline architecture to support heterogeneous clients. *Proceedings of the 22nd International Conference on Distributed Computing Systems (ICDCS02)*, July 2-5, Vienna, Austria. Los Alamitos, CA: IEEE Computer Society, 507-516.
- Pintrich, P.R. (2002). The role of metacognitive knowledge in learning, teaching, and assessing. *Theory Into Practice*, **41**, 4, 119-225.

- Pitkänen, S.H., Silander, P. (2004). Criteria for pedagogical reusability of learning objects enabling adaptation and individualised learning processes. *Proceedings of the 4th IEEE International Conference on Advanced Learning Technologies (ICALT2004)*, August 30 – September 1, Joensuu, Finland. Los Alamitos, CA: IEEE Computer Society, 246-250.
- Polsani, P.R. (2003a). The liquid spaces of learning: Architecture and the knowledge ecologie. *Proceedings of the Fourth International Conference on Information Communication Technologies in Education*, Samos, Greece.
- Polsani, P.R. (2003b). Network learning. In K. Nyíri (Ed.) *Communications in the 21st Century: Essays on Philosophy, Psychology, Education*. Vienna: Passagen Verlag, 139-149.
- Popkewitz, T.S. (1984). *Paradigm & Ideology in Educational Research - the Social Functions of the Intellectual*. The Falmer Press.
- Quintana, C., Krajick, J., Soloway, E. (2002). A case study to distill structural scaffolding guidelines for scaffolded software environments. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Changing Our World, Changing Ourselves (CHI'2002)*, April 20-25, Minneapolis, Minnesota, USA. New York: ACM Press, 81-88.
- Ratner, C. (2002). Subjectivity and objectivity in qualitative methodology. *Forum Qualitative Sozialforschung / Forum: Qualitative Social Research*, **3**, 3. Retrieved, March 29, 2005, from <http://www.qualitative-research.net/fqs-texte/3-02/3-02ratner-e.htm>.
- Rautama, E., Sutinen, E., Tarhio, J. (2001). Web tools for creative problem solving. *Proceedings of the International Conference on Computers in Education (ICCE/Schoolnet2001)*, November 12-15, Seoul, Korea, 524-530.
- Rose, J. (2000). *Information Systems Development as Action Research - Soft Systems Methodology and Structuration Theory*. PhD Thesis, Management School, Lancaster University, England.
- Schroeder, U. (2002). Meta-learning functionality in elearning systems. *Proceedings of the International Conference on Advances in Infrastructure for Electronic*

Business, Education, Science, and Medicine on the Internet 2002 (SSGRR2002W), January 2002, L'Aquila, Italy.

Shneiderman, B. (2000). Creating creativity: User interfaces for supporting innovation. *ACM Transactions on Computer-Human Interaction*, **17**, 1, 114-138.

Silva, D., Viera, M. (2002). Using data warehouse and data mining resources for ongoing assessment of distance learning. *Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT2002)*, September 9-12, Kazan, Russia. Los Alamitos, CA: IEEE Computer Society, 40-45.

Soloway, E., Jackson, S.L., Klein, J., Quintana, C., Reed, J., Spitulnik, J., Stratford, S.J., Struder, S., Eng, J., Scala, N. (1996). Learning theory in practice: Case studies of learner-centered design. *Proceedings of the Computer Human-Interaction Conference (CHI'96)*, April 13-18, Vancouver, BC, Canada. New York: ACM Press, 189-196.

Sommerville, I. (1998). *Software Engineering (3rd Ed.)*. Addison-Wesley Publishing Company.

Steeple, C. (2004). Using action-oriented or participatory research methods for research on networked learning. *Proceedings of the Networked Learning 2004*. April 5-7, Lancaster University, England.

Stemler, S. (2001). An overview of content analysis. *Practical Assessment, Research & Evaluation*, **7**, 17. Retrieved August 12, 2004 from <http://PAREonline.net/getvn.asp?v=7&n=17>.

Storey, M.-A., Phillips, B., Maczewski, M., Wang, M. (2002). Evaluating the usability of web-based learning tools. *Educational Technology & Society*, **5**, 3. Retrieved, February 15, 2005 from http://ifets.ieee.org/periodical/vol_3_2002/storey.html.

Stoyanova, N., Kommers, P. (2002). Concept mapping as medium for shared cognition in computer-supported collaborative problem solving. *Journal of Interactive Learning Research*, **13**, 1/2, 111-133.

Sugiyama, T., Kakehi, N., Kura, T., Takahashi, T. (2002). CoCoFolio: A web-based electronic portfolio for enriching students learning by collaboration. *Proceedings of*

the World Conference on E-Learning in Corporate, Government, Healthcare & Higher Education (E-Learn2002), October 15-19, Montreal, Canada. Association of the Advancement of Computing in Education, 916-921.

Suhonen, J. (2005). Transcriptions of the interview, extra assignment and questionnaire data within the LEAP development. Available from the author on request.

Suhonen, J., Sutinen, E. (2002). Creative problem management in web-tutored programming projects. *Proceedings of the International Conference on Computers in Education (ICCE2002)*, December 3-6, Auckland, New Zealand. Los Alamitos, CA: IEEE Computer Society, 557-558.

Suhonen, J., Sutinen, E. (2003). Portfolio-based analysis in developing a tool for a web-based programming project course. *Proceedings of the IASTED International Conference on Computers and Advanced Technology in Education (CATE2003)*, June 30-July 2, Rhodes, Greece. ACTA Press.

Suhonen, J., Sutinen, E. (2004). Formative development model in action - Case of a digital learning tool. *Proceedings of the IEEE International Conference on Advanced Learning Technologies (ICALT2004)*, August 30-September 1, Joensuu, Finland. Los Alamitos, CA: IEEE Computer Society, 720-722.

Suhonen, J., Sutinen, E. (2005). FODEM: A formative method for developing digital learning environments in sparse learning communities. Accepted for publication in the 5th IEEE International Conference on Advanced Learning Technologies (ICALT2005), July 5-8, 2005, Kaohsiung, Taiwan.

Sutinen, E., Tarhio, J. (2001). Teaching to identify problems in a creative way. *Proceeding of the 31st ASEE/IEEE Frontiers in Education Conference (FIE2001)*, October 10-13, Reno, Nevada.

Sutinen, E., Torvinen, S. (2003). The CANDLE scheme for creating an on-line computer science program - experience and vision. *Informatics in Education*, **2**, 1, 93-102.

Thomas, J.C., Lee, A., Danis, C. (2002). Enhancing creative design via software tools. *Communications of the ACM*, **45**, 10, 112-115.

- Torvinen, S. (2004). *Aspects of the Evaluation and Improvement Process in an Online Programming Course - Case: The ViSCoS Program*. Licentiate Thesis, University of Joensuu, Department of Computer Science, Finland.
- Trevor, J., Hilbert, D.M., Schilit, B.N., Koh, T.K. (2001). From desktop to phonetop: A UI for web interaction on very small devices. *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology (UIST2001)*, November 11-14, Orlando, Florida, USA. New York: ACM Press, 121–130.
- Vrasidas, C. (2004). Issues of pedagogy and design in e-learning systems. *Proceedings of the ACM Symposium on Applied Computing (SAC2004)*, March 14-17, Nicosia, Cyprus. New York: ACM Press, 911-915.
- Vygotsky, L. (1978). *Mind in Society*. Cambridge, MA: Harvard University Press.
- Ward, T., Finke, R., Smith, S. (1995). *Creativity and the Mind: Discovering the Genius Within*. Cambridge, MA: Perseus Publishing.
- Watanabe, S., Nakabayashi, T., Satoh, H., Jiang, T., Oda, T. (1999). Web-based educational system: Monitoring and assisting learners. In G. Cumming (Ed.), *Advanced Research in Computers and Communications in Education*. IOS Press, Amsterdam.
- WebMacro. (2004). *WebMacro homepage*. Retrieved, May 11, 2005, from <http://www.webmacro.org>.
- Whitten, J.L., Bentley, L.D., Dittman, K.C. (2004). *Systems Analysis and Design Methods (6th Ed.)*. New York: McGraw-Hill.
- Wiley, D.A. (2001). Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy. In D.A. Wiley (Ed.), *The Instructional Use of Learning Objects*. Bloomington, Indiana: Agency for Instructional Technology and Association for Educational Communications & Technology.
- Wilson, B.G. (1997). Reflections on constructivism and instructional design. In C.R. Dills, A.A. Romiszowski (Eds.), *Instructional Development Paradigms*. Englewood Cliffs, New Jersey: Educational Technology Publications.
- Wixon, D. (1995). Qualitative research methods in design and development. *Interactions*, October 1995, 19-24.

Wixon, D., Holtzblatt, K., Knox, S. (1990). Contextual design: An emergent view of system design. *Proceedings of the Conference on Human Factors in Computing Systems (CHI'1990)*, April 1-5, Seattle, WA, USA. New York: ACM Press, 329-336.

Yin, R.K. (1994). *Case Study Research: Design and Methods (2nd Ed.)*. Thousand Oaks, CA: Sage.

Zimmerman, B.J. (2002). Becoming a self-regulated learner: An overview. *Theory Into Practice*, **41**, 2, 64-70.

Appendix A: Interview invitation letter in Study 1 (including the sketch of questions)

Invitation letter

Hi,

I am asking for your contribution to the future development of PPA [working name of LEAP] portfolio tool used in the Problem Solving course. Next week I have organized room for 15-25 minutes individual interviews about the PPA tool. The interview will consist of two parts

- 1) You will show me how you have used the PPA and tell me basically everything that comes to your mind related to it. I will ask you some questions during this time.
- 2) I will ask you a set of questions after the first part (if there is a need).

The interview will be recorded on a digital audio tape for the future analysis. The interview will begin with you showing the usage of the program. After that I will have some specific questions. The questions can be found in the attachment, so you can prepare for the interview. I will let the situation lead the interview, so there might be some extra questions or some questions may be given less emphasis.

This interview is totally voluntary and it is not a part of the Problem Solving course. If you do not want to participate, please let me know as soon as possible. It will not affect your grade at all ;)

However, I would be more than grateful to all of you who can participate. The analysis of the interviews will be a part of my PhD Thesis and it will play a crucial role in the future development of PPA.

The place for the interview is room 2D/309 (where we had the film session). The room is reserved for the interview at the following times;

- Tuesday 8.4, time 12-18
- Wednesday 9.4, time 16-18
- Thursday 10.4, time 10-12 and 14-18
- Friday 11.4, time 10-14

You can reserve a time that suits your timetable. The interview will last 25 minutes at the maximum. If you cannot make it during the above times, we can arrange a new time at the start of week 16 or at the end of week 17.

Question sketch

These questions will be dealt with during the interview. However, according to the situation some questions may be given either less or more attention.

- When and where did you use the PPA?
- What is your impression about the function and/or purpose of the PPA?
- Why did you use the PPA?
- Why did not you use the PPA?
- How did PPA help you during the course?
- Name three (3) good aspects of the program
- Name three (3) not so good aspects of the program
- What is the biggest challenge for future development of the program?
- How would you develop the program?
- Any further comments are highly appreciated

Appendix B: Questionnaire about the use of LEAP in Study 2

Background info

Name: _____

Questions

1. General use of the tool

Estimate the following arguments using the scale of 1-5 (where 1 = strongly disagree, 2 = disagree, 3 = undecided, 4 = agree, 5 = strongly agree).

	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
PPA was easy to use	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
It was effortless to start using the tool	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Given instructions were clear	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I felt that PPA was useful for me during the course	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
The structure of the tool was clear	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

2. Estimate the frequency of using the tool during the course (at the time of filling the questionnaire)

1 = daily, 2 = weekly, 3 = one or twice a month, 4 = few times during the course
5 = never

	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>
I logged on to the tool	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I added challenges	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I used the Distant Analogies activity	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I used the Question Lists activity	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
I used the BruteThink activity	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

3. Estimate the following statements

	<i>No</i>	<i>No opinion</i>	<i>Yes</i>
Distant Analogies, Question Lists and BruteThink activities were useful for defining new ideas.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Comment

It was easy for me to write about the Programming Project Course with the tool.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
---	--------------------------	--------------------------	--------------------------

Comment

I cannot find any use for PPA	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
-------------------------------	--------------------------	--------------------------	--------------------------

Comment

PPA helped me understand the phases of the programming	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--	--------------------------	--------------------------	--------------------------

Comment

PPA helped me organize my work	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--------------------------------	--------------------------	--------------------------	--------------------------

Comment

I used PPA to solve problems during the course	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
--	--------------------------	--------------------------	--------------------------

Comment

4. Opinions and feedback

Appendix C: XSL-file for Distance Analogies activity in the third experiment. Note: some content of the file is in Finnish.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!-- $Id: distant.xsl,v 1.2 2003/09/14 10:21:54 vkainu Exp $ -->

<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  xmlns:java="http://xml.apache.org/xalan/java"
  xmlns:format="xalan://WMUtils"
  xmlns:log="xalan://PmaToolLog"
  version="1.0"
  exclude-result-prefixes="format log java">

<xsl:import href="pma.xsl"/>

<xsl:template match="/solutions/tool" name="tooltemplate">

<xsl:param name="thisurl" />
<br/><br/>
<b>Kaukainen assosiaatio</b>

<p/>
Tutki
<xsl:element name="a">
<xsl:attribute name="href">pma2?m=extredir&amp;log=&amp;url=<xsl:value-of
select="tool/distanturl"/></xsl:attribute>
<xsl:attribute name="target">distant</xsl:attribute>
webbisivua
</xsl:element>
<br/><br/>
Mink&#228;laisia ajatuksia tai ideoita k.o sivu tuo mieleesi? <p/>
Mieti entry&#228; <xsl:value-of
select="object[@class='PmaProblem']/field[@name='m_strStatement']"/> sivun
antamaan virikkeen kannalta<p/>
Kirjaa kaikki saamasi ideat alla olevaan tekstikentt&#228;&#228;n ja paina
Lis&#228;&#228; painiketta. <p/>
Lis&#228;&#228; painikkeen painaminen tuottaa sinulle my&#246; uuden
kaukaisen assosiaation.
```

<p/>

<!-- [<xsl:value-of select="tool/distanturl"/>] contain word(s):

<i><xsl:value-of select="tool/terms"/></i>

 -->

<xsl:choose>

<xsl:when test="toolsettings/use_iframe='true'">

<xsl:element name="iframe">

<xsl:attribute name="src">pma2?m=extredir&log=&url=<xsl:value-of
select="tool/distanturl"/></xsl:attribute>

<xsl:attribute name="SCROLLING">yes</xsl:attribute>

<xsl:attribute name="width">100%</xsl:attribute>

<xsl:attribute name="height">400</xsl:attribute>

</xsl:element>

</xsl:when>

<xsl:otherwise>

<script language="JavaScript">

window.open("pma2?m=extredir&log=&url=<xsl:value-of
select="tool/distanturl"/>", "distant");

</script>

</xsl:otherwise>

</xsl:choose>

<!-- <form action="pma2" method="POST" target="distant">

<input type="hidden" name="m" value="extredir"/>

<input type="hidden" name="log" value=""/>

<xsl:element name="input">

<xsl:attribute name="type">hidden</xsl:attribute>

<xsl:attribute name="name">url</xsl:attribute>

<xsl:attribute name="value"><xsl:value-of
select="tool/distanturl"/></xsl:attribute>

</xsl:element>

<input type="submit" value="Avaan webbisivu omaan ikkunaan"/>

```
</form> -->
```

```
<br/>
```

```
<xsl:element name="form">
```

```
<xsl:attribute name="action"><xsl:value-of  
select="$thisurl"/>#tool</xsl:attribute>
```

```
<xsl:attribute name="method">POST</xsl:attribute>
```

```
<input type="submit" value="Uusi kaukainen assosiaatio&gt;&gt;&gt;"/>
```

```
</xsl:element>
```

```
<xsl:call-template name="solutionform">
```

```
<xsl:with-param name="toolmessage" select="tool/terms"/>
```

```
<xsl:with-param name="toolresult" select="tool/distanturl"/>
```

```
</xsl:call-template>
```

```
<hr/>
```

```
<xsl:element name="a">
```

```
<xsl:attribute
```

```
name="href">pma2?m=problems&amp;a=view&amp;process=<xsl:value-of
```

```
select="object/field[ @name='m_iProcessId']"/>&amp;phase=<xsl:value-of
```

```
select="object/field[ @name='m_iId']"/>&amp;problem=<xsl:value-of
```

```
select="object[ @class='PmaProblem']/field[ @name='m_iId']"/></xsl:attribute>
```

```
<!-- <xsl:value-of
```

```
select="object[ @class='PmaProblem']/field[ @name='m_strStatement']" />&#160;
```

```
-->
```

```
&lt;&lt;Takaisin entryyn
```

```
</xsl:element>
```

```
</xsl:template>
```

```
</xsl:stylesheet>
```

Appendix D: Summary of the data from evaluations in LEAP development

Summary of students' suggestions from interview evaluation in Study 1

Development ideas theme

Target	Students' suggestions
Tie the functionality of the tool more deeply to the course	<ul style="list-style-type: none"> - LEAP should be more course oriented - LEAP should include a course manager function where a student could see and discuss all the activities related to the course.
History page	<ul style="list-style-type: none"> - To see the previous texts when writing new text in the tool - The opportunity to add and modify phases, steps and solutions through the history page.
Graphical expression	<ul style="list-style-type: none"> - Graphics in the interface - The opportunity to express oneself graphically - The opportunity to create or upload pictures
Intelligence and interaction	<ul style="list-style-type: none"> - Function to show the other students logged into the tool. Ability to discuss to other students. - A welcome page to instruct the use of the tool - A tailored tool where LEAP could suggest students to write down to the tool
Phase page	<ul style="list-style-type: none"> - Remove the modify buttons and narrow text fields in the phase page. - The ability to change the order of the steps in the page.

Summary of students' opinions from comment evaluation in Study 1

Function of the tool

Issue	Students' opinions
General use	<ul style="list-style-type: none"> - It was also positive that the tool was usable in operating systems other than Windows, too.
Text fields	<ul style="list-style-type: none"> - Uncertainty about what should be written to the text fields - Small text fields and modify button in the phase page does not work properly. - Looking for an ability to freely write longer texts
Addition of text	<ul style="list-style-type: none"> - Difficulty on writing steps in different ideas. - Ideas are written first in paper and then added to the tool - A history page to include addition/modifications of steps

Development ideas theme

Target	Students' suggestions
New features	<ul style="list-style-type: none"> - A discussion forum inside the tool for questions and answers - The ability to see who is logged on to the tool - Tie the tasks in the course to the tool - A concept map activity
Graphics	<ul style="list-style-type: none"> - Include smiley or other graphical icons to outline the content - Highlighting of the text to mark the important content - The possibility to add mathematical symbols and Greek alphabet - Addition to figures to the tool or drawing pictures with the tool
Interface	- More focus on improving the usability of the tool. For instance, repetition of content could be found.
Instructions	- Examples of the expectations

Summary of students' comments from the questionnaire in Study 2

Comments from Part 3

Theme	Opinions
General use	<ul style="list-style-type: none"> - the idea was good, but the student felt that it was not easy to use the tool during the course. - do not want to write down the ideas - because the tool was used so late, it was confusing.
Specific functionality	<ul style="list-style-type: none"> -.graphical bugs were found. - problem solving support activities could not been found
Opinion	- LEAP was useless and the student did not spent a lot of time to use the tool

Summary of students' answers from the extra assignments in Study 2

Function of the tool

Target	Explanation
Navigation problem in most of the pages	Users could click on the link in the upper navigation bar, although they were already in the page where the link was pointing. This indicates a problem in the navigation.
Similar appearance of the phase and history pages	A student could not recognize the difference between the phase and history pages
Own information page	Information was not stored in the tool, although students tried to add text. The text that the student tried to add disappeared from the tool.
Bugs and errors	<ul style="list-style-type: none"> - Problems with browser configuration - Unsatisfactory error messages

Development ideas theme

Improvement	Concrete ideas
Incorporate the tool to the Programming Project Course	<ul style="list-style-type: none">- The selection of the topic is only done in LEAP- A matrix for collecting bugs- Every phase students need to return a document with questions about the progress of the project , i.e. class diagram in a planning phase- More text [instructions] related to the content of the project
Interface more appealing	<ul style="list-style-type: none">- Removal of all unnecessary colors and use of two to three color variations- Addition of inspiration for the eye
New features to phases and steps	<ul style="list-style-type: none">- A meter indicating the number of steps in each phase- Reminders for new and modified steps- Addition date of steps available
Public steps	<ul style="list-style-type: none">- Students were not interested in someone else's problems

Summary of students' comments from the questionnaires in Study 3

Comments from Part 3

Comments
<p>"I could not understand the meaning of those tools." "I did not know about those.." "These names do not mean anything to me." "I did not use them."</p>
<p>"Entries become forced." "If you skipped a phase or acted differently it was difficult to come up with anything." "Frustrating ..." "It felt useless." "Writing was a little bit unclear. In the summary there was entry information that could not be filled in."</p>
<p>"For me it had no advantage at all." "It might be useful if one would know what to do." "It could have been useful if I would have taken a different attitude towards the tool." "The idea is good, but LEAP is not so good. It is useful to write down the progress of the project."</p>
<p>"Instructions at the phases helped me a little bit." "I started from the third step [phase]." "Here is some point" "Explanations [instructor defined text fields] in each phase were good." "The purpose of the tool did not come clear through LEAP. The division of phases seemed silly at the beginning."</p>
<p>"My working was as unorganized as before." "I do not understand the meaning [of this feature]." "I did not use it while programming, so no. I organised my working with a similar diary tool."</p>
<p>"I trust my own head and asked for help [from instructors or peers]." "I did not realize to find out [about the problem solving features]." "Problems were solved without the help of the tool."</p>

Comments from Part 4

<p>"The comment/solution [entry and solution] part worked well and clearly, but I did not feel a need to use the other features. I did not know/remember that there were tools for problem solving, and I did not have any problems."</p>
<p>"I think that the portfolio tool was totally useless. It had more negative affect than positive, because it required too much extra work. I have my own project management methods, so the tool was useless also from that perspective."</p>
<p>"The idea of monitoring the progress of the project is good and useful. LEAP is bad, because it is unclear, and it does not adapt to the needs of the user. I used text fields to monitor my project."</p>

Appendix E: Content of a paper-based programming portfolio

Frame	Instructions	Examples of content
My Topic	Describe your topic for the project. Why did you choose that particular topic?	Description about reasons why certain topics were not chosen
Strengths	Describe your strengths in programming. What are you good at? What are your strengths as a person?	Description of the programming skills before ViSCoS program
Weaknesses	What programming skills or knowledge needs improvement?	Description about the lack of skills and aims for the course.
Goal	Describe your goal for the project	Goals of some students; to pass the course, to learn something new. Description also about the goals of ViSCoS. Estimation whether the goals can be achieved
Plan	Sketch your plan for the project	Detailed list of tasks in the project
Road Map	What were the most important phases of the project? What was difficult?	Concept map presenting the phases or features of the software; graphical presentation with a time line
Mood	Regularly update a diagram of your mood during the project	Graph representing the mood changes during the course from joyful via neutral to desperate
Team	What kind of help did you receive during the course? Which people helped you most during the course?	Feedback about the support given during the course
Source of Inspiration	Write down the ideas that occurred during the project. Where did you get ideas from? Were there any crazy ideas?	List of ideas during the project; description about the origins of the ideas: whether the idea was implemented
Mirror Hall	Are you satisfied with the result? Would you do something in a different way? How would you continue the project?	Reflection about the features of the program and tasks of the project. Issues that could be considered in future
Advertise	Draw an advertisement for your program	Graphical add with main features of the program
New Ideas	Which new ideas does your application include? Did you learn new concepts or skills?	Features of the program that satisfy the student; description about new skills and knowledge learned during the course

Dissertations at the Department of Computer Science

Rask, Raimo. Automating Estimation of Software Size during the Requirements Specification Phase – Application of Albrecht’s Function Point Analysis Within Structured Methods. Joensuun yliopiston luonnontieteellisiä julkaisuja, 28 – University of Joensuu. Publications in Sciences, 28. 128 p. Joensuu, 1992.

Ahonen, Jarmo. Modeling Physical Domains for Knowledge Based Systems. Joensuun yliopiston luonnontieteellisiä julkaisuja, 33 – University of Joensuu. Publications in Sciences, 33. 127 p. Joensuu, 1995.

Kopponen, Marja. CAI in CS. University of Joensuu, Computer Science, Dissertations 1. 97 p. Joensuu, 1997.

Forsell, Martti. Implementation of Instruction-Level and Thread-Level Parallelism in Computers. University of Joensuu, Computer Science, Dissertations 2. 121 p. Joensuu, 1997.

Juvaste, Simo. Modeling Parallel Shared Memory Computations. University of Joensuu, Computer Science, Dissertations 3. 190 p. Joensuu, 1998.

Ageenko, Eugene. Context-based Compression of Binary Images. University of Joensuu, Computer Science, Dissertations 4. 111 p. Joensuu, 2000.

Tukiainen, Markku. Developing a New Model of Spreadsheet Calculations: A Goals and Plans Approach. University of Joensuu, Computer Science, Dissertations 5. 151 p. Joensuu, 2001.

Eriksson-Bique, Stephen. An Algebraic Theory of Multidimensional Arrays. University of Joensuu, Computer Science, Dissertations 6. 278 p. Joensuu, 2002.

Kolesnikov, Alexander. Efficient Algorithms for Vectorization and Polygonal Approximation. University of Joensuu, Computer Science, Dissertations 7. 204 p. Joensuu, 2003.

Kopylov, Pavel. Processing and Compression of Raster Map Images. University of Joensuu, Computer Science, Dissertations 8. 132 p. Joensuu, 2004.

Virmajoki, Olli. Pairwise Nearest Neighbor Method Revisited. University of Joensuu, Computer Science, Dissertations 9. 164 p. Joensuu, 2004.

Suhonen, Jarkko. A Formative Development Method for Digital Learning Environments in Sparse Learning Communities, Dissertations 10. 154 p. Joensuu, 2005.