Supporting the tutor in the design and support of adaptive e-learning

(Docentondersteuning bij het ontwerpen en begeleiden van gepersonaliseerde leeromgevingen)

Supporting the tutor in the design and support of adaptive e-learning

Proefschrift

ter verkrijging van de graad van doctor
aan de Open Universiteit Nederland
op gezag van de rector magnificus
prof. dr. ir. F. Mulder
ten overstaan van een door het
College voor promoties ingestelde commissie
in het openbaar te verdedigen

op vrijdag 18 april 2008 te Heerlen
om 15.30 uur precies

door

Petrus Martinus van Rosmalen

geboren op 19 juli 1956 te Poortugaal

Promotor:
Prof. dr. E.J.R. Koper
Open Universiteit Nederland

Copromotor:
Prof. dr. P.B. Sloep
Open Universiteit Nederland

Overige leden van de beoordelingscommissie:
Prof. dr. A.A. Angehrn
INSEAD, France

Prof. dr. R.W.J.V. van Hezewijk
Open Universiteit Nederland

Prof. dr. E.O. Postma
Universiteit Maastricht

Dr. J.G. Boticario
UNED, Spain

SIKS Dissertation Series No. 2008-07
The research reported in this thesis has been carried out under the auspices of SIKS, the Dutch Research School for Information and Knowledge Systems.

The research in this thesis has been carried out at the Open Universiteit Nederland as part of the OTEC Learning Networks Programme.

Printed by: Datawyse Maastricht
Cover design: Carla Feijen

ISBN 978-90-79447-01-5

Supporting the tutor in the design and support of adaptive e-learning

Peter van Rosmalen

Synopsis

The further development and deployment of e-learning faces a number of threats. First, in order to meet the increasing demands of learners, staff have to develop and plan a wide and complex variety of learning activities that, in line with contemporary pedagogical models, adapt to the learners' individual needs. Second, the deployment of e-learning, and therewith the freedom to design the appropriate kind of activities is bound by strict economical conditions, i.e. the amount of time available to staff to support the learning process. In this thesis two models have been developed and implemented that each address a different need. The first model covers the need to support the design task of staff, the second one the need to support the staff in supervising and giving guidance to students' learning activities. More specifically, the first model alleviates the *design task* by offering a set of connected design and runtime tools that facilitate adaptive e-learning. The second model alleviates the *support task* by invoking the knowledge and skills of fellow-students. Both models have been validated in near-real-world task settings.

Table of Contents

Chapter 1

Introduction

Introduction

The further development and deployment of technology enhanced learning faces a number of challenges. Firstly, in order to meet the increasing demands of learners, staff have to develop and plan a wide and complex variety of learning activities that, in line with contemporary pedagogical models, adapt to the learners' individual needs. Secondly, staff have only limited time to support learners. Their available time resources depend on tight economical conditions. In this thesis, two models are developed and implemented that each address a specific need. The first model covers the need to support staff in their design task, the second one looks at supporting staff in supervising and giving guidance to students.

The first model seeks to alleviate the design task by offering a set of connected design and runtime tools that facilitate adaptive e-learning. The second model attempts to ease the support task by invoking the knowledge and skills of fellow-students. Both models will be developed autonomously. However, ideally they will influence each other. Helping to reduce the number and intensity of support activities to be provided by staff, can positively influence the designers' freedom to develop activities that are educationally more relevant but otherwise would have led to unacceptable workloads.

The design-task model builds on experiences with an adaptive e-learning system developed in a European project (aLFanet project, IST-2001-33288). The peer-support model, in principle, can be added to any e-learning system. In the case discussed in this work, it will be part of a prototype of a Learning Network developed in the OTEC RTD Programme Plan 2003-2008 (Koper & Sloep, 2002) and the European TENCompetence project (IST-2004-02787). The hypothesis, that the models reduce the efforts of tutors needed to develop learning designs and provide student support, will be validated in near-real-world task settings.

THE FIRST MODEL: AUTHORING ADAPTATION

Adaptation in the context of learning in general and technology enhanced learning in particular is about creating a learner experience that, over a period of time, adjusts to various conditions (e.g. personal characteristics and interests, instructional design knowledge, the learner interactions, the outcome of the actual learning processes, the available content, the similarity with peers). The intention is thus to increase success in terms of e.g. learning outcomes, time spent on a task, economical costs, user involvement, user satisfaction. Adaptation in the above senses has been on the e-learning research agenda for well over three decades with themes such as Intelligent Tutoring Systems (Wenger, 1987), Adaptive Hypermedia (now Web-based adaptive educational systems) (Brusilovsky, 2001), and Multi-agent systems (Lin, 2005; Ayala, 2003; Boticario, Gaudioso, & Hernandez, 2000). Adaptation is often based upon an Instructional Design model or guidelines, e.g. Learning Styles (Felder & Silverman, 1988), or Concept Understanding (Leshin, Pollock & Reigeluth, 1992), from which 'rules' are derived to implement the adaptation logic in an application specific representation.

Despite this research, a review of systems commonly used in universities and other institutes of higher education, e.g. WebCT, Blackboard, TopClass, Ingenium, Docent, etc. (De Croock *et al.*, 2002), reveals that these systems are not explicit about the didactical methods and models supported, nor is it possible to explicitly express them, as methods and content are intertwined. Adaptation tends to be offered in the shape of mere predefined settings requiring extensive customisation. Also, only a limited number of authors actually use adaptive designs. In practice, it appears to be difficult to use existing Instructional Design models outside the context of specialized teams. Koper (2003) summarizes the current practice in the following way: When teachers have to design or plan a lesson or course, there are several ways they can proceed. The majority of teachers employ an implicit design idea based on 'knowledge transmission'. When preparing a lesson or course they think about the content, the potential resources (texts, figures, and tools), the sequence of topics and how to assess the learners. In e-learning practice this results in a sequence of topics with dedicated content without a learning design that can be inspected or processed.

The lack of learning environments or environments with adaptive features is partly due to the lack of sufficient support for adaptive behaviour in existing learning standards, which leads to the unfortunate combination of higher initial costs and a low level of possible reuse due to proprietary models and representations (Paramythis, Loidl - Reisinger & Kepler, 2004). Starting at the beginning of the nineties, steps were taken to design and develop authoring systems for intelligent tutoring systems (Murray, 1999) and to look at generally applicable approaches. Examples of such approaches are the use of a task and domain ontology (Mizoguchi, Sinitsa & Ikeda, 1996) to support reuse of components and the use of agent architectures, which enable agents, e.g. a learner modelling agent (Paiva, 1996), to be reused in different settings. However, so far, the transfer to commonly used systems is limited.

In an attempt to remedy this, we will design and develop a framework that makes extensive use of a combination of learning standards and fits the following requirements:

- it supports active and adaptive e-learning;
- it is open to different types of pedagogical models, to alternative learning scenarios and to new components, such as agents;
- it offers a set of services that provide support to different types of users (author, student and tutor).

With this, authors should find the design of adaptive e-learning much simpler since they will have access to existing examples of adaptation and to adaptive services that can be tailored to their demands. The framework will support adaptation, both based on an initial design as well as on information inferred from user interactions. The adaptation offered will use a combination of (e-learning) standards. This will allow the creation of an open architecture composed of reusable components. The central standard will be IMS-LD (Koper & Tattersall, 2005). It enables the modelling of a variety of pedagogical designs and is novel in that it separates the design from the content. IMS-LD (IMS-LD, 2003) offers a semantic notation to describe an educational scenario in a formal way. At design time, an author or a design team can create or inspect a learning design model and use it in multiple courses. At runtime, a tutor or agent (an autonomous piece of software), can interpret a learning design and students' progress and subsequently, while a course is in progress, take action, e.g. make suggestions to learners. To complement this standard, IMS-Metadata (IMS-Metadata, 2001) describes the learning resource, which helps provide the most appropriate learning resource to a certain learner in a certain situation. IMS-LIP (IMS-LIP, 2001) is used for the representation of the user whereas IMS-QTI (IMS-QTI, 2003) is used to generate adaptive questionnaires by applying selection and ordering rules based on the metadata defined. All content is delivered in IMS-CP (IMS-CP, 2003) (cf. Van Es *et al.* (2005), for a detailed overview and discussion on the standards used).

When development of the design-task model began, few standards were actually available. Standards that could have been useful, such as IMS-AccessForAll (IMS-AccessForAll, 2004), did not yet exist. IMS-LD only existed in a conceptual form. It was first officially accepted and published in the beginning of 2003 and most systems and available experience focused on single, predominantly content related standards (e.g. IMS-Metadata) that could not represent an adaptive instructional design. Moreover, there was little connectivity between standards. As a result, it was necessary to build both the tools to support the staff (authors, tutors, administrators) and tools to support the learners in the actual learning environment, as well as to design and implement solutions that could operate with the selected set of standards in an integrated way. In this thesis project, however, our main focus lies on how to support the author in implementing adaptive e-learning. More in particular, we will hypothesize that:

- *the task of staff to design adaptive e-learning can be facilitated and simplified by following a standards-based approach*. A standards-based approach is one in which staff are given ubiquitous access to knowledge and experience embedded in standards-based learning designs, content, and adaptation services.

The hypothesis is based on two assumptions: Firstly, standards-based e-learning can facilitate the exchange of both educational design and content. In this way knowledge or experience embedded in instructional design models, good practice or content can be transferred to other situations (Sloep, 2004). Secondly, standards-based e-learning can support the exchange of components dedicated to specific types of adaptation. In this way innovations in adaptive e-learning systems can be exchanged between systems and user experiences can be exchanged among a larger audience. The hypotheses will be investigated by looking at the extent to which tutors can make use of existing design examples and adaptation services.

In Chapter 2, we will introduce the system, its components and the types of adaptations they support. We will explain the role of standards in order to accomplish a system that is adaptive, extensible and interoperable. Next, in Chapter 3, we will discuss the life cycle model of adaptation that is proposed in this project, and its evaluation. Given that such models and the connected authoring tools are rare, we will follow a formative evaluation approach. This will help us to get a better picture of which part of the model (and its tools) is effective, to what extent it is so and why. Four organizations are involved in the evaluation, two companies and two universities. The authors will have differing expertise in technology enhanced learning, varying from novice to professional. They will work independently on the design of a course over a period of several months. A design research approach will be followed. That is, the designs will be evaluated in three successive rounds. The results of each round will be fed into the next development phase of the system. In addition, in each round the functionality offered is increased. Because of the nature of the system to be developed, the evaluation data discussed will have a qualitative nature.

THE SECOND MODEL: FACILITATING SUPPORT ACTIVITIES

It is well known that the introduction of e-learning often leads to an increase in the workload for staff (Bartolic-Zlomislic & Bates, 1999; Bacsich & Ash, 2000; Koper, 2004). One of the most important reasons for this is that often an extended classroom model has been followed. That is, a teacher would lecture as usual and keep regular office hours. In addition to this, he would typically create a website to support the course and be available for email help between classes. Part of the answer to this problem is to move away from the extended classroom model and adopt a distributed learning approach (Ellis, Longmire & Wagner, 1999).

Networks for Lifelong Learning ('Learning Networks') exemplify the latter approach. A Learning Network (Koper *et al*., 2005) is a self-organized, distributed system, designed to facilitate lifelong learning in a particular knowledge domain. A Learning Network is defined in a certain domain of knowledge (e.g. an occupation) and consists of three entities:

- Users (lifelong learners): people with the intent to learn and the willingness to share their knowledge in the specified domain.
- Activity Nodes (ANs): collections of learning activities that are created and shared in order to exchange knowledge and experience or to develop competences in the domain.
- A set of competences which may be achieved by studying the ANs in the Learning Network.

But even if a Learning Network approach is used, it is still necessary to look critically at the staff time required to support students:

- Learners in a Learning Network typically do not arrive in groups, nor have the same objectives or background. The heterogeneity of the users and the lack of a readily available social structure to give mutual support make large demands on the tutors. Tutors in an online learning context (Anderson, 2004) are no longer restricted to well-defined and pre-planned tasks but have to adopt to student needs on the fly. The tutor has to make provisions for negotiation of activities to meet unique learning needs and at the same time stimulate, guide and support the learning in a way that responds to common and individual student needs.
- The availability of tutors through email makes online students expect a quick answer to the emails they have sent (Salmon, 2000); even worse, they expect personalized answers.

As a consequence, there is a need for a model to organize and support the users. One characteristic of a Learning Network makes a support model even more urgent. A Learning Network does not merely focus on formal learning. One of its objectives is also to support informal learning. In informal learning, there may not be any teacher involved at all. However, even in informal learning scenarios users will want to know how to proceed or how to understand the available Activity Nodes. Before we discuss the requirements of our model, we will present an overview of the types of support activities we are looking at.

Support activities

A brainstorm session (De Vries *et al*., 2005) with a group of stakeholders identified four groups of critical student support activities. On the one hand, they were critical in that they should enable the tutors to supply and support both simple types of learning and contemporary pedagogical approaches such as competence-based learning; on the other hand, they were critical in that each of these support actions are time consuming for the tutors. The four groups cover the following issues:

- Assessment of student contributions. The most important issues mentioned here were: to be able to give formative feedback and to be able to detect fraud, in particular plagiarism, in an efficient way.
- Answering questions of the students. This includes an efficient way to route questions to the appropriate person and to help identifying and formulating a personalized answer.
- Monitoring and assessment of study progress. An easy and effective way to monitor the progress of students ranging from general support to prevent drop-outs to the specific enabling of personalized advice.
- Community and group support. This includes (basic) functions such as select and create a group, or tasks such as ordering and archiving threads to high level overviews of the activities of a community as a whole or of individual actors.

For any of these support activities an option is to deploy greater numbers of staff. However, the limited economics of lifelong learning make this impossible but for exceptional cases. A Learning Network as we envisage it - and to which this project will contribute -, should somehow self-organize to solve these issues without extra staff involvement. We chose to first concentrate on the issue of answering questions of students because:

- Question-and-answering involves continuous interaction and consequently can be very disruptive for the tutors.
- Learning may improve when students can ask questions and subsequently receive relevant answers. Few learning environments offer students the opportunities and facilities to ask questions and receive answers (Howell, 2003).

Support activities in a Learning Network

In this project we propose a support model that automatically invokes peer-users who then provide the necessary support. The model has to fulfill the following four requirements:

- *It has to alleviate the support task for the tutor while maintaining the quality.* In the selected case, answering questions, this means that (part of) the answering is done without interference of the tutor but also that the answer has to be provided within an agreed timeframe and has to be satisfactory to the student asking the question.
- *The model has to involve a substantial part of the members of a Learning Network community and make optimal use of their knowledge.* If only a

small portion of the users is involved they may get overloaded or there will be no sharing of knowledge at all. Equally important, supporting each other on a topic just mastered can be a valuable experience (Kester *et al*., 2006).

- *The model should be able to support the selected actor in performing the task at hand.* A clear support structure is beneficial to the quality of the support task; if necessary it may even contain a quality control loop. For the current case it implies that we are looking into how learners can help each other in answering a question.
- *Finally, the model should be portable.* The model proposed should not depend on the domain, nor should it require extensive domain dependent tuning. In the same vein, the implementation of the model should be system independent. It should be relatively straightforward to add the model to any e-learning system by building on a combination of learning technology standards and technical interface standards.

Based on these requirements we will build an application that helps a student to ask a question and helps other students to answer it. In our view, the application is only successful if:

- It helps to solve a substantial number of the questions posed by students, without invoking any staff support. In our view a substantial number is about 50%, as this is the minimum percentage sufficient to justify the investment in this kind of systems.
- It selects the right students to assist. The groupings that are established by the model should outperform groups whose members have been selected at random (with workload balancing in mind only).
- It offers the students text fragments that relate to the question discussed.
- It is ‘portable’.

These conditions for success each lead to a hypothesis to be investigated empirically. The first hypothesis will be investigated by looking at the number of questions solved successfully. Whether a question is solved, is assessed by the student posing the question and by two tutors. For the second hypothesis, we will look at the difference between the two groups in responsiveness of the students and in the quality of the answers. Dependent variables measured are the number of invitations accepted, the time to answer a question, the number of questions answered, and the quality of the answers given. For our third hypothesis, we will ask our students which sources they used to answer the questions; we will also ask them to which extent they perceive the text fragments as useful. Portability we will not investigate, we will only outline the conditions for portability when discussing the application.

In Chapters 4 and 5 we will proceed to discuss the background, the design and calibration of the model. In Chapter 4 we survey the literature for relevant implementations. A cursory search for ways to answer content related questions already reveals a wide choice of solutions, ranging from groupware (Caron, 1999), helpdesks (Woudstra, Huber & Michalczuk, 2004) to virtual assistants (Gaston, 2003). Because of the nature of our setting, a Learning Network, i.e. a distributed, self-organized system, we will also look at work on agents. Multi-agent approaches have recently appeared as an alternative to distributed learning applications (Webber, Bergia, Pesty & Balacheff, 2001). In addition, because of its potential relevance to many of the items in the four groups of critical tasks identified above, we will investigate the use of Latent Semantic Analysis (Landauer, Foltz & Laham, 1998) as an example of a supportive language technology. We conclude Chapter 4 with a description of our model and a discussion of the results of a simulation with it. In Chapter 5, we focus in detail on a key aspect of our model i.e. the usage of Latent Semantic Analysis (LSA). A successful usage of language technologies such as LSA very much depends on the corpus, its preparation and the parameters applied. Our test corpus was derived from an existing Learning Network, which had been developed for a study on navigation (Janssen *et al.*, 2007). We pre-processed the corpus and investigated how to calibrate the LSA-parameters and tested the usage. In our case this means we verified how we can use LSA to identify the topic of a question (that is to which Activity Node(s) a question belongs) and to select text fragments out of the available ANs (i.e. that are of use in answering a question). In Chapter 6, we describe and discuss the results of an experiment with a prototype of the model. For the experiment, we set up a course in the Learning Network on Internet Basics, the same course that was used to calibrate the model. Students were invited from among students and staff of our organisation. They were divided at random into two groups. In the experimental group, we used our model to select the students to help answering a question. In the control group, we only made sure that the questions would be divided evenly between the students. To verify our hypotheses we looked at a combination of logging data, student ratings, staff ratings and data from a questionnaire.

Finally, in Chapter 7 'General Discussion', we look back at our findings. We started with the statement that the further development and deployment of technology enhanced learning is facing a number of challenges. We will review to what extent our work did indeed address these and discuss what insight we gained from the work on the two models proposed. Finally we put forward suggestions for further research.

Chapter 2

Towards an open framework for adaptive, agent-supported e-learning

Van Rosmalen, P., Brouns, F., Tattersall, C., Vogten, H., Van Bruggen, J., Sloep, P. and Koper, R. (2005). Towards an open framework for adaptive, agent-supported e-learning. International Journal of Continuing Engineering Education and Lifelong Learning, Vol. 15(3-6), 261–275.

Abstract

E-learners require activities and content based on their preferences and prior knowledge, not merely fully static, page-turning sequences. In this paper we present a framework that integrates and supports two approaches towards adaptation to the learner's needs – design and runtime adaptation. The framework is based on IMS Learning Design (IMS-LD). IMS-LD offers a semantic notation to describe an educational scenario in a formal way. At design time a teacher or a design team can create or inspect a learning design model and use it in multiple courses. At runtime a tutor or agent, an autonomous piece of software, can interpret a learning design and students' progress and subsequently take action while a course is in progress, e.g. make suggestions to learners. We will discuss the study that lead to the framework, and explain the role of IMS-LD and the promising role of agents in adaptive e-learning.

Introduction

Adaptation to a learner's personal interests, characteristics and goals is a key challenge in e-learning. Three decades ago, in the early 1970s, when the use of computers to capture and transfer knowledge began, the first knowledge based tutoring applications appeared in artificial intelligence, a relatively small but influential research area. In contrast to the first generation of computer assisted instruction programmes, which offered simple automated instruction, intelligent tutoring systems (Wenger, 1987) used artificial intelligence approaches to capture and deal with aspects of knowledge. Microworlds were shaped; built in various ways, but in general containing at least a detailed domain or expert model, a personal or student model and a knowledge transfer or instructional model. Persons involved in such a microworld can acquire new knowledge actively or in a guided way. They can immerse themselves in e.g. a device simulation or a programming world and practice their skills, as well as receive feedback depending on their progress. Alternatively, they can be guided through the study domain, while the best fitting chunks of information are presented (according to their knowledge level and the instructional methods applied). The intelligent tutoring systems that have been built to date are qualitatively strong, but offer only small chunks of information and knowledge from small-scale worlds and thus have limited applicability. Moreover, in general they were all built from scratch, little or no effort being paid to reusability thus making it difficult to come to a more widespread use.

In this paper we discuss an open framework developed in the aLFanet project that addresses the learners' need for activities and content based on their preferences and equally takes into account the designer's and tutor's need for efficiency. ALFanet aims to develop new methods and services for active and adaptive e-learning. Active means that the learners are involved in applying (new) knowledge or solving problems. Adaptive means that the learners are provided with a learning design that is adapted to their personal characteristics, interests and goals as well as the current context. The project's target is to deliver a tested set of components for e-learning providers that will provide significantly enhanced individual learning, through technologies with adaptive features and approaches.

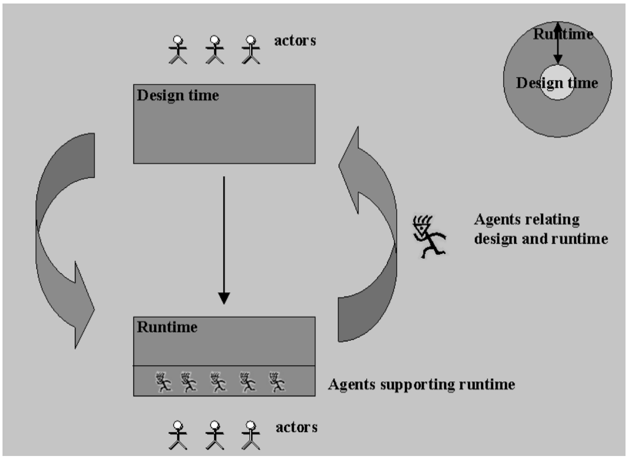


Figure 2.1 Relating design and runtime adaptation.

Within e-learning currently two approaches to adaptation are common. In the first, dominated by a strong tradition in instructional design, a team produces a detailed design of content, interaction and presentation. Within the design different options may be worked out for different learners based on *user* data, e.g. level, interest or learning style. The options for adaptation are prepared at design time and require limited, if any, interaction of tutors at runtime. The second approach is based on the assumption that author and tutor are one and the same person. The author designs the material. Next, at runtime the author, now tutor, adapts the course based on a direct interpretation of *usage* data, i.e. how well the learners succeed and what questions arise. However, both approaches tend to be (too) expensive because of high development costs or high delivery costs through extensive support.

To enable the design of the framework, a study (De Croock *et al.*, 2002) was conducted of tools, technologies and methods that take into account and can support the outlined approaches in an efficient and effective manner. The next two sections give an overview of the results of this study and the most important conclusions. Next, the aLFanet perspective and framework is outlined and the validation of the approach in a first prototype is discussed. The paper closes with conclusions including an overview of further work to be done.

E-learning platforms and the options for adaptation

The overall e-learning market in Europe is in a very early stage of development. It is highly fragmented and has a low transparency, showing a wide array of products and services offered by many different types of suppliers. Many tools arose following the promise the internet offers to organise learning, teaching and education. The internet should allow for flexibility in delivery but also in learning, in time, and place. It should also be easier to differentiate didactical models and scenarios depending on users' preferences. According to Hambrecht (2000) the supply side of the global e-learning market currently comprises approximately 5,000 participants offering every imaginable method of e-learning.

In the context of aLFanet only those environments or tools are relevant that consist of at least content delivery and tutoring facilities via internet technologies. They should allow for personalised and active learning. Following Merrill (2000) learning environments are effective if they are problem based and address the first principles of instruction for each phase of the activation-demonstration-application-integration learning cycle, i.e. learning is facilitated when:

- learners are engaged in solving real-world problems
- existing knowledge is activated as a foundation for new knowledge
- the instruction demonstrates what is to be learned rather than merely telling information about what is to be learned
- learners are required to use their new knowledge or skill to solve problems
- learners are encouraged to integrate (transfer) the new knowledge or skill into their everyday life.

It is important to note that Merrill does not see collaboration as a first principle of instruction. In Merrill's view collaboration is only one of many possible ways to implementing first principles. For the aLFanet environment we underline the importance of discussion and interactions with others. Learning is not just on a one-to-one basis with a student and information (Michaelson, 1999).

A review of systems (WebCT, Blackboard, TopClass, Ingenium Docent, etc. (De Croock *et al*., 2002; Van der Klink *et al*., 2002)) commonly used in universities and higher education showed two types of platforms. The first type takes a course as a basis, the second the organisation. Systems that take the course as a basis (e.g. WebCT, TopClass) normally do not distinguish between teacher and author (course-developer). In this way they allow the teacher much flexibility but also assume that the teacher will create material. Systems that take the organisation as a basis (e.g. Ingenium, Docent) have clearly defined and distinct roles. Content can be developed outside the system. All systems advertise themselves to be innovative and to offer new possibilities. The systems do stress the importance of content, but unfortunately for both types of systems there is hardly any information about which didactical methods and models are used and it is not possible to explicitly express them. As far as adaptation is possible it would require extensive customisation. Most of the systems do support collaborative learning tasks; however they do not allow imposing any specific scenario. They allow collaboration by merely providing the basic tools.

Currently, originating from research, a new generation of systems emerges, e.g. Edubox, that builds on an educational modelling language (EML) (Rawlings *et al*., 2002). Edubox does not prescribe a learning scenario; instead every scenario can be modelled in EML (Koper, 2001). EML is a formal language that allows a learning design to be described in such a way that automatic processing is possible. EML allows to fully describe the teaching-learning process including integration of the learners' and staff members' activities, integration of resources and services used during learning and support for both single and multiple user models of learning. Every activity or piece of content can be personalised or made available for specific users. EML is accepted as a basis for the IMS Learning Design (IMS-LD) specification (IMS-LD, 2003).

How to prepare a learning design is the main goal of any instructional design process, i.e. to construct a learning environment in order to provide learners with the conditions that support the desired learning processes. With regard to models that may sustain this process, Van Merriënboer (1997) makes a distinction between instructional systems development (ISD) models and instructional design (ID) models. ISD-models have a broad scope and typically divide the instructional design process into five phases:

- analysis
- design
- production
- implementation and/or delivery
- summative evaluation.

In such stage-models, formative evaluation is typically conducted during all phases. ISD-models provide guidelines and directions for performing the activities that form part of each of the phases. ID-models are less broad in scope and focus on the first two phases of ISD-models (i.e., analysis and design). They concentrate on the analysis of a to-be-trained skill in a process of job and task analysis and the conversion into a training strategy, or the design of a learning environment (often taking the form of some kind of blueprint) that is ready for production. If it comes to the analysis of to-be-trained skills and the design of learning environments, ID-models typically provide more specific guidelines and directions than ISD-models.

Despite these more specific guidelines it appears to be difficult to use these ID-models outside the context of specialised teams. Koper (2003) summarises the current practice in the following way. When teachers have to design or plan a lesson or course, there are several ways they can proceed. The majority of teachers employ an implicit design idea based on 'knowledge transmission'. When preparing a lesson or course they think about the content, the potential resources (texts, figures, and tools), the sequence of topics and how to assess the learners. In e-learning practice this results in a sequence of topics with dedicated content without a learning design that can be inspected or processed.

Adaptive e-learning systems and technologies

Web-based adaptive educational systems (AES) are not an entirely new or unique kind of systems. Historically, web-based AES inherit from two earlier kinds of AES: intelligent tutoring systems and adaptive hypermedia systems. Traditionally, the problems addressed in AES were investigated in the area of intelligent tutoring systems (ITS). Intelligent tutoring systems use knowledge about the domain, the student, and about teaching strategies to support flexible individualised learning and tutoring. Adaptivity was one of the goal features of any ITS. Adaptive hypermedia is a much newer research domain. Adaptive hypermedia systems apply different forms of user models to adapt the content and the links of hypermedia pages to the user. Adaptive hypermedia research also includes e.g. information retrieval. However, the most applied examples are hyperspaces of educational material. The goal here is to guide the students through the material and show them the optimal path or the optimal content. This can be achieved in several ways. The most popular use is direct guidance, i.e. they offer the best page given the student's current knowledge and learning goal. This is done through adaptive link annotation and hiding (i.e. annotating the most suitable links and disabling a link, if a page is not yet ready to be learned). Brusilovsky (2001) gives an extensive overview of what can be adapted. He describes a taxonomy with two main areas of adaptation, i.e. adaptive presentation and adaptive navigation. Adaptive presentation includes text adaptation and multimedia adaptation. Adaptive navigation or link level adaptation includes direct guidance, link hiding, link sorting and link annotation, link generation and finally map adaptation.

Looking at existing examples of AES, three important issues arise, i.e.

- the use of agents
- standards
- the types of user data available in web-based systems and how they are obtained.

AGENTS

Web-based technologies (Webber *et al*., 2001) in conjunction with multi-agent methodology form a new trend in modelling and development of learning environments. Multi-agent methodology has recently appeared as an alternative to conceive distributed learning applications. The main reasons for this are the evolution of multi-agent technology itself and the fact that multi-agent methodology deals well with applications where crucial issues, such as distance, cooperation among different entities, and integration of different components of software are found. Agents have proven to be useful in many different types of applications (Jennings *et al*., 1998) from e-mail filters to traffic control. Still, researchers do not share the same vision of what agents are. The most common way in which the term agent (Wooldridge and Jennings, 1995) is used is to denote a (usually) software-based computer system with the following properties:

- *autonomy*: agents work by their own and have some kind of control over their actions and internal state
- *social ability*: agents interact with other agents (and humans beings) via some kind of *agent-communication language*
- *reactivity*: agents perceive their environment, (which may be the physical world, a user via a graphical user interface, a collection of other agents, the internet, or all of these combined), and respond in a timely fashion to changes that occur in it
- *pro-activeness*: agents do not simply act in response to their environment; they are able to exhibit goal-directed behaviour by *taking the initiative*.

Both weaker and stronger notions of agents are used. For our case it is probably more important to notice that the application of agents in AES not necessarily stops with the taxonomy described for adaptive hypermedia systems. For example Ayala (2003) reports on agents that support the construction of knowledge. WebDL (Boticario *et al*., 2000) includes agents to guide cooperation and communication among students and with lecturers. The new Learning Technologies Development Programme at the Open University of the Netherlands (Koper and Sloep, 2002) will explore the use of autonomous agents to support tutors and others to perform their tasks more effective and efficiently e.g. by using natural language technology to answer questions (Buchholz and Daelemans, 2001) or assess essays (Van Bruggen, 2001).

STANDARDS

Starting at the beginning of the 1990s, steps were made to design and develop authoring systems for intelligent tutoring systems (Murray, 1999) and to deal with generic approaches, e.g. how to use a task and domain ontology (Mizoguchi *et al*., 1996) to support reusable components and how to use agent architectures, which enable agents (e.g. a learner modelling agent (Paiva, 1996)) to be reused in different settings. Similarly for aLFanet, to enable an open framework, it is important to build upon existing standards. Current learning technology standards only allow for simple ordering and sequencing of resources (e.g. SCORM, IMS Content Packaging, and IMS Simple Sequencing (Van Es, 2003)).

Only IMS-LD adds to this the ability to integrate learning designs (instructional designs) to enable more advanced e-learning applications, e.g. to model competency based education, portfolios, collaborative learning and personalisation. It is a semantic specification, based on a pedagogical meta-model, which describes the structure and processes in a unit of learning. It aggregates learning objects with learning objectives, prerequisites, learning activities, teaching activities and learning services in a workflow (or better learning flow), which itself is modelled according to a certain learning design. IMS-LD can be used to prepare a design and to communicate it between the different actors, teachers and agents, in the framework.

This does not necessarily imply that an actor's internal reasoning deals with IMS-LD. Suppose we have an actor that can assess an essay. The actor will only want to communicate about information on the activity that imposes the essay and the learner associated with it. The assessment itself will be based on the actor's internal knowledge. The actor could be a domain expert as well as a software agent applying text data mining algorithms.

USER DATA

Originally adaptation would take place on user data e.g. goals, tasks, background, experience, preferences combined with their progress. However, based on the characteristics of the web, user modelling is extended with data about the interaction with a system by monitoring the actual behaviour. A well-known example of this approach is the Amazon bookshop. It is based on a data mining technique called nearest neighbourhood or affinity grouping or clustering. Once customers are registered, a profile is composed of their interests and their behaviour i.e. the books ordered. The profiles are compared and clustered. The purpose of this is to give an individual advice to each customer, i.e. an advice to have a look at books that have been ordered by people with similar interests. This approach uses little knowledge about the topic involved; to a large extend it relies on the actual shopping behaviour of the customers. A wide range of possible tasks, each relying on different kinds of machine learning techniques (see Meij (2002) for an overview of techniques), exists that automatically can contribute to an e-learning environment: e.g. grouping of users for collaboration in subgroups or identifying students who progress through their learning differently from their peer group members.

ALFanet perspective and architecture

In the introduction we started to formulate the aim of aLFanet, i.e. to develop new methods and services for active and adaptive e-learning. Next, we gave an overview of tools, technologies and methods in the context of the framework. In this section we look into detail how the main requirements of the framework are fulfilled, we will discuss how we used an early prototype to check the validity of the approach and finally we will introduce the framework itself and the experiments planned.

The requirements of the framework can be summarised into three main categories, i.e. to which extent the framework:

- supports active and adaptive e-learning
- is open both with regard to the use of different types of learning models and to new components, e.g. agents
- supports the user in an efficient way.

ACTIVE AND ADAPTIVE LEARNING

The commonly used e-learning systems hardly offer any information about which didactical methods and models they use nor is it possible to explicitly express them. IMS-LD offers the possibility to explicitly define the pedagogical model. Learners can be provided with a learning design that is adapted to their personal characteristics, interests and goals as well as the current context. Obviously, this requires that the framework includes the required services to execute a design, e.g. facilities for collaborative learning tasks. A learning design approach does not imply that everything can (or should) be foreseen. During the actual learning process a lot of unforeseen events can take place or specific support can be demanded. However, an explicit learning design makes it possible to interconnect the actions proposed following the results of the automatic monitoring of the learning behaviour and the specific support actions anticipated.

OPENNESS

Open in this context we defined in two meanings. First of all the system should make it possible to express any kind of learning design and to execute it. As discussed earlier IMS-LD should be capable of expressing this diversity. A successful execution will depend – as mentioned above – on the services integrated. Secondly, it should be possible to integrate new components, services and agents. Adding a new, general service at the design level is relatively straightforward. IMS-LD functions as a high level wrapper to the service. At runtime it is mainly a technical issue, which we will discuss later in this paragraph. Adding agents to it is more complex. Agents perform a certain task, that has to be allocated and coordinated and agents may need to communicate on the context of their task. This is achieved in the following manner. First, a task can be allocated by modelling the agent as a staff role and assigning the task to the staff role. Next, IMS-LD can be used to coordinate its functioning by defining the appropriate conditions at the concerning level, i.e. activity, act or unit of learning (cf. Figure 2.2 and IMS-LD (2003)). Finally, the agent can query or parse a learning design for the required information, because a learning design can be read both at a semantic and a machine interpretable level. For example it can ask information on the current activity for a selected learner and its system log and subsequently compare the design with the actual results and report or give an advice on this. Openness at the technical level is striven for by using Java and a J2EE environment, allowing multi-platform applications, for the current implementation of the system and services. This does not preclude any other type of technology, which can be added by the inclusion and configuration of new service interfaces.

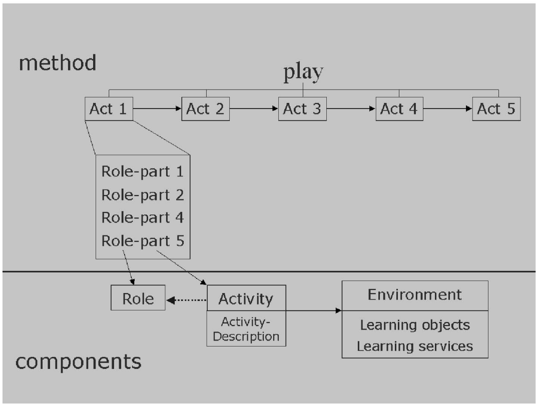


Figure 2.2 A schematic overview of a unit of learning in IMS-LD.
Source: Drawing from (Olivier, 2003).

EFFICIENCY

IMS-LD enables the use of templates for and examples of different learning scenarios without the need for course developers or teachers to design them themselves. This facilitates the enhancement and promotion of (advanced) ID-models. During the actual course a tutor will be active to support the students. The role of the tutor is specified with IMS-LD. In addition the tutor may want to intervene if unforeseen events occur. This will be easier in the case of an explicit and therewith inspectable design. Finally, because also agents can interpret the learning design they can be incorporated for many different types of tasks to support the tutor directly or indirectly by helping the learner.

Validation - a first prototype to validate the approach

The assumption underlying the use of IMS-LD is that it can be used to represent learning scenarios in a way that both tutors and agents can operate on it. To validate the idea behind this approach a minimal learning scenario (cf. Box 2.1 and Table 2.1) was designed, which involved the active participation of a tutor and two agents.

***Narrative*:** In a course in Political Sciences students get -as soon as they have answered a number of questions- the task to read and comment upon an article:

- An agent that continuously monitors the student interactions assesses the level of the student. The agent triggers himself to finalize the assessment as soon as a set of questions is answered that is sufficient to determine the level. The agent notifies a tutor of the outcome of the assessment.
- As soon as the student level is known, the tutor decides on which topic the student should focus first.
- However, the actual material to study is selected by an agent that uses an external article database to select the best fitting article. As soon as the selection is made the agent notifies the student.

Box 2.1 The narrative for the validation.

Table 2.1 The main design of the unit of learning.

Method: Agents supported education			
Play 1:	Activity 1	Role 1: student	Read and answer a set of questions
	Activity 2	Role 2: staff – agent to score assessment	Monitor the assessment Score the assessment Notify the tutor
	Activity 3	Role 3: staff tutor	Select topic area for student
	Activity 4	Role 4: staff – agent to select a resource from a paper database	Monitor Select a paper from the paper database based on (level, topic) Notify the student
	Activity 5	Role 1: student read the introduction and the advised paper	Read the paper

IMS-LD is not explicit on how agents should be integrated; in our case it was chosen to model the agents as a staff role. The agents communicated with the other actors by sending a notification when they were finished. The resulting unit of learning was successfully executed in the e-learning environment Edubox to which two dedicated agents had been added.

The framework

After the initial validation a final architecture has been worked out. The following diagram shows the aLFanet framework, the technical architecture and the way in which IMS-LD is positioned.

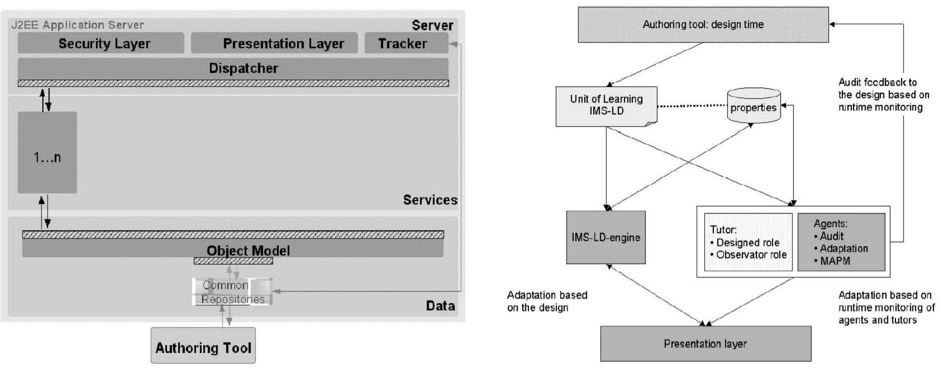


Figure 2.3 The aLFanet framework: (left) the technical architecture (Carrión et al., 2004); (right) IMS-LD as 'communicator' in between the various services.

Authors use the authoring tool to create new IMS-LD compliant courses, from scratch or based upon existing learning scenarios. After publication this results in a personalised unit of learning and a set of properties to capture the dynamic data related to the learner and the unit of learning. The IMS-LD engine processes this into a learning scenario that can be presented and executed, i.e. with the required services activated. An Interaction Module will offer the facilities for the common collaborative tasks. The Learning Adaptation Module (LAM) provides the personalised guidance to the learner. For this purpose it uses different agents applying a suitable combination of machine learning algorithms to analyse the data gathered from the user interactions. Additionally, the MAPM module will offer support depending on the instructional model applied. As a consequence the tutor should benefit from a reduction of workload since the system will take over tasks. The Audit module will supply reports including an analysis of the difference between the design and the actual learning process. This will help the author to adapt his design if required. The agents – LAM, MAPM and Audit – and the tutor can, if required for their tasks, query the design or the properties.

The architecture is a three layer composition where:

- The *Server* layer is in charge of the user front-end, managing the application security, showing user interface and tracing user interactions.
- The *Services* layer is a group of services, which provide the application functionality and main logic. It is open to include new (types of) services.
- The *Data* layer comprises the data management and storage.

The *Authoring Tool* is an independent component that allows the user (authors and editors) to create the courses.
The architecture offers an open framework in order to allow the integration of any kind of services, both in the first development and for future services. At first it will start the integration of the core modules i.e. the Interaction Module and the IMS-LD engine, followed by the Learning Adaptation Module and the Audit Module.

Figure 2.4 gives a first impression of the interface as it is currently being developed. Two parts are of interest. The first is ‘recommendations’. It contains both the suggestions automatically created by the system and those provided by the tutor for the learner. The second one is ‘roles’ identifying the role the learner has within the current context; if appropriate the learner can switch to another role.

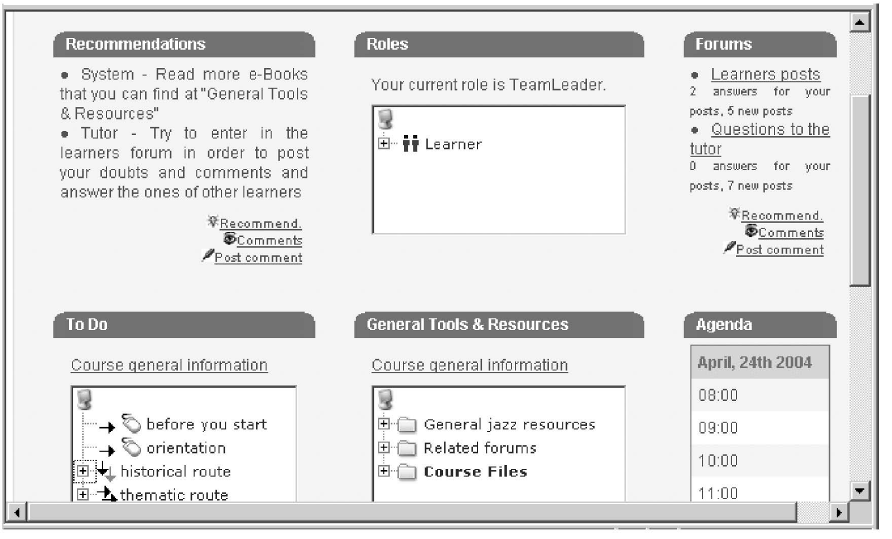


Figure 2.4 A screen shot of the interface.

The actual system will be built in three steps. The first prototype will contain the authoring tool, the IMS-LD engine and the Interaction Module and a first proof of concepts of the agents' modules (start of 2004). The second prototype will integrate the agents (mid-2004). The 'final' system will elaborate on the second prototype and address any technical issues pending. At each step a validation round is included with students from different backgrounds, company, private and university students, and in different domains, internet technology, language and waste management. The validation will mainly focus on authors, tutors, and students and include a full cycle from course development, to actual use, to a course update.

Conclusions

The objective of this paper was to outline a framework for an e-learning environment that integrates new methods and services for active and adaptive e-learning. The proposed framework is based upon IMS-LD. With IMS-LD it should be possible to describe any design in a formal way. IMS-LD will be used to communicate between the different actors, tutors and agents, in the framework. Additionally we introduced the first set of modules and agents that will populate the framework.

The first 'proof of concepts' of the approach was giving in a mock-up prototype. Obviously, the validation results of the real experiments will have to show into more detail whether the approach taken is successful. This will include questions on the usability of the approach but also the functional level, e.g.:

- what types of interventions (and when) will be appreciated by the learner
- whether the planned cooperation between humans and agents is successful and efficient
- to which extent authors can successfully use IMS-LD
- whether IMS-LD is sufficient to enable and structure the communication between the different actors.

At a later stage with the introduction of new modules and agents, it will be possible to validate the claim of openness for new components of the framework. For this it will be important to continue the analysis for which tasks agents can be of use and which techniques should be explored to enable them.

Acknowledgements

The authors thank their colleagues at SAGE, UNED, EDP, KLETT, ACE-CASE and OUNL for their contributions to the deliverables in the aLFanet project upon which this paper builds. The authors' efforts were partly funded by the European Commission in aLFanet (IST-2001-33288). For more information see http://alfanet.ia.uned.es or http://www.learningnetworks.org. Finally, the authors wish to thank the management and staff of the Schloss Dagstuhl International Conference and Research Center for Computer Science for providing a pleasant, stimulating and well-organised environment for the writing of this paper.

Chapter 3

Authoring a full life cycle model in standards-based, adaptive e-learning

Van Rosmalen, P., Vogten, H., Van Es, R., Passier, H., Poelmans, P. and Koper, R. (2006). Authoring a full life cycle model in standards-based, adaptive e-learning. Educational Technology & Society, Vol. 9(1), 72-83.

Abstract

The objective of this paper is to introduce a standards-based model for adaptive e-learning and to investigate the conditions and tools required by authors to implement this model. Adaptation in the context of e-learning is about creating a learner experience that purposely adjusts to various conditions over a period of time with the intention of increasing pre-defined success criteria. Adaptation can be based on an initial design, runtime information or, as in the aLFanet system, a combination. Adaptation requires the functionality to be able to interact with and manipulate data on the learning design, the users and the system and its contents. Therefore, adaptation is not an add-on that can just be plugged into a learning environment. Each of the conditions for adaptation have to be represented in a rigorous way. We will introduce a model based on a set of key learning technology standards that enables a structured, integrated view on designing, using and validating adaptation. For the author however, it appeared that the model is demanding both through the requirements imposed by the adaptation and the use of standards. We will discuss their experiences in applying it, analyse the steps already taken to tackle the complexity and come with additional suggestions to move forward to implementations suitable for a wider audience.

Introduction

Adaptation in the context of e-learning is about creating a learner experience that purposely adjusts to various conditions (e.g. personal characteristics and interests, instructional design knowledge, the learner interactions, the outcome of the actual learning processes, the available content, the similarity with peers) over a period of time with the intention of increasing success for some pre-defined criteria (e.g. effectiveness of e-learning: score, time, economical costs, user involvement and satisfaction). Adaptation focussed on one or more of the above mentioned conditions has been on the e-learning research agenda for well over three decades in different research topics such as Intelligent Tutoring Systems (Wenger, 1987), Adaptive Hypermedia (now Web-based adaptive educational systems) (Brusilovsky, 2001) and Multi-agent systems (Lin, 2005; Ayala, 2003; Boticario *et al*., 2000) often based upon an Instructional Design model or guidelines (e.g. Learning Styles (Felder & Silverman, 1988), and Concept Understanding (Leshin *et al*., 1992)) from which 'rules' are derived to implement the adaptation logic in an application specific representation.

Despite this research, a review of systems commonly used in universities and higher education (e.g. WebCT, Blackboard, TopClas, Ingenium, Docent, etc.) (De Croock *et al*., 2002) reveals that they are not explicit about the didactical methods and models supported, nor is it possible to explicitly express them, as methods and content are intertwined. Adaptation tends to be offered in the shape of mere predefined settings requiring extensive customisation. Also, at the design side the take-up is limited. In practice it appears to be difficult to use existing Instructional Design models outside the context of specialized teams. Koper (2003) summarizes the current practice in the following way. When teachers have to design or plan a lesson or course, there are several ways they can proceed. The majority of teachers employ an implicit design idea based on 'knowledge transmission'. When preparing a lesson or course they think about the content, the potential resources (texts, figures, and tools), the sequence of topics and how to assess the learners. In e-learning practice this results in a sequence of topics with dedicated content without a learning design that can be inspected or processed.

The lack of adaptive learning environments or environments with adaptive features is partly due to the lack of sufficient support for adaptive behaviour in existing learning standards which leads to the unfortunate combination of higher initial costs and a low level of possible reuse due to proprietary models and representations (Paramythis *et al.*, 2004). To cope with these issues, in the aLFanet project a framework has been designed that fits with the following requirements and makes extensive use of a combination of learning standards (for a detailed discussion see Van Rosmalen *et al*. (2005):

- it supports active and adaptive e-learning;
- it is open to the use of different types of learning models, alternative learning scenarios and to new components, such as agents;
- it offers a set of support services to different types of users (author, student, and tutor).

For the authors this should imply that the design of adaptive e-learning is eased by giving them access to existing examples of adaptation and adaptive services that could be tailored to their demands.

The framework supports adaptation both based on an initial design and on information inferred from user interactions depending of the components activated. The adaptation offered builds on a combination of e-learning standards. This allowed building an open architecture composed of re-usable components. The central standard is IMS-LD (Koper & Tattersall, 2005). It enables the design of a variety of pedagogical models and separates the design of the pedagogical model from the content. IMS-LD (IMS-LD, 2003) offers a semantic notation to describe an educational scenario in a formal way. At design time, a teacher or a design team can create or inspect a learning design model and use it in multiple courses. At runtime a tutor *or* agent (an autonomous piece of software), can interpret a learning design and students' progress and subsequent take action while a course is in progress, e.g. make suggestions to learners. To complement this standard, IMS-Metadata (IMS-Metadata, 2001) describes the learning resource, which facilitates to provide the most appropriate learning resource to a certain learner in a certain situation.

IMS-LIP (IMS-LIP, 2001) is used for the representation of the user and IMS-QTI (IMS-QTI, 2003) is used to generate adaptive questionnaires by applying selection and ordering rules based on the defined metadata. Everything is delivered in IMS-CP (IMS-CP, 2003) (See Van Es *et al*. (2005) for a detailed overview and discussion on the standards used in aLFanet).

At the start of the project (spring 2002) the actual use of standards was limited. Standards that could have been useful, such as IMS-AccessForAll (IMS-AccessForAll, 2004), did not yet exist. IMS-LD only virtually existed. It was first officially accepted at the start of 2003 and most systems and available experience focused on single, predominantly content related standards. Moreover, the compliance between standards was sub-optimal and only partially explored. As a result it was necessary to both build the tools to support the staff (authors, tutors, administrators), tools to support the learners in the actual leaning environment, *and* design and implement solutions to work with the selected set of standards in an integrated way. In this paper we will in particular discuss the way in which we addressed the question of how to support the author in implementing adaptive e-learning. To do so in the next section we will first introduce the aLFanet system, its components and the types of adaptation they support. Next, we will discuss the authoring process including the life cycle model of adaptation as adopted in aLFanet. This model in combination with the available authoring tools forms the backbone of the authoring process. In the third section 'Pilot Experiences' we will discuss the experiences of the authors with the tools and the approach offered. We conclude the paper with a discussion of the results, in particular the usability issues identified, and come up with suggestions for a next cycle of research and development.

Adaptation in aLFanet

System Overview

The aLFanet system (Figure 3.1) has been designed as a services-based architecture with three layers (for a detailed description see Fuentes *et al*. (2005)):

- The *Server* layer is in charge of integrating the services, the user front-end, managing the application security and tracing user interactions.
- The *Services* layer is a group of services, which provide the application functionality and main logic. It is open to include new (types of) services.
- The *Data* layer comprises the data management and storage.

In addition, and out of the three-layer architecture, aLFanet provides authoring tools i.e. an IMS-LD- and an IMS-QTI authoring tool. The IMS-LD authoring tool (www.sourceforge.net/projects/alfanetat) allows the authors to create e-learning courses based on IMS-LD including metadata (IMS-Metadata) that are optional depending on the use of the various services. The IMS-QTI authoring tool (http://rtd.softwareag.es/alfanetqtitools/) supports the addition of metadata to externally defined IMS-QTI items and the definition of selection & ordering data in order to generate dynamic adaptive questionnaires at runtime. IMS-QTI items and other types of content are created with ‘external’ tools (Figure 3.4).

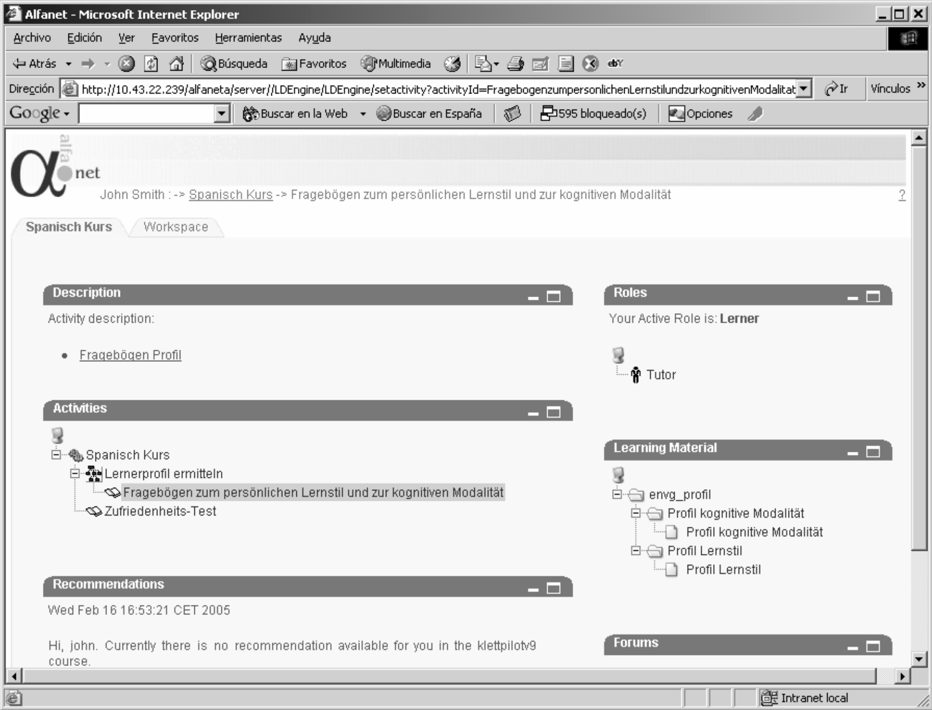


Figure 3.1 The aLFanet system: Workspace of the Spanish (German) course.

The aLFanet system includes the following adaptive and interactive components in the Services layer:

- The Presentation module provides a personalised interface (the learner can select out of a number of presentation templates) and an adaptive interface (based on the learners’ characteristics) for the different services that configure the platform. The adaptive presentation uses the information in the User Model, based on IMS-LIP and the metadata associated to the LOs to adapt the order of presentation of the LOs to the interests of the learner.
- The IMS-LD-engine, CopperCore (Vogten *et al*., 2006), provides the system with the functionality to execute UOLs (Unit of Learning) following an (adaptive) design modelled in IMS-LD. At the e-learning system level, the adaptation can be based on the UOL or the adaptation can be augmented by the other components. Information exchange between the engine and other components is supported through naming conventions. For example

data synchronization between the IMS-LD and the IMS-QTI engine is based on the use of the prefix 'sync_qtiresult_' in the properties, which is recognised and followed up at the server layer.

- The IMS-QTI-engine (http://rtd.softwareag.es/alfanetqtitools/) provides the support for the interpretation and presentation of dynamic adaptive questionnaires defined in IMS-QTI. The questionnaires are dynamically generated based on the properties in the User Model (IMS-LIP) and the metadata of the QTI-items. For example a questionnaire may adapt to the knowledge level of the student.
- The Adaptation module (Santos *et al*., 2004) provides recommendations and advice to learners while interacting with a course based on the experience derived from previous users' interactions. It combines information from the user model (IMS-LIP), the general course structure (IMS-LD), the metadata associated to the LOs (IMS-Metadata) and the results of the questionnaires (IMS-QTI). The technological base of this package is a combination of User Modelling, Machine Learning and Multi-Agent Architecture. Examples of recommendations supplied by the Adaptation module are remediation advice to study specific materials, advice to contact learners with similar interests or problems, advice to study additional learning material for learners with high interests and alike.
- The Interaction Module supports individual and collaborative users' tasks in terms of interactive services (forums, file storage area, agenda, etc). They can be based on the course definition at design time (IMS-LD).
- The Audit module generates a number of reports derived from the actual usage of the system combined with data entered in the course design in IMS-LD. Examples are: the learners who studied a specific course; the study path taken; the mean study time of an activity. The author can include additional data, e.g. 'planned study time' for an activity, in which case the system reports on the difference between planned and actual study time. The author can use the reports to close the design loop, this means to compare the anticipated use with the actual use and adapt the design if required.

AUTHORING PROCESS

Once starting the design of a course (Sloep *et al*., 2005) in aLFanet, the author has to be aware in each of the design steps from analysis to evaluation what adaptation is required, what information on the learner is of relevance and how it fits with the platform components (Figure 3.2). In the analysis phase in addition to the regular questions the author has to ask if, e.g. for the reason of the effectiveness of the learning (to achieve a higher score or reduce study time or drop out) or to achieve a higher user involvement, the design should include adaptive options. The adaptation options are constrained by the instructional design, the additional data available and the analysis of the learner interactions. The adaptation can be realised by using a specific pedagogical template or by relying on runtime information that is collected by mining the learner interactions, but in any case the data required by the responsible modules have to be represented in a rigorous way depending on the required adaptation. Also if the authors want to make use of e.g. agent-based remediation as supplied by the Adaptation module, they have to add specific metadata to the learning activities, learning objects and test items. This information is used by the Adaptation module to trace which objective or competence has been addressed and at which level of complexity and which alternatives can be used to suggest the remediation.

For authors to be able to carry out the above introduced authoring process in an effective and efficient way they:

- have to be aware of the adaptation options (*transparent*)
- have to have a clear overview of the requirements -tasks, situation and data- to be able to make a decision on including the option (*affordable*: conceptual -being able to meet the requirements- *and* economical –balancing the perceived benefits with the additional work-)
- have to have the tools to include or ‘code’ the required adaptation (*facilitate*)
- ideally, should be able to validate the results (*verifiable*).

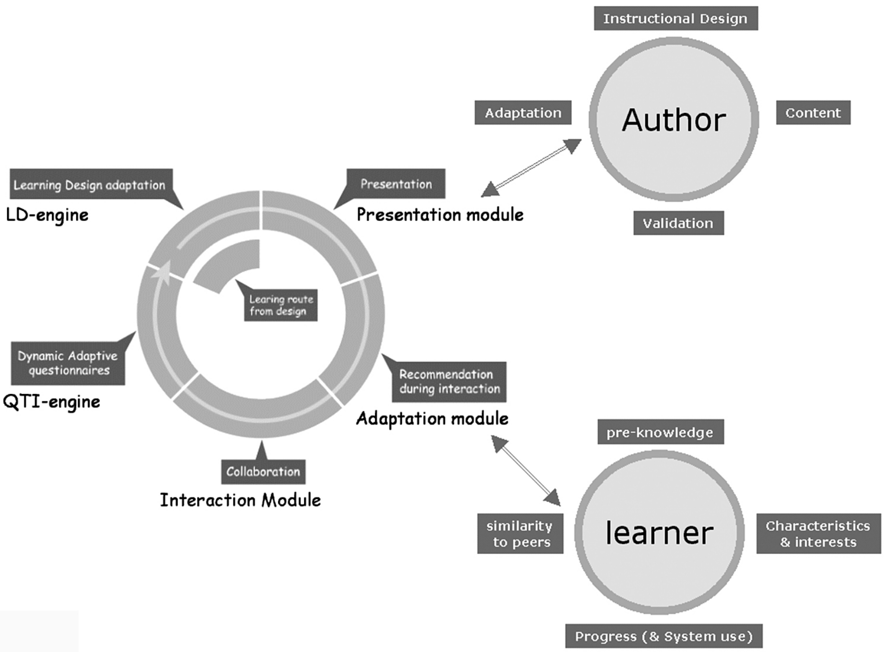


Figure 3.2 The aLFanet components and the type of adaptation they can offer related to the author’s choices and the learner’s profile.

To cope with these demands the authors received a combination of tools and documentation including a description of the aLFanet life cycle model for adaptation (*transparency* and *affordability*), a template (*transparency*), an IMS-

LD and IMS-QTI authoring tool and manuals (*facilitation*), and the access to the Audit module to support the validation (*verifiability*).

The description of the aLFanet life cycle model (Figure 3.3) includes a global description of each phase, its components and the requirements the Publication, Use and Validation have with regard to the Design phase. In the Design phase, the options for the other phases are prepared. In the Publication and administration phase, besides the normal functionality, tutors have the option to add static interventions triggered by events, e.g. based upon successful completion of a learning activity. Moreover they can define adaptive presentation rules so that e.g. the interface displays the course content following the learner's interest profile. Finally, students and tutors get assigned the roles and the rights they have in the course. The Use phase merely performs. It means the Presentation module, Adaptation module, the IMS-QTI engine and IMS-LD engine follow the design created in IMS-LD and within this context dynamically adapt and come up with recommendations based on the student interactions and their user model. Finally, the Validation phase closes the cycle. For the validation phase the system collects general data, e.g. the path through a course for a learner, and data requested by the author, e.g. whether the performance on an activity meets a pre-specified norm. The author can inspect the data and depending of their value decides if there is a need to reconsider the design.

The design contains the logic for the pre-designed adaptations and should provide the information upon which the runtime adaptation bases its reasoning. As a first step the author can select a pedagogical model template and apply it for the course at hand (note: other templates are possible, in the project however, we did offer only one) or start from scratch. The template bundles the results of research in instructional design (Felder & Silverman, 1988; Leshin *et al.*, 1992) in a UOL modelled with IMS-LD. The objective is to ease for authors the complex task of designing their courses (and, see the quote of Koper in the introduction, improve the access to best practice and the take up of results of research in instructional design). In addition the author has to define properties and add metadata depending on the adaptation required. At this stage the author has to be fully aware of which type of adaptation is required and the corresponding data and actions expected. Part of the adaptation can be fine tuned at publication time, i.e. the choice to use static interventions or to adapt the interfaces to the characteristic of the learner. Also there is the opportunity to influence the course by assigning specific roles to selected learners. Nevertheless, all underlying data and the IMS-LD has to be prepared here and now. For example an Adaptive test (Figure 3.3) in the context of the template requires the definition of metadata to the test-items and history and selection rules (IMS-QTI authoring tool) and the definition of properties following a specific format. The latter is necessary in order to be able to exchange the results of the Adaptive test between the IMS-LD and IMS-QTI engine.

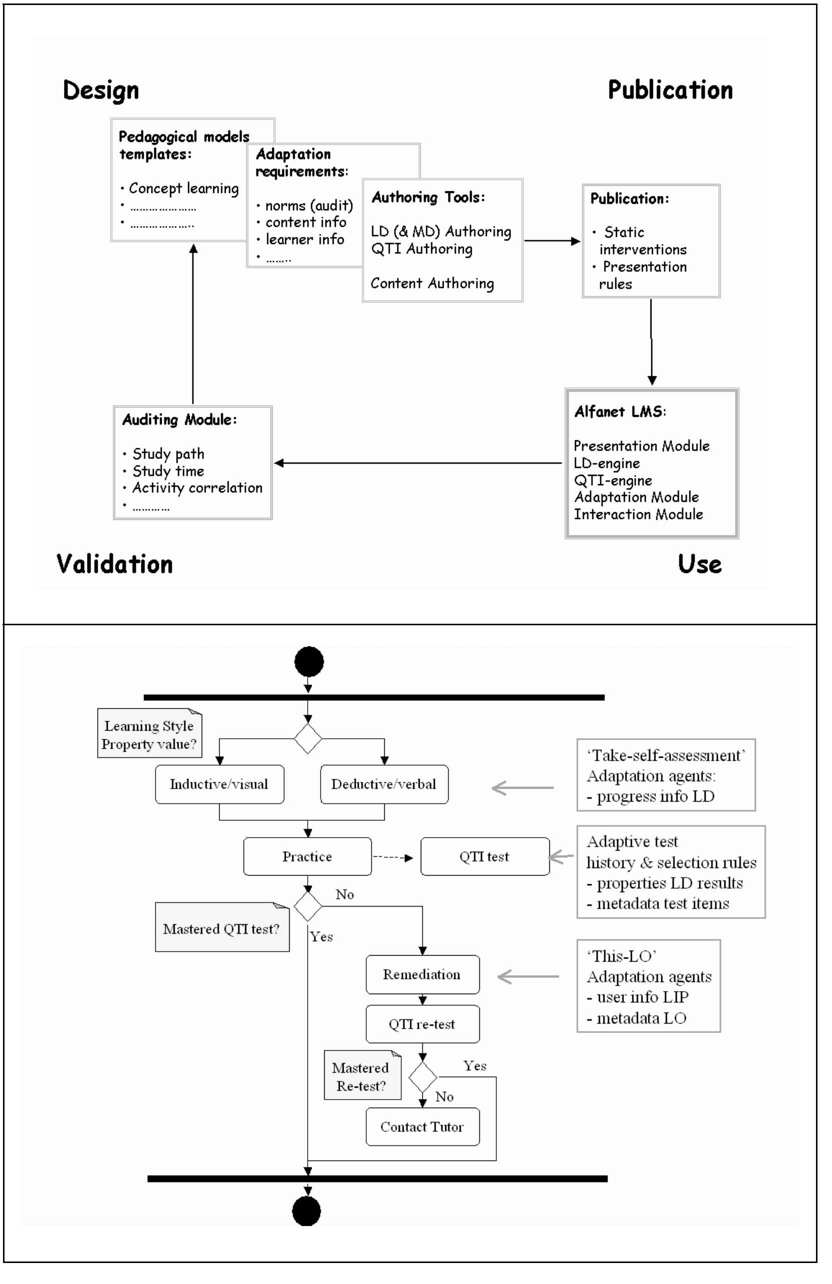


Figure 3.3 The aLFanet four step life cycle model: Design, Publication, Use and Validation and the applied pedagogical model template for 'Concept Learning'.

IMS-LD AUTHORING TOOL

The technical authoring (Figure 3.4) in aLFanet consists of the following steps:

- The creation of learning content. This is not supported in aLFanet. The authors can use different types of documents such as HTML, text, PDF, etc.
- The creation of assessments. The question items must be created in an IMS-QTI compliant tool. Once the items are created, aLFanet provides the IMS-QTI Authoring Tool. It allows the definition of dynamic questionnaires that can be adapted to each user depending on the user characteristics, course behaviour and questions' metadata that can be included while using the tool.
- The creation of the overall course structure (note the author can use the Concept Learning template) and, if required, additional adaptation scenarios based on the other services and/or modelled in IMS-LD. For instance to take advantage of the results of a questionnaire, the author has to add properties, conditions and metadata at the right place. The IMS-QTI assessment process is in charge of evaluating an exam and to generate a score value (or several score values) according to the item definitions. The IMS-QTI process has no information in order to determine whether an assessment has failed or not. The information about the required score for passing an exam is part of the design in IMS-LD. To synchronize the information of the assessment and the design it is necessary to generate scoring variables in the item definitions and in the IMS-LD design in order to determine whether the learner has passed or not.

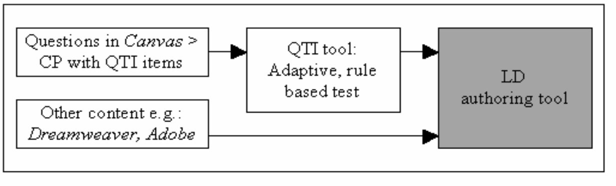


Figure 3.4 The technical authoring in aLFanet.

As a consequence the most complex and most important part of the authoring takes place in the IMS-LD Authoring Tool (Figure 3.5). The authoring tool has been created in Groove (www.groove.net), a peer-to-peer collaborative environment which is, as such, particularly suitable for teams to create and share content over the Internet. Users can add tools to a workspace from a predefined tool-set, such as forums, shared files and calendars. Additionally, it is possible to integrate custom-made tools. The core part of the Authoring Tool is the IMS-LD Editor. This sub-module allows the user to create and edit courses in IMS-LD which can be published in the aLFanet LMS. The IMS-LD Editor closely reflects the structure of the specification with only some adaptations to enhance user-friendliness. It wraps the different concepts of the learning design in sub-structures in order to be more intuitive and conceptually organized to the user. Making sure that the user always saves a valid IMS-LD-file also at intermediate stages is another characteristic of the authoring tool.

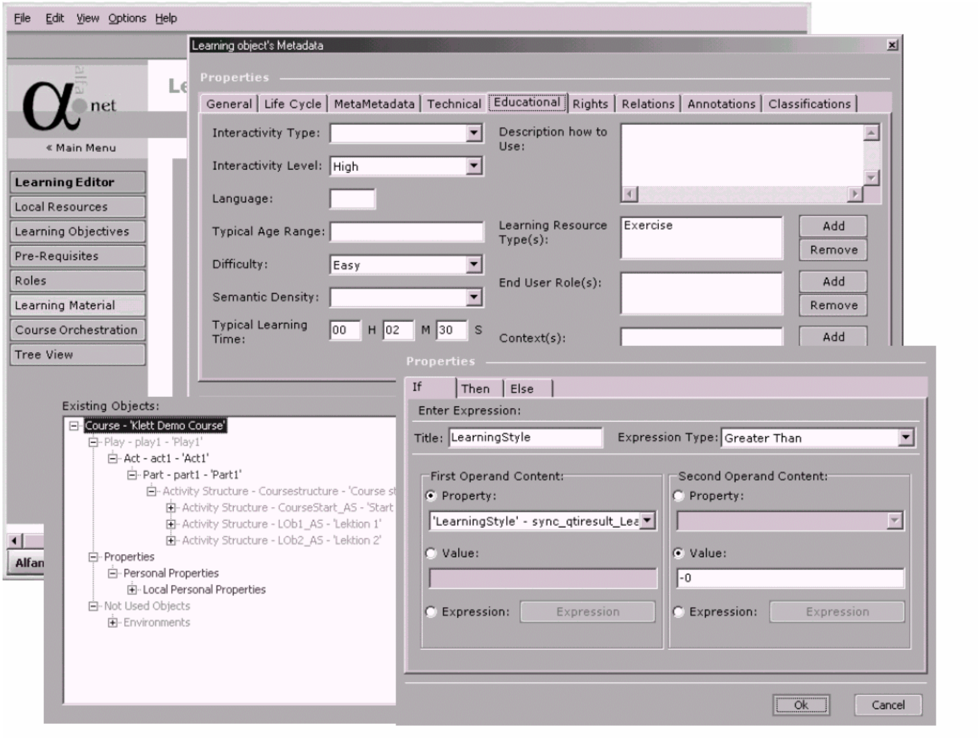


Figure 3.5 The main menu of the IMS-LD Authoring Tool and, on top the Learning Object Metadata, the Tree Representation and the Condition Editor window.

Moreover, it enables the definition of common metadata at the top-level, so that it only has to be entered once. Another useful option is that the author can get a tree overview of the course. The final result, a UOL can be saved as zip file following the IMS-CP specification (IMS-CP, 2003). The reasons for building the editor in this way, closely resembling the original specification, are twofold. First, according to the requirements the editor should be able to deliver different types of learning models and alternative learning scenarios. Following the specification should avoid any limitations resulting from the tool. Next, when the tool was built, there were, besides the official documentation, no examples of lessons modelled in IMS-LD. Examples of sets of lessons modelled in IMS-LD have only been recently explored (e.g. Van Es and Koper, 2006). Therefore for the aLFanet authoring tool, being one of the first of its kind, the only related experience available was with editing EML, the predecessor of IMS-LD. This editing was done directly in a customised, general-purpose SGML editing tool (Tattersall *et al.*, 2005). Nevertheless, although the actual IMS-LD code is hidden in the authoring tool, it still requires a solid understanding of IMS-LD and its interdependencies and, on top of this, from the specific requirements derived from the different components.

Pilot experiences

ALFanet has been built in three main cycles, in each cycle incrementally increasing its functionality. The first cycle ended with a base system operating on top of IMS-LD level A. The second version included an initial version of all components on top of IMS-LD level B. The third prototype offered an extensive set of adaptive features to choose from. Each cycle included an evaluation round with users from different backgrounds, companies, private and university students, and in different domains. More precisely two courses for university students i.e. 'How to teach through the Internet' (UNED) and 'Communication technology' (OUNL), a 'Spanish course for German Learners' intended for private students interested in learning Spanish (KLETT) and 'Environment and Electrical Distribution' for internal staff training (EDP). The evaluation did focus on the full course cycle from course design to course validation (and subsequent updates) and included authors, tutors, and students. Given the focus of the article we will only look at results of the validation by the authors (a complete description can be found in Barrera *et al*. (2005)).

EVALUATION ROUND ONE

The first evaluation round did focus on the authoring of IMS-LD level A. It contained a technical validation and a usability assessment. An IMS-LD expert did a technical pre-test with the aim to check that the functionalities provided by the authoring tool were conformant to the IMS-LD Information Model and to validate the resulting IMS-LD Code. In addition, a group of in total 8 authors were trained in IMS-LD and the use of the Authoring tool. All authors did have previous experience in creating at least one e-learning course. Only the university authors had background knowledge in the use of formal representations such as XML. The usability of the authoring tool and process was assessed with a combination of surveys and a questionnaire containing a diagnostic evaluation to identify usability problems and a subjective evaluation to get an impression on how the users felt about the software being tested. The overall feedback from the authors was that both usability and satisfaction were rated between low-medium, with the industry authors more close to low and the university authors more close to medium. Strengths and weaknesses mentioned were the following:

Table 3.1 Evaluation feedback round 1.

STRENGTH	WEAKNESS
- The lesson designer does not have to learn XML to use IMS-LD. - User-friendly interface. - It is clearly structured. - The tool generates alerts when errors occur. - Provides the option to see a diagram of the course structure.	- It assumes a great deal of knowledge of IMS-LD, and therefore the Authoring Tool requires much training. - The complexity of IMS-LD concepts. - To create a course needs a lot of time due to the excessive number of items the author is required to insert. - Lack of logic in the workflow of the course. The editor is based on a technological view of learning design rather than an educational view.

EVALUATION ROUND TWO

For the second evaluation round the initial version of the complete prototype was available. Adaptive scenarios could be added making use of IMS-LD properties and conditions and by making use of the functionality offered by one of the system components. Based on an analysis of the first round two additional support items were developed for the authors: (1) a ‘Concept Learning’ template with documentation and (2) a description of the life-cycle model adopted, the components included and its consequences for the authoring process. The template should give the authors a well structured example showing the application of an instructional design example and its translation to IMS-LD and also, equally importantly, it should give insight to the developers in the creation and use of this kind of template. The life-cycle model and its description should make clear to the author why, where and what to include in the design in order to achieve the desired system behaviour for instance adaptive testing. The authors worked at their own pace to create their courses. On request, assistance was available for minor issues by means of a forum or for more complex questions by directly contacting a specially assigned expert. At the end of this evaluation round a questionnaire was used with the following findings:

Table 3.2 Evaluation feedback round 2.

Issue	Findings
Template and life-cycle model	The template could be applied, but it was time consuming. Additionally, to use and integrate at the same time the guidelines to integrate the features of the other components e.g. to include an adaptive test resulted in a complex task.
Effectiveness	In principle the authors think that after extended experience with the tool they can work effectively with it. Nevertheless work is very time consuming due to the amount of data the author needs to process. They also complained that the work is too formalized: there is no integration of production and presentation (i.e. no What You See Is What You Get).
Efficiency	Authors said it is difficult to learn the use due to its complexity and the amount of components. On the one hand there are lots of options but on the other hand you need to be highly concentrated to be always aware of where you are and what to do.
Satisfaction	As a result of the critical aspects authors mentioned regarding effectiveness and efficiency the test persons were not satisfied working with the tool.

EVALUATION ROUND THREE

For the final prototype, only the number of adaptive features were extended. Besides some technical patches the authoring environment was the same as in the second round. The final evaluation did mainly focus on the learners, the authors did only update their course following the feedback of the second round and to include the new features of the system. In this round the feedback on the authoring process was derived only indirectly, i.e. based on the problems the authors had to get their courses running and the corresponding support they received. The findings of the evaluation in the second round were confirmed. The authoring tool could be applied - more or less - for relatively simple straight forward UOLs. However, the use of the concept template and the use of adaptive scenarios supported by the various components caused problems, i.e. without support, none of the industrial authors were capable of fully implementing the desired scenarios. The number of steps required within the IMS-LD authoring tool and between the general content tools and the IMS-QTI authoring tool were too much. Also after missing just one step it was (too) difficult to trace, identify, and solve the problem without support. It was possible for the available support staff to get the required data in interaction with the authors, so the data itself were not the problem. The amount of steps to be taken to enter the required data, the continuous awareness of which data to enter where and equally important what to ignore and finally the length of the feedback loop made it too complex to easily find omissions or mistakes. To test, the author first had to validate the UOL on IMS-LD conformance, next it had to be published and populated and finally to check the behaviour the author had to try out different scenarios – the latter a consequence of the use of adaptivity.

Discussion

The framework designed in aLFanet offers the opportunity to create a wide variety of active and adaptive e-learning scenarios. The framework has been built upon a set of leading learning technology specifications in order to assure future uptake and use of its developments. Authors can create their adaptive courses making use of pedagogical templates expressed in IMS-LD or of the adaptivity offered by the runtime services or they can create an adaptive course on their own from scratch making use of the properties and conditions in IMS-LD. At the end of the third evaluation round each of the pilot sites did include an interesting variety of - sometimes relatively complex - adaptation scenarios. The results achieved have two sides.

First of all, the results show that it is possible to support open and active learning and to create and support a set from simple to complex examples of adaptivity by combining the expressive power of IMS-LD combined with other standards supported by a combination of services. In this way the authors' work is clearly eased. They are not necessarily responsible to create the full design but they can take advantage of existing services, including agents, which can be used by taking care of in principle a simple set of assumptions. The approach taken illustrates that the complexity of the adaptation desired is not merely depending on IMS-LD (Towle & Halm, 2005). IMS-LD can be used successfully in combination with other services, including agents.

Secondly, however, despite the tools and documentation offered, only the university authors were capable of implementing the desired adaptation scenarios without support. The requirement that the design of adaptive e-learning is eased by giving the authors access to existing examples of adaptation and adaptive services (that can be tailored to their demands) has been worked out insufficiently. Though each of the authors, when asked, could deliver the appropriate data, actually entering them was only possible for the more skilled university authors. The challenge - not yet met - in aLFanet is to have the tasks to be accomplished not only clear at a general level but also to facilitate them at the micro-level concerning technical authoring. In other words, even when the tasks to achieve a selected kind of adaptation were judged to be *transparent* and *affordable*, the tools did not *facilitate* the actual technical authoring enough.

Griffiths *et al*. (2005), given the complexity of IMS-LD, distinguish two types of users, which may be involved in the actual editing of a UOL, i.e. the designers of UOLs and the adaptors or assemblers of UOLs. A similar distinction can be made between authors in aLFanet. Additionally, they distinguish two dimensions to distinguish IMS-LD tools, i.e. the distance to the specification and whether the tool is general or special purpose. The need for tools in a specific quadrant obviously depends on the type of user and the context of use e.g. the complexity and variation in courses or the access to different types of skills. The aLFanet editor has correctly been categorised in the quadrant ‘close to the spec’ and ‘general purpose’. With the exception of the content authoring, the same can be said about the rest of the aLFanet authoring process. However, the authors involved belong to both designers and adapters of UOLs with a significant difference in background and skills. In particular, for the authors with a non-IT background the usage of a complex tool in combination with the requirements to model complex adaptive scenarios appeared to be too much. The available support in the form of a template was seen as very useful but insufficient. Looking at the factors (table 3.3) that are commonly used to get an estimate of the usability of a system, it is clear that the lack of technical integration between the tools and consequently the lack of support to follow a well defined workflow negatively influences the ease of learning, the efficiency of use and the memorability. Even though the users claim that the user interface in itself is friendly and clearly structured (table 3.1), the lack of support and focus for the task at hand (e.g. to enable adaptive presentation) force the user to have knowledge about much more than they actually need for their task. It is not the information they have to enter (when asked they know) but how to get there and what to ignore that causes the problems. Additionally, the lack of direct feedback as discussed before, makes it difficult to learn and recover from errors.

Table 3.3 Factors of the user's experience that can be measured to estimate the usability of a system (see http://www.usability.gov).

Ease of learning	How fast can a user who has never seen the user interface before learn it sufficiently well to accomplish basic tasks?
Efficiency of use	Once an experienced user has learned to use the system, how fast can he or she accomplish tasks?
Memorability	If a user has used the system before, can he or she remember enough to use it effectively the next time or does the user have to start over again learning everything?
Error frequency and severity	How often do users make errors while using the system, how serious are these errors, and how do users recover from these errors?
Subjective satisfaction	How much does the user *like* using the system?

As a general rule of thumb one can argue that user-friendly editors i.e. 'distant from the specification' and 'close to the users concepts' and dedicated to a 'specific purpose' (Griffiths *et al.*, 2005) should significantly increase the success of IMS-LD and the acceptance of the aLFanet system, in whatever order. This would be much in line with the mass uptake of the Internet following the development of user-friendly html-editors. However, it is not the only way ahead. Using the same vocabulary, IMS-LD, also has clear advantages. It facilitates the discussion in and between communities and it takes away the burden to develop and learn additional metaphors. The template used and the additional additive scenarios supplied in aLFanet were received positively, however, the workflow and the tools did not use the constraints, which could be derived from these to facilitate the authors. The selection of the template and the technical authoring were perceived as two distinct not integrated processes. For example, the authors have to construct and remember the right property names (with an additional prefix 'sync_qtiresult_') to enable data synchronization between the IMS-QTI engine and the IMS-LD engine and insert them at the right place. Yet another example, to make use of the automatic remediation recommendation offered by the Adaptation module, the authors only have to add the appropriate metadata to the learning material. However, this has to be done at the right place and from a metadata selection known by the Adaptation module. In both examples it should be relatively straight forward, once the global design choices are clear, to constrain the authoring with the consequences from the choices made. To achieve this, the authoring process should be layered in two steps. In the first step the author should select and set the boundaries of the initial template and the adaptation scenarios to be included. This also emphasises better the design nature of this step. The result should be a blueprint in IMS-LD accompanied by guidelines and explanations both at an instructional and a technical level. In the next step, the authoring process should make use of the constraints imposed by the blueprint and ease the work by limiting the choices to be made and making use of the information available.

Conclusions

ALFanet is (one of) the first e-learning environment developed on a set of five e-learning standards to provide adaptation in the full life cycle of the e-learning process. Each of the phases is influenced by the requirements of the adaptation capability provided by the system. The author provides at design time all data to provide adaptation. This information is properly stored at publication time and used to adapt the course during the execution, adapt the presentation to the learners interests, present the user a more focused learning path, provide the user with adaptive assessments (use phase) and to identify critical issues of the actual usage to the course authors that can be used to update the course (validation phase). Being one of the first to explore the combination of five standards within the context of an adaptive system obviously gave rise to a lot of unexpected challenges including technical ones, i.e. standards not ‘prepared’ to work with other standards; functional ones, i.e. how to apply these standards for the functionality required; and usability ones, i.e. how to enable designers, tutors, and learners to make the most effective use of the systems while at the same time guaranteeing a system committed to a complex set of standards and a variety of adaptive learning scenarios. The first two challenges have been met the standards have been integrated and the system offers a set of adaptive features. The last one, the usability of the tools, however, is open for significant improvement. The expertise required to operate the current tools is not commonly available and is not likely to emerge on a large enough scale. The use of a template and a catalogue of adaptive scenarios were judged as useful by the authors but not translated sufficiently in the tools itself. To assure further uptake, future research and development should focus on how to clearly articulate the design choices and to translate the constraints and requirements imposed by these choices directly in the tools available to the authors to minimize complexity and to take advantage of information that can be derived automatically.

Acknowledgements

The authors thank their colleagues at SAGE, UNED, EDP, KLETT, ACE-CASE and OUNL for their contributions to the deliverables in the aLFanet project upon which this paper builds. The authors’ efforts were partly funded by the European Commission in aLFanet (IST-2001-33288). For more information see http://alfanet.ia.uned.es or http://www.learningnetworks.org. Finally, the authors wish to thank the management and staff of the Schloss Dagstuhl International Conference and Research Center for Computer Science for providing a pleasant, stimulating and well-organised environment for the writing of this article.

Chapter 4

A learner support model based on peer tutor selection

Van Rosmalen, P., Sloep, P., Kester, L., Brouns, F., De Croock, M., Pannekeet, K. and Koper, R. (2008). A learner support model based on peer tutor selection. Journal of Computer Assisted Learning, 24, 74-86.

Abstract

The introduction of e-learning often leads to an increase in the time staff spend on tutoring. To alleviate the workload of staff tutors, we developed a model for organizing and supporting learner-related interactions in e-learning systems. It makes use of the knowledge and experience of peers and builds on the assumption that (lifelong) learners, when instructed and assisted carefully, should be able to assist each other. The model operates at two levels. At level 1, prospective peer tutors are identified, based on a combination of workload and competency indicators. At level 2, the thus identified prospective peer tutors become the actual tutors; this is performed by empowering them with tools and guidelines for the task at hand. The article will situate the model in networks for lifelong learning. For one kind of interactions, answering content-related questions, we will review a set of existing approaches and emerging technologies and describe our model. Finally, we will describe and discuss the results of a simulation of a prototype of the model and discuss to what extent it matches our requirements.

Introduction

The introduction of e-learning often leads to an increase in the time staff spend on tutoring (Bartolic-Zlomislic & Bates, 1999; Bacsich & Ash, 2000; Koper, 2004). This occurs because often an extended classroom model is followed: a teacher would lecture as usual and keep regular office hours. In addition to this, he or she would typically create a website to support the course and be available for e-mail help between classes. Part of the answer to this problem is to move away from an extended classroom model and adopt a distributed learning approach (Ellis *et al*., 1999).

Networks for Lifelong Learning ('Learning Networks') exemplify such a distributed approach. A Learning Network (Koper *et al*., 2005; Koper, 2006) is a self-organized, distributed system, designed to facilitate lifelong learning in a particular knowledge domain. A Learning Network is specific to a certain domain of knowledge (e.g. an occupation) and consists of:

1. Lifelong learners (Learning Network users): people with the intent to learn and the willingness to share their knowledge in the specified domain.
2. Activity Nodes (ANs): collections of learning activities that are created and shared in order to exchange knowledge and experience, or to develop competences in the domain.
3. A set of defined learning outcomes, or 'goals' (e.g. competence levels).

But even in a Learning Network's approach, it remains necessary critically to look at the time staff requires to support students:

- Learners likely do not arrive in groups, nor have the same objectives or background. The heterogeneity of the group of learners and the lack of a readily available social structure to give mutual support make large demands on staff tutors. In an online learning context (Anderson, 2004), staff can no longer assume well-defined and pre-planned tasks but have to adapt to student needs on the fly.
- The accessibility of staff tutors by e-mail makes online learners expect a quick answer to e-mails they have sent (Salmon, 2000); even worse, they expect personalized answers.

As a consequence, also for a Learning Network a model is needed that details how to organize and support the learners. One characteristic of Learning Networks makes the need for a support model even more urgent. A Learning Network does not merely focus on formal learning but also aims to support non-formal learning. In such cases, no staff at all may be available. And yet, also here, learners will want to know, e.g. how to proceed or how to understand the available ANs.

SUPPORT ACTIVITIES

A brainstorm session (De Vries *et al.*, 2005) with a group of stakeholders identified four groups of 'critical' student support activities. They are critical in that they easily lead to staff work overload. The four groups are:

- *Assessment of student contributions*: in particular, to give formative feedback and to detect plagiarism.
- *Answering questions of students*: to route questions to the appropriate person and to formulate a personalized answer.
- *Monitoring and assessment of study progress*: ranging from drop-out prevention to providing personalized advice.
- *Community and group support*: to select and create groups, to order and archive threads, to provide overviews of the activities of a community as a whole and of the individual actors.

We chose first to concentrate on answering questions because:

- Question-and-answering involves continuous interactions and consequently can be very disruptive for staff.
- Learning may improve when students can ask questions and subsequently receive relevant answers. Few learning environments offer students the opportunities and facilities to ask questions and receive answers (Howell 2003).

SUPPORT ACTIVITIES IN A LEARNING NETWORK

In this article, we propose a support model that automatically invokes peer learners to give support. Suppose we have a Learning Network in domain *D*, e.g. psychology, with a set of ANs A_1–A_{10} (Figure 4.1). Moreover, we have a lifelong learner P (Paul) who has formulated a goal that can be achieved by studying A_1, A_2, A_3, A_6, A_7, A_9 and A_{10}. Next, we know that Paul, in view of his working experience and prior studies, has exemptions for A_5 and A_6 and has already successfully finished A_7. Finally, let's assume that Paul while studying A_1 runs into problems. He has a problem understanding the relations between a number of concepts and as a consequence he is not able to complete an assignment. He studies some additional literature and searches the web, to no avail. Paul decides to pose a question; he describes the general problem and his question.

This scenario suggests various requirements for our support model. We will discuss these now more formally and then move on to review existing approaches and emerging technologies that might help meet these requirements.

However, before doing so, we should point out that the present article is part of a series of articles. Koper *et al*. (2005) set the stage by defining the context, that of a Learning Network. De Vries *et al*. (2005) identified the needs, as just discussed. Kester *et al*. (2007) described the model from an educational, pedagogical and community perspective. Van Rosmalen *et al*. (2006) focused on the usage of Latent Semantic Analysis (LSA), the (required) calibration approach, its result, and a simulation. In these articles, little attention has been paid to what technologies exist to implement the question-answering model we seek to develop. The current article tries further to elaborate the picture by articulating requirements, reviewing existing approaches and – underpinned by these findings – detailing a model.

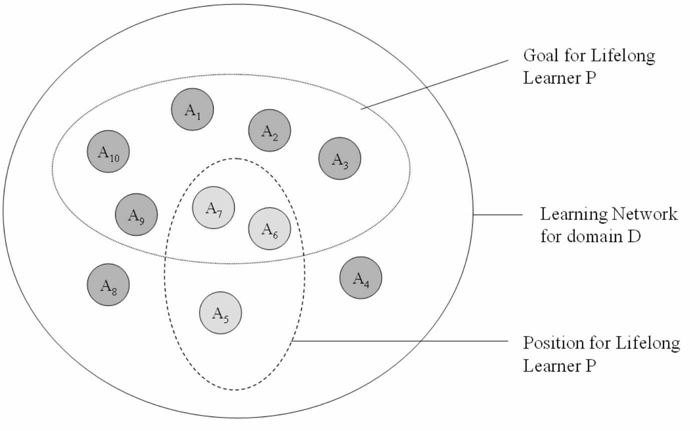


Figure 4.1 A Learning Network for domain D.

Requirements

We distinguish four types of requirements: quality, involvement, empowerment, and portability.

- *The model has to* ***alleviate*** *the support task for the staff tutor while maintaining quality.* It means that (part of) the answering is performed without staff intervening and that the answer has to meet a minimum quality level. Thus, the model should increase the number of students a staff tutor can support. Wiley (2004) captures this challenge in one concept: the teacher bandwidth, the number of students a teacher can serve in distance education.
- *The model has to* ***involve*** a substantial fraction of the members of a Learning Network community and make optimal use of their knowledge. A Learning Network as a self-organized, distributed system depends for its functioning on the learners' willingness and time to share their knowledge. If only a small portion of the learners actually contribute answers they themselves now may become overloaded or there will be little sharing of knowledge. Equally important, supporting each other on a topic just mastered can be a valuable experience (for a detailed discussion on the underlying theoretical aspects of our model on learning in communities and peer tutoring see: Kester *et al*. (2007)). Providing peer support may strengthen the social relations and can help achieve better learning outcomes (Fantuzzo *et al*., 1989). In particular, lifelong learners can, given their experience, easily change roles from student to coach and move between learning and working (Anderson, 2004). Obviously, we have to acknowledge the time constraints of lifelong learners. Therefore, the model should be able to involve competent peers while at the same time evenly spreading the workload.
- *The model should be able to* ***support*** *the selected actor in performing the task at hand.* A clear support structure is beneficial to the quality of the support task, if necessary it may even contain a quality control loop. The structure should also allow the learners to concentrate on the content of the task; this benefits their learning outcomes. For the current case, it implies that we are looking into how learners can help each other answering a question.
- *Finally, the model should be* ***portable***. The model proposed should not require extensive domain dependent tuning, preferably none at all. In the same vein, the implementation of the model should not be system dependent. It should be relatively straightforward to add the model to any virtual learning environment by building on a combination of learning technology standards and technical interface standards.

Existing solutions

A wide choice of solutions exists for the task selected, answering content-related questions, ranging from groupware, helpdesks to virtual assistants. We will discuss each of them paying special attention to an example of language technology, i.e. LSA. Question-answering depends on an understanding of natural language. The use of language technology may enable us partially to automate question-answering. LSA has been used already in a variety of educational settings, such as essay grading and question-answering.

Caron (1999) gives a broad overview of groupware systems. They range from general purpose, pre-web technology Usenet discussion groups; via dedicated question-answer systems intended to solve problems building on a combination of posting and brokering; to still popular recommender systems such as Slashdot (http://www.slashdot.org/). Two of his findings are of interest here. Often there is a small group of users who 'altruistically' reply to contributions. Thus, on the whole, only a small number of participants is responsible for a large percentage of the contributions. This makes the use of groupware rather unpredictable and hence unreliable, unless there is a facilitator or a high number of users. Similar conclusions have been drawn in educational settings (Guzdial, 1997; Anderson, 2004). Both Guzdial and Anderson underline that if participation is desired, there should be clear incentives and guidelines. This seems true in particular for lifelong learners. They participate in many activities that compete for their time, and thus need convincing arguments to join in yet another activity.

Helpdesks (Woudstra *et al*., 2004) are another common solution to deal with questions. A helpdesk is often used as a first-line aid, or as a means to forward a question to an appropriate person in the organization. Ideally, a helpdesk learns from previously asked questions and it accumulates relevant data on its customers. A helpdesk therefore requires staff tutors but only if the type of question requires their expertise or their formal involvement. A successful helpdesk should quickly pay back its investment. Unfortunately, in our case a substantial number of the questions learners will pose is directly related to the content of the activities they are involved in. Given the broad coverage of topics a Learning Network is expected to deal with, it will be difficult to staff a helpdesk adequately and yet avoid running into the teacher bandwidth problem.

Another way of helping customers with their questions, separately or in combination with helpdesks, is to create a Frequently Asked Questions (FAQ) or online virtual assistant. There is a fast growing number of virtual assistants in all areas of business (see e.g. http://mysiteagent.com/, http://www.nominotechnologies.com/). They apply a combination of agent and language technologies and operate not only via the web, but also via instant messaging or cell phones. At the EDUCAUSE2003 Conference (Gaston 2003), an example of such an assistant was presented that allowed students to ask questions such as ‘when do classes start’. Though useful if created carefully, they are insufficient if they operate on their own because it will be too difficult and time consuming for them to offer sufficient coverage. Other more general examples of agents are I-Help (Vassileva *et al.*, 2001), Yenta (Foner, 1997) and Expertfinder (Vivacqua & Lieberman, 2000). They do not rely on a set of pre-designed question-answer pairs but, based on a set of characteristics, try to find a suitable person or, as in the case of I-Help, a suitable person or material.

I-Help is based on a multi-agent architecture, consisting of personal agents (PAs) (of human users) and application agents (of software applications). Each agent manages specific resources of the entity it represents, including, for example, knowledge resources or instructional materials. If a user requests help, the agents communicate with each other and with matchmaker agents (MMAs) to identify appropriate help resources. If an electronic resource is found (represented by application agents), the PA ‘borrows’ the resource and presents it to the user in a browser. However, if a person is identified, the agents negotiate the price for help, as human help involves inherent costs (time and effort) for the helper. Help is arranged (negotiated) entirely by the PAs, thus freeing the users from the need to bargain. In this way, the PAs trade the help of their users on a virtual help market.

Yenta, a multi-agent matchmaker system, has been designed to find people with similar interests and introduce them to each other. Yenta seeks to assist people in finding people with relevant expertise. It does so by involving the majority of ‘lurking’ people instead of turning to those people who are already active. Yenta assumes that two users have a similar interest if both possess similar documents (e-mails, newsgroup articles, files).

Expertfinder is an agent that classifies novice and expert knowledge by analysing documents created while working in the domain of Java programming. The user models are automatically generated and allow for matching of a novice’s query to an appropriate expert. The system tries to distribute the workload evenly when more experts are available. It also does not prioritize the best expert but someone whose knowledge level is close to the questioner’s. This way, it is more likely to bring together people who share a similar mental model of the problem discussed. The number of success cases reported, i.e. experts able to find an answer, was 85%. Interestingly, in 50% of the cases, the expert was able to give an answer only after looking it up.

LSA

Question-answering depends on understanding natural language. Therefore, it is worthwhile to consider the use of language technologies. They may help us automate question-answering, if only in part. An example of particular interest because of its widespread use in educational settings is LSA (Landauer *et al*., 1998; Van Bruggen *et al*., 2004; for a brief technical introduction to LSA see http://research.nitle.org/lsi/lsa_ definition.htm). LSA has its roots in research on document retrieval. LSA connects related words in a number of steps (e.g. in documents on Computer Science the words human, computer and interface are related). In this way, although the actual keywords in documents may differ, LSA may show them to be associated through these kinds of semantic similarities. By relying on measures of semantic similarities between documents, LSA is able to improve retrieval beyond keyword matching (Dumais, 2003). Among other things, LSA has been used extensively and successfully for automated essay grading (Foltz *et al*., 1999), in intelligent tutoring environments (Graesser *et al*., 2000) and to help answer questions. HURAA (Person *et al*., 2001) and FAQO (Caron, 2000) are examples of systems in which the user can ask questions formulated in natural language.

HURAA is a web-based information delivery and retrieval system that guides the user through six distinct learning trajectories. At any point during a learning session, the user may ask a question. The question is mapped into an LSA text space built of a variety of documents plus a corpus of question-answer pairs. LSA is used to locate the five best text segments for the user. FAQO is a (prototype) system that allows the users to query questions in natural language in order to find relevant documents to solve their problems for specific technical problems. The objective of the system is to support the staff involved in answering these questions. The system constructs an LSA text space from e-mail archives and other existing documents in the problem area concerned. LSA is then used for query matching.

SUMMARIZING THE VARIOUS APPROACHES

All examples discussed deal with answering questions. Looking at the way in which the answers are given, one can distinguish three types of approaches. The first relies on stored answers (helpdesks, FAQ, virtual assistants); helpdesks are included because of the limited capability of their staff to answer not-anticipated questions. The second approach relies on finding the right person to answer. The person can be loosely coupled as with groupware. Here the poser of the question just has to wait until someone volunteers. Alternatively, a person is carefully identified as in the agent-based systems (I-Help, Yenta, Expertfinder). In the third approach, (a contribution to) an answer is automatically identified with the help of LSA from a corpus of documents built from the topic under discussion.

The first approach does not fit our requirements. It relies on a labour-intensive preparation of possible answers for each domain and in many cases it will still need staff to assist. The second approach, however, seems to fit the bill, even more so if we combine it with the third approach. LSA can be used to assist in identifying relevant documents to answer questions formulated in natural language. The resulting documents can then be used to assist the persons identified in giving an answer. This combination of carefully selected persons *and* documents we will therefore adopt to develop our own support model.

The model: alleviating the tutor load

Broadly speaking, the model describes how to select and support a group of lifelong learners that will help to answer a question of one of their peers. The staff tutor will only interfere if triggered, for example because an answer is not in time or does not meet a pre-specified minimum quality rating. Staff may also interfere of their own volition, for instance to assure the quality over time by sampling answers regularly. The model addresses both the need of learners to receive personalized, individual feedback and the need of staff tutors to keep their workload within bounds. It makes use of the knowledge and experience of peer learners. It builds on the assumption that lifelong learners, when instructed and assisted carefully, should be capable to assist each other, e.g. in carrying out joint assignments, giving peer-assessments or answering question of each other. The model distinguishes four types of participants (Figure 4.2):

- a learner (tutee) who asks for support;
- a learner who acts as peer tutor and provides support;
- for every learner, a PA that assists in maintaining his or her data;
- an MMA to organize and control the interactions between the actors (learners and their PAs). Both the PAs and the MMA will consist of a set of specialized agents which deal with specific tasks, e.g. an agent that proposes pieces of text suited to help answering the question.

The model builds on the assumption that learners have been registered and that their ‘position’, the combination of successfully completed ANs and the ANs they have exemptions for, is known. The model assumes that learners know the contents of an AN if their position includes the AN in question.

The approach followed contrasts with other approaches in which people are appointed beforehand (tutors, outside experts or peers from the same class). In Learning Networks, in general, there are no classes and people will have a variety of backgrounds and study plans. Hence, the group is created 'on demand' and expected to exist only for as long as is required to support the request. Clearly, although this 'ad hoc' community itself will be transient, the relations that have been forged during its existence may last. Indeed, it is hoped that they will thus be establishing a higher degree of self-organization of the Learning Network.

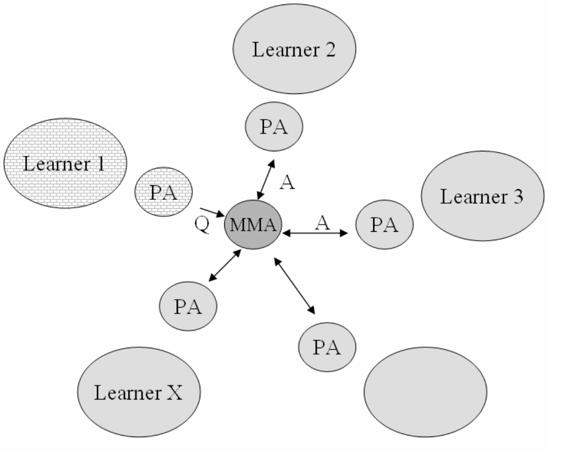


Figure 4.2 Schematic drawing of asking a question: (1) Learner 1 poses a question. (2) The Match Maker Agent selects and negotiates with the Personal agents. (3) Learner 2 and Learner 3 supply an answer.

The model recognizes five main steps. In the first three steps, the working context is defined. The steps are *creating a request*, *defining its context*, *identifying suitable candidate peer tutors*. In the last two steps, the actual request for support is addressed (*creating the answers*) and the question poser (tutee) passes judgment on the answer and the contributors (*the tutee receives the answer*). The assistance of the staff tutor is required only if a question is not successfully resolved or if a learner (repeatedly) is refusing to participate or is rated poorly.

CREATING A REQUEST

The learner who intends to ask a question will receive a form with guidelines and a request for additional information, e.g. on the urgency of the question. We have decided to restrict the model to content-related questions. The learner receives instructions that technical questions (e.g. 'I cannot access the content. What to do') or procedural questions ('when and where can I do my examination on . . .') are considered to be out of scope and should be asked elsewhere.

DEFINING THE CONTEXT OF THE REQUEST

Usually, the question asked will be related to the AN the learner is studying at that moment. This need not be the case, though, the learner might study more ANs at the same time and there could be other ANs that relate to the question. Therefore, this step determines the ANs containing information that is relevant to the question. In a way similar to Yenta, we look at the similarity of documents. We use LSA to calculate the similarity between the question and the documents of the ANs. The ANs that best fit (a combination of the number of documents that have a high similarity and the level of similarity) the question are considered relevant.

IDENTIFYING SUITABLE CANDIDATE PEER TUTORS

The next step is to find and select, based on the context defined, suitable peer tutors and to decide on the optimal number of peer tutors. The community that thus arises should be large enough to guarantee that an answer becomes readily available but small enough to minimize the chance of duplication of efforts. Obviously, what the optimal size is cannot be decided *a priori*; it is an empirical question. A size of 1 could in principle suffice, but this one person may not be available or may give an inadequate answer; the entire Learning Network would maximize the chance of a quick answer, but such a strategy is bound to lead to duplication of efforts. Also, too large a community would dramatically increase the number of lurkers. About five seems to be adequate (Kester *et al*., 2007). The system now attempts to form such an ad hoc and transient community by inviting learners who, according to four different criteria, are most suited to answer the question (see Table 4.1 for the selection formula). The suitability ranking is a weighted sum of *tutor competency*, *content competency availability*, and *eligibility*:

- The *tutor competency* (TL) is the ability of a peer learner to act as a tutor. The tutor competency is derived from a combination of data logging, i.e. from the frequency and size of the contributions, and ratings on answers given previously.
- The *content competency* (CL) indicates if a learner has successfully finished the ANs related to the question; more precisely, it is the weighted sum of the status of all relevant ANs. A more sensitive measure could be obtained by weighting the ANs according to the time elapsed since their completion: the more recent, the larger the weight.
- *Availability* (AL) is based on the actual availability as derived from the personal calendar of the learners and their past workload. This measure is time-dependent: recent workloads should affect availability more than ancient workloads.
- Finally, *eligibility* (EL) measures the similarity of the learners. It looks at which other ANs, outside the question-specific ANs, the potential peer tutor

and the tutee have in common. There are two reasons to use this measure. Some learners will have more expertise than others. The total tutoring load is therefore likely to increase rapidly with increasing expertise. However, an unequal spread of the tutoring load is undesirable. Learners should only spend limited time and effort on tutoring. By considering similarly advanced learners only, one avoids piling up questions on the advanced students. There is an additional, pedagogical twist to this argument. If tutoring is an educationally valuable experience *per se* – and not just a matter of community service – then learners should act as tutors for learners with a similar not too distant expertise level and background to achieve higher learning outcomes themselves. The eligibility of a learner guarantees that 'near-experts' (near in the meaning of having expertise close to the user asking the question) are prioritized.

SUPPORTING AND CREATING THE ANSWERS

Based on the suitability ranking above, a number of learners are invited to join a wiki and assist in answering the question. The invitation includes the question, guidelines, and a small set of documents (or paragraphs thereof) that have been identified as relevant to drafting an answer. The guidelines *and* the documents together form a support structure for the invited peer tutors. The documents are derived with the help of LSA, in a similar way as explained before. The objective is to help the peer tutors to get a quick overview of documents relevant to the question.

THE TUTEE RECEIVES THE ANSWER

After some time, the peer tutoring process ends and a response becomes available. Ideally, the process ends because the tutee is satisfied with the answer. However, if this is not the case, it may also end because a predefined period of time has elapsed or because the learners agree to end it. Whatever the reason, the tutee should rate the work of the peer tutors by rating their collective answer. If necessary, these data are used, to alert a staff tutor that there is an unresolved question or (in combination with other logging data) that some learners do not perform as peer tutors as required.

A first simulation

To test our model, we decided to build a prototype. We used a server-based architecture since, in this way, most of the required components (Figure 4.3) were readily available. To assure that the prototype is viable, we calibrated the LSA-parameters, and simulated and tested two key aspects. First, we checked how well we can use LSA to identify the topic of a question (i.e. to which AN(s)

Table 4.1 The main formula to select peer tutors and the parameter setting applied.

Explanation	Formula	Parameter setting
Tutor suitability of learner L: Ts_L. A number between 0 (not suitable at all) and 1 (very suitable). Parameters WT, WE, WA, WC to adjust the relative importance of the four factors. **Notes**: (1) to assure a minimum level of knowledge, the four factors are *only* calculated if the Content competency > 0. (2) to assure presence, if available time in the question period is zero the learner in question is removed from the list.	Ts_L = ((WT x T_L) + (WE x E_L) + (WA x A_L) + (WC x C_L)) / (WT + WE + WA + WC)	WT = 0 WE = 0.5 WA = 0.5 WC = 1
Tutor competency: T_L. A number between 0 and 1. Parameters Tw_1 and Tw_2 to adjust the relative importance of Te (on average how active the learner behaved in previous questions) and Tr (on average how previous answer were rated).	T_L = ((Tw_1 x Te) + (Tw_2 x Tr)) / (Tw_1 + Tw_2)	Not available, since (WT = 0)
Eligibility: E_L. A number between 0 and 1. E_L is taken relative to Lq, the learner who asked the question. It is calculated over all ANs that do not relate to the question.	E_L = (Sum $_{(i=i,2,..,N \text{ \& all i \| ANi is not question related})}$ (score(ANi_L) = score(ANi_{Lq})))/(N - # question related ANi's)	The score of AN can be 0 (not started), 0.3 (started), 1 (assessment completed successfully).
Availability: A_L. A number between 0 and 1. Parameters M (max_extra_workload) and Tp (timeperiod over which the workload is calculated). The availability depends on the relative past workload. It compares the number of times a learner is involved in answering a question relative to the other learners in a given time period.	A_L = one of {0,.0.25, 0.5, 0.75, 1}. The value is 0.5 if L has contributed on average; 0.25 if L has contibuted above average but no more than M above average; 0 if L has contibuted more than M above average etc…	M = 1
Content competency: C_L. A number between 0 and 1. Parameter D to adjust the number of documents to calculate correlations for. Dt is the number of text fragments offered. W_{ANi} is based on the correlation between the question and the documents in ANi. The correlation is calculated with LSA. C_{ANi} is the Content competency for ANi. **Note**: The value of C_{ANi} takes into account the score, the time expired since completion and the study time of the ANi.	C_L = (W_{AN1} x C_{AN1}) + (W_{AN2} x C_{AN2}) + …+ (W_{ANn} x C_{ANn}) / (W_{AN1} + W_{AN2} + …+ W_{ANn})	D = 3 Dt = 3 Note: C_{ANi} only based on the score of ANi

a question belongs) and to select text fragments useful for answering the question (Van Rosmalen *et al*., 2006). Second, we checked if the peer selection formula met our expectations.

THE PROTOTYPE

The prototype (Figure 4.3) consists of five modules. The learners will only notice a Learning Network, its ANs and a question interface; additionally, for each question, there is a wiki that includes the question and three documents selected from the Learning Network's ANs. All are implemented in Moodle (http:// www.moodle.org/). The wiki is populated with both the tutee and the learners who accepted the invitation to help (the peer tutors). For the designer and for the runtime system we have three additional modules: a General Text Parser (GTP; Giles *et al*., 2001), a GTP calibrator [GTP Usability Prototype (GUP); De Jong *et al.*, 2006] and a tutor locator [ASA Tutor Locator (ATL); Brouwers *et al.*, 2006]. We use GTP, an LSA implementation, to map the questions on the documents in the Learning Network. The GTP module returns correlations between the question and documents. The correlations are used to determine the AN to which a question fits best and to select relevant text documents. The GUP module supports the calibration of the LSA parameters. Finally, the ATL module finds and invites the peer tutors.

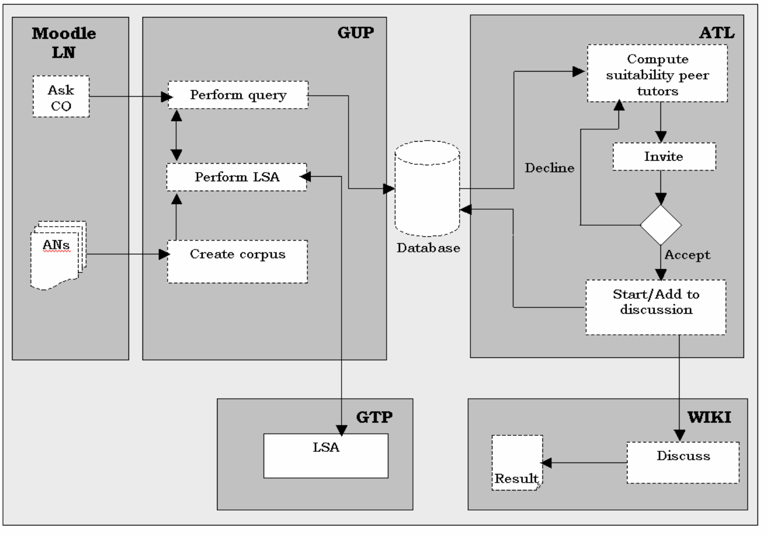


Figure 4.3 The main modules of the prototype: Moodle with a Learning Network (LN), the Ask Content Question Module (Ask CQ) and Activity Nodes; GUP (GTP Usability Prototype); GTP (General Text Parser); ATL (ASA Tutor Locator); and a WIKI.

AN IDENTIFICATION AND TEXT FRAGMENT SELECTION

For the simulation, we used an existing Learning Network, the domain of which is basic Internet skills (Janssen *et al.*, 2007). It contained 11 ANs, each of which introduced a different aspect of the Internet and consisted of an introduction, exercises, references to external web pages for further study and an assessment. The Learning Network matches our two initial requirements, i.e. (1) the text corpus could be accessed (a combination of Moodle and external web pages); and (2) the users' progress could be tracked (by the data available from the AN assessments). We formulated a set of 16 test questions, each related to exactly one AN. For each question, the prototype proposed three text fragments as well as determined the source AN.

Table 4.2 Position of learners L_1-L_5 for the selected Activity Nodes (ANs).

	L_1	L_2	L_3	L_4	L_5
Score AN1	1	1	0.3	0	0
Score AN2	0.3	1	1	0	0
Score AN3	0	0.3	1	1	0.3
Score AN9	0	0	0	1	1
Score AN10	0	0	0	0	0
Score AN11	0.3	0	0	0.3	1
A_L	0.5	0.5	0.5	0.5	0.5

PEER TUTOR SELECTION

To test the peer tutor selection process, we created five learners (see Table 4.2) and we assigned a set of test values to the parameter of the peer selection formula (cf. column 3, Table 4.1). Content competency as the most important element received weight 1. To simplify the preparation of the learners' data, we set the weight of the Tutor Competency to 0. Furthermore, given that we only have five learners, we let them be always available, we assigned only one peer tutor per question, and we gave M, the bandwidth, value 1. Finally, we had learner 1 'ask' two of the 16 questions mentioned above. Next we assumed that a question is resolved by the learner with the highest rank and we asked the same questions once more to show the effect of workload. The results of this exercise on the behaviour of the model are given in Table 4.3.

Table 4.3 The selection results of question Q_5 and Q_{16}.

	Question-id		Q_5							Question-id		Q_{16}					
	Correlation	0.78	AN2: 'Using Internet Explorer'							Correlation	0.77	AN11: 'Secure paying'					
1	**Learner-id**				**L_2**				**3**	**Learner-id**				**L_5**			
	Tutor Suitability				0.83					Tutor Suitability				0.73			
	WC	C_L	WE	E_L	WA	A_L	WT	T_L		WC	C_L	WE	E_L	WA	A_L	WT	T_L
	1	1	0.5	0.80	0.5	0.5	0	0		1	1	0.5	0.40	0.5	0.5	0	0
	Learner-id				L_3					Learner-id				L_4			
	Tutor Suitability				0.78					Tutor Suitability				0.38			
	WC	C_L	WE	E_L	WA	A_L	WT	T_L		WC	C_L	WE	E_L	WA	A_L	WT	T_L
	1	1	0.5	0.6	0.5	0.5	0	0		1	0.3	0.5	0.40	0.5	0.5	0	0
2	**Learner-id**				**L_3**				**4**	**Learner-id**				**L_5**			
	Tutor Suitability				0.78					Tutor Suitability				0.73			
	WC	C_L	WE	E_L	WA	A_L	WT	T_L		WC	C_L	WE	E_L	WA	A_L	WT	T_L
	1	1	0.5	0.6	0.5	0.5	0	0		1	1	0.5	0.40	0.5	0.50	0	0
	Learner-id				L_2					Learner-id				L_4			
	Tutor Suitability				0.76					Tutor Suitability				0.44			
	WC	C_L	WE	E_L	WA	A_L	WT	T_L		WC	C_L	WE	E_L	WA	A_L	WT	T_L
	1	1	0.5	0.80	0.5	0.25	0	0		1	0.3	0.5	0.40	0.5	0.75	0	0

The learner selected is in bold face.

DISCUSSION OF THE RESULTS

The first results of the application of LSA suggest that it delivers as expected. The prototype identified the correct AN for 12 out of the 16 questions (75%). Moreover, two developers of the Learning Network in question, evaluated the text fragments, three for each question, that the prototype suggested. Ignoring the very small discrepancies in judgements between these experts, for about six to seven of the questions, one or more text fragments were identified that in their opinion were useful for answering those questions. This figure seems far less accurate. The experts, however, indicated that 5 of the 16 questions posed were beyond the scope of the contents of the AN studied. As a consequence, the AN could not possibly contain any useful fragments. Taking this into account, six to seven questions with useful text fragments out of a total of 11 is a much better score (about 60%, for details, see Van Rosmalen *et al*., 2006). Together the results are encouraging, taking into account the limited nature of the test. For about 75% of the questions the correct AN was identified; this means that in 75% of the cases content competent peer tutors may be selected. These will then be helped by providing them with text fragments; in the majority of the cases, at least one of those fragments was deemed useful by experts.

Also the first test of the selection rules is positive. The selections illustrate that we can balance the selection of peers with the help of workload and eligibility. In selection 1, the value of eligibility favoured Learner 2 over Learner 3, i.e. it prioritized the selection of a student in the same study phase. However, if we pose the question again, the balance is shifted because of the workload of Learner 2. In selection 3, Learner 5 is selected based on his content competency. But note that Learner 5 is selected again in selection 4. Learner 4 has not been involved yet, Learner 5 is simply too good. Obviously, the test has too limited a nature to allow one to draw general conclusions for the application of the selection rules in practice. How learners will behave and particularly how they will appreciate the selection rules should be assessed in empirical tests.

Conclusion

We started our discussion by arguing that a model is needed to organize and support learner-related interactions in Learning Networks in a more efficient manner. For one type of support actions, answering content related questions, we articulated our requirements and proposed a model. The test results of the first prototype showed that we were able to identify the relevant AN for some question, to select text fragments useful for answering the question, and to test our peer selection formula to the extent that it warrants carrying out an empirical study with ‘real’ students. This indicates that we can at least satisfy two of our requirements ‘involvement’ and ‘support’. The first requirement ‘the model has to alleviate the support task for the staff tutor while maintaining quality’ one can only test empirically. Most steps of the model are executed automatically. Nevertheless, empirical evidence has to shed light on how many questions will be resolved, what the quality of the answers is, and how much involvement of a staff tutor still is needed. The final requirement ‘portability’ is not yet met, but such is the nature of prototypes. The portability of the model is influenced by a number of factors. First of all, it should be possible to move the model from one system to another. This can be achieved by following for instance a service or an agent oriented approach. At a detailed level, the ‘portfolio’ of the learner should be accessible in an interoperable format. This can be achieved by applying the IMS-LIP standard (IMS-LIP, 2001). Moreover, for LSA to work efficiently, the course corpus has to be retrievable in a standard manner. This can be achieved by adopting the widely accepted IMS-CP standard (IMS-CP, 2003).

The next task now will be to carry out actual experiments. Questions to be addressed are (1) if and to which extent is the task of the staff tutor alleviated; (2) are peer learners capable and willing to answer questions; and (3) is there a measurable effect on the social cohesion of the Learning Network. Our first experiment, just started, will focus on questions 1 and 2. Connected and subordinated to these questions, a number of critical conditions and parameters have to be determined, among others: the optimal size of the document corpus, the precise contents of the guidelines and the optimal size of the text fragments, the best size of the group, and the weights related to the selection of the peer tutors.

Acknowledgements

The authors would like to thank André de Jong and Maurice Brouwers, who contributed to the building of the test application. They are also indebted to an anonymous referee, who provided many comments that led to significant improvements of the manuscript. This project was funded by the OTEC Technology Development Programme in the project ASA (http://www.learningnetworks.org/) and by the European Commission in TENCompetence (IST-2004-02787) (http://www.tencompetence.org/).

Chapter 5

Knowledge matchmaking in Learning Networks: Alleviating the tutor load by mutually connecting learning network users

Van Rosmalen, P., Sloep, P., Brouns, F., Kester, L., Kone, M. and Koper, R. (2006). Knowledge matchmaking in Learning Networks: Alleviating the tutor load by mutually connecting learning network users. British Journal of Educational Technology, Vol. 37 (6), 881-895.

Abstract

Tutors have only limited time to support the learning process. In this paper, we introduced a model that helps answering the questions of students. The model invoked the knowledge and skills of fellow students, who jointly formed an ad hoc, transient community. The paper situated the model within the context of a Learning Network, a self-organised, distributed system, designed to facilitate lifelong learning in a particular knowledge domain. We discussed the design of the model and explained how we selected and supported capable peers. Finally, we examined the calibration of the model and a simulation, which was intended to verify if the model is fit for use in experiments with students. The results indicate that, indeed, it is possible to identify and support capable peers efficiently and effectively.

Introduction

In modern learning settings, students typically spend a significant amount of time learning online. In this respect, these settings diverge from the classroom-based, face-to-face learning situations that we are all so familiar with. But they differ in more significant ways too. The advent of the knowledge economy and the individualisation of our society are two leading factors that underpin the increasing demand for flexibility: students want to be able to study at the place, time, and pace of their own choosing (logistic flexibility); also, students are unwilling to submit themselves to pre-planned, rigid programmes, but want their prior competences honoured and their specific study plans catered for (subject matter flexibility).

These developments called for a new perspective on learning that has become known as lifelong learning, which upholds a central position for the learner. The lifelong learner is self-directed, and can perform different formal and informal learning activities in different contexts at the same time. Inherent to this is that learning activities take place in environments populated with learners in any given domain of knowledge with different levels of competence, varying from novices to top experts, and different foci, varying from practitioners to researchers and developers. To accommodate lifelong learners adequately, it is necessary to *maintain a record* of their growth in competency in a persistent and standard way to ensure that they can search for new learning facilities that fit and extend their current knowledge.

Networks for lifelong learning ('Learning Networks') embody these changes and at the same time seek to address the challenges they pose. A Learning Network (Koper *et al.*, 2005) is a self-organised, distributed system, designed to facilitate lifelong learning in a particular knowledge domain. A Learning Network is special in that it follows a particular domain model (Koper, 2006) that defines the concepts used and the overall architecture. A Learning Network is specific for a certain domain of knowledge (e.g. an occupation) and consists of three entities:

1. users (lifelong learners): people with the intent to learn and the willingness to share their knowledge in the specified domain;
2. Activity Nodes, i.e. a collection of learning activities that are created and shared in order to exchange knowledge and experience or to develop competences in the domain;
3. a set of defined learning outcomes, or ‘goals’ (e.g. competence levels).

Learning and teaching in a Learning Network may have some unfortunate side-effects:

1. Users are unlikely to arrive in groups, nor will they share their objectives or background. Missing the social structure of a class, students easily become socially isolated, ‘lone’ learners (Kester *et al.*, 2006).
2. The heterogeneity of the users and the lack of a readily available social structure that provides mutual support, makes a large demand on the tutors (Bacsich & Ash, 2000; Bartolic-Zlomislic & Bates, 1999; Koper, 2004). Tutors in an online learning context (Anderson, 2004) are no longer restricted to well-defined and pre-planned tasks but have to adopt to user needs on the fly. The tutor has to make provisions for the negotiation of activities to meet users’ unique learning needs, and equally well has to stimulate, guide and support the learning in a way that responds to common and unique user needs.

Moreover, and of particular relevance in the context of this paper, there is the additional challenge that Learning Networks are not meant merely to serve formal learning but also to cater for informal learning. For informal learning, there may not be any staff at all. However, also informal learners will have questions on where to start, how to proceed, how to understand and apply the available Activity Nodes, or will want to have their contributions assessed. As a consequence, there is a need to organise *and* support both formal and informal learning.

In this paper, we will concentrate on one element of this challenge, to wit, answering questions related to the content studied. For a tutor, this is considered a time consuming and disruptive task (De Vries *et al.*, 2005). Yet, learning may improve if learners can ask questions and receive timely and relevant feedback (Howell, 2003). A number of models exist that address this particular problem. *Expertfinder* (Vivacqua & Lieberman, 2000) is an agent that classifies novice and expert knowledge by analysing documents created while working in the domain of Java programming. The model tries to distribute the question load evenly over several experts. It also prioritises not the best expert available, but someone whose knowledge level is close to the questioner's level. This way, it is more likely to bring together people who share a similar mental model of the problem discussed. Interestingly, in 50% of the cases in which an answer was supplied, the expert did give an answer, not directly, but by looking it up. Yenta (Foner, 1997), a multi-agent, matchmaker system, has been designed to find people with similar interests and introduce them to each other. The similarity of interest is based on the assumption that two users have similar interest if both possess similar documents (emails, newsgroup papers and files). FAQO (Caron, 2000) relies on the use of latent semantic analysis (LSA) (Landauer, Foltz & Laham, 1998; Van Bruggen *et al.*, 2004), a technology with a relatively widespread use in educational settings (Haley, Thomas, Roeck & Petre, 2005). LSA connects related words in a number of steps (e.g. in documents in Computer Science the words human, computer and interface are related). In this way, although the actual keywords in the documents might differ, if there is sufficient similarity, documents are associated. FAQO allows the users to query questions in natural language in order to find relevant documents to solve their problems for specific technical problems.

In our model, we combine a number of the characteristics of the previously mentioned models. Crucially, we seek to solve content-related questions by involving peers in answering them (peer tutoring). To that end, we identify appropriate and available users as well as documents, and bring these together in a so-called *ad hoc, transient* community. Such a community is *ad hoc* in that its only purpose is to solve a particular question; it is transient in that it vanishes the moment the question has been solved. In our view, *ad hoc*, transient communities are particularly well suited to assist peer tutoring (for a detailed discussion on the underlying theoretical aspects of our model on learning in communities and peer tutoring, see Kester *et al.* (2006) and Kester *et al.* (2007). Obviously, one will have to heed the lessons learned on community building and peer tutoring.

First, for a social space to emerge, one should establish continuity of contact, recognisability of members, and a historical record of actions (Kollock, 1998). Furthermore, to assure the liveliness of a community, it should be populated with a heterogeneous group consisting of veterans and newbies; connectors, mavens, and salesmen; and lurkers and posters (Preece, Nonneke & Andrews, 2004). Also, to facilitate cooperation in a community, clear boundaries and a clear set of rules that can be monitored and sanctioned within the community are required (Kollock & Smith, 1996). With respect to peer tutoring, we found out, among other things, that peer tutoring enhances the social embedding of students in a learning environment that facilitates social processes as engagement, commitment, and a sense of belonging, and that peer tutoring does indeed help tutors and tutees to achieve higher learning outcomes (Fantuzzo, Riggio, Connelly & Dimeff, 1989).

At this point in time, we do not test any of these community formation conditions, but provisionally assume that we can sufficiently support the community with the help of e-portfolios, the expected heterogeneity of a Learning Network, and by setting clear guidelines for the tasks supported. Similarly, although we will have to validate in future experiments that the expected benefits for learners and tutors will materialise, we provisionally assume them to be present. In the remainder of this paper, we concentrate on the main assumptions underlying our model, i.e. that we can indeed identify appropriate and available peers and documents.

We now explain our model by depicting it in a context, a Learning Network, and by describing its current implementation. In the sections that follow, we will discuss the calibration of our model and the results of a simulation. The simulation will show how well we can map a set of predesigned users' questions onto the Activity Nodes in a selected Learning Network. With this information, we can identify capable peers and relevant textual resources in the network.

Model implementation

A LEARNING NETWORK

In order to describe clearly the context of our model implementation, we introduce a Learning Network example (Figure 5.1). Suppose we have a Learning Network in domain D, e.g. psychology, with a set of Activity Nodes A_1–A_{10}. Moreover, we have a Learning Network user P (Anne) who has formulated a goal that can be achieved by studying A_1, A_2, A_3, A_6, A_7, A_9 and A_{10}. Next, we know that Anne, given her working experience and prior studies, has exemptions for A_5 and A_6, and has already successfully finished A_7. Finally, let us assume that Anne runs into problems while studying A_1. She has a problem understanding the relations between a number of concepts, and as a consequence, she is not able to complete an assignment. She studies some additional literature and searches the Web, though to no avail. Anne, studying on her own and thus out of touch with any peers, decides to pose a question to the 'online tutor'; she describes the general problem and her question.

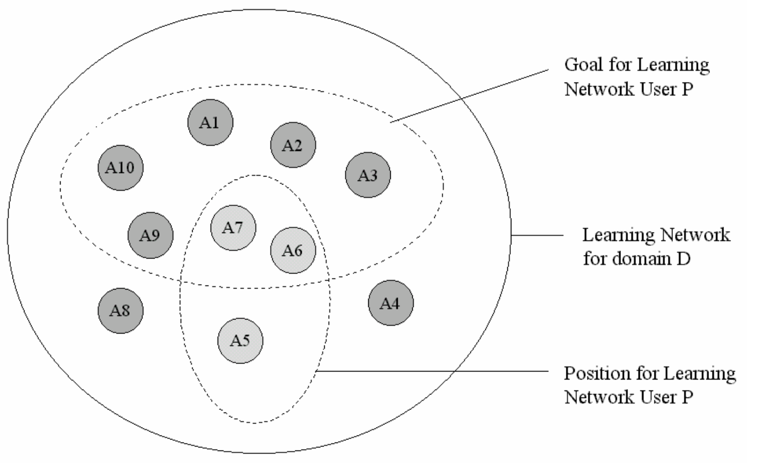


Figure 5.1 A Learning Network for domain D (psychology), user P (Anne) and activity nodes A1–A10.

The 'online tutor' in our model consists of an ad hoc, transient community populated with peer users who have complementary content expertise. The goal of this community is to share knowledge and jointly come to an answer about the question in point. The central aim of our model (Table 5.1) is to set up and support the ad hoc, transient community that will help answer the question within an agreed timeframe (e.g. 2 days) and to a mutually agreed quality (i.e. the peer users decide together).

Table 5.1 The main steps of the model.

Pre-condition	A Learning Network with a set of Activity Nodes and a set of users with their profiles
Main steps	1. *Anne* poses a question. 2. The *system* determines: - the most relevant text fragments; - the appropriate Activity Node(s); - the most suitable users. 3. The *system* sets up a wiki with the question, the text fragments and guidelines. 4. The selected *users* receive an invitation to assist. 5. *Anne and the users* discuss and phrase an answer in the wiki. 6. If answered (or after a given period of time), *Anne* closes the discussion and rates the answer.
Post-condition	The answer is stored.

The prototype of the model (Figure 5.2) consists of five modules. For the users, we have a Learning Network, its Activity Nodes and a question interface. They are implemented in an instantiation of the Moodle environment (http://www.moodle.org). Additionally, each time a question is posed, a wiki is made available that includes the question and three documents selected from the Learning Network material. The wiki is populated with a selection of users who are invited to help. For the designer and for the run-time system we have three modules: a general text parser (GTP; Giles, Wo & Berry, 2001), a GTP calibrator (GTP usability prototype [GUP]; De Jong *et al.*, 2006) and a tutor locator (Agents for Support Activities [ASA] tutor locator [ATL]; Brouwers *et al.*, 2006). We use GTP, an LSA implementation, to map the questions on the documents in the Learning Network. The GTP module returns correlations between the question and documents. The correlations are used to determine the Activity Node to which a question fits best and to select relevant text documents. The application of LSA, however, is not straightforward. It depends on the corpus (the documents in the Learning Network) and its application. To assure optimal use, one has to calibrate a set of parameters. The GUP module has been built to ease the calibration. Finally, the ATL module takes care of the selection of the peer users who will assist. The selection is based on a weighted sum of four criteria that are derived from the users' background and performance. The designer can adjust the weightings.

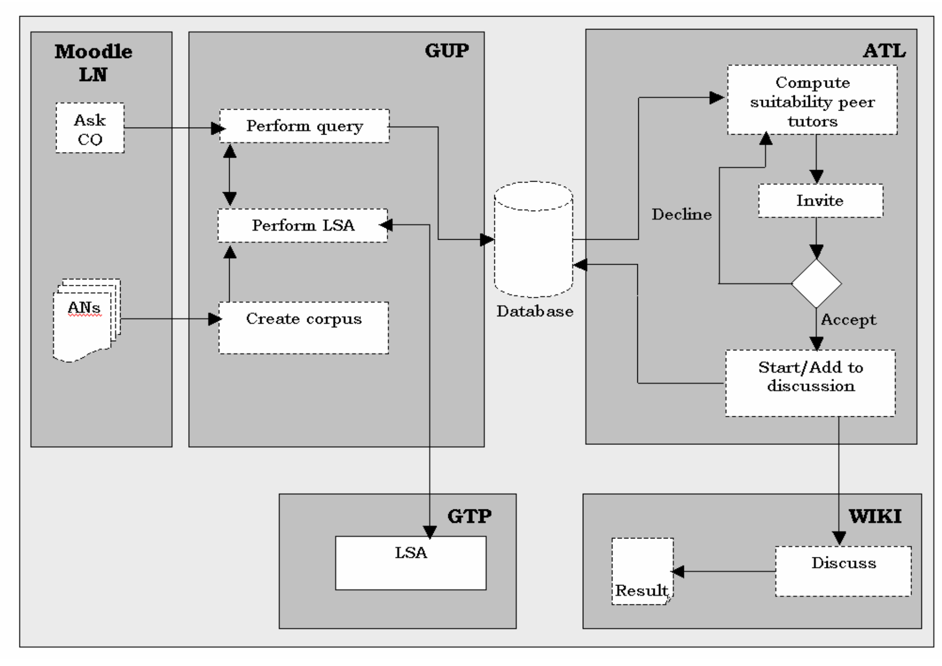


Figure 5.2 The main modules of the model:
LN, learning network; GTP, general text parser; GUP, GTP usability prototype; ATL, Agents for Support Activities (ASA) tutor locator; LSA, latent semantic analysis; CQ, content question.

The model covers three phases. In the first phase, the working context is defined. The model is connected to the Learning Network. All text of this Learning Network is captured and put into a corpus for further analysis. This includes the calibration of a suitable set of parameters for LSA. The next phase starts when a user poses a question (Figure 5.3). First, the Activity Node(s) is (are) identified to which the question fits best. This is done by mapping the question with LSA on the documents of the corpus and to look for the three documents with the highest correlations. Later, the same three documents are given to the ad hoc community to help the users get a quick overview of relevant documents in relation to the question. We chose three documents because it should be sufficient to distinguish and not too much to be read by the supporting peers. However, this number may be altered if experience suggests so. Next, knowing to which Activity Node the question fits best, the ATL module can identify peers who are competent in the pertinent Activity Node(s). ATL selects three to five users who, according to four different criteria, are best equipped to answer the question (Kester *et al.*, 2006). The suitability ranking is a weighted sum of *tutor competency*, *content competency*, *availability,* and *eligibility*:

1. The *tutor competency* is the ability of a user to act as a tutor. The tutor competency is derived from a combination of data logging, i.e. from the frequency and size of the contributions, and ratings on answers given previously.
2. The *content competency* indicates if a user has successfully completed the Activity Nodes related to the question.
3. *Availability* is based on the actual availability as derived from the personal calendar of the users and on their past workload. Someone who has recently answered none or only a few questions should be preferred over someone who has answered many.
4. Finally, *eligibility* measures the similarity of the users. It can be used to favour the selection of users with an almost identical competence level.

With all information available, ATL now attempts to form an ad hoc community. It creates a wiki and invites the selected users. The invitation includes the question, some guidelines and a small set of documents that have been identified as being relevant to drafting an answer.

Finally, in the last phase, the users jointly formulate an answer to the question. After some time, the peer tutoring process ends and a response becomes available. Ideally, the process ends because the question-asking user (tutee) is satisfied with the answer. However, if this is not the case, it may also end because a predefined period of time has elapsed or because the participants agree to end it. Whatever the reason, the tutee should rate the performance of the peer tutors involved. If necessary, these data are used to alert the institution-bound tutor that there is an unresolved question or (in combination with other logging data) that some users perform suboptimally.

Question 1
Wanneer ik me laat registreren om gebruik te kunnen maken van een chatroom kan ik dan met dezelfde registratie meerdere pseudoniemen gebruiken? [When I register for a particular chat room, does my registration allow me to use several pseudonyms?]

Proposed text fragments*	Activity Node						
237.txt	Net chatten en Netsletten (Net chat)						
Suitability	Not useful	**1**	**2**	**3**	**4**	**5**	useful
Your text fragment(s) in case of a rating of 1 or 2							
58.txt	Omgaan met ongewenste inhoud (Dealing with undesired content)						
Suitability	Not useful	**1**	**2**	**3**	**4**	**5**	useful
Your text fragment(s) in case of a rating of 1 or 2							
329.txt	Zoeken op het web (Searching on the web)						
Suitability	Not useful	**1**	**2**	**3**	**4**	**5**	useful
Your text fragment(s) in case of a rating of 1 or 2							

* Open the link to see the text

Figure 5.3 An example of a question and the way to assess the proposed text.

Method

Before actual experiments with the model, involving real people, can be carried out, one has to prepare the required data structures (the text corpus) and calibrate the model, i.e. determine a default setting for the LSA parameters and for the weights of the peer selection criteria. In this paper, we concentrated on the corpus preparation and the LSA parameters. The selection of proper weights is out of the scope of this paper; it will be determined in a future experiment with students. We carried out a partial simulation of the model to ensure that the model operates according to its design. For a set of predesigned questions, we looked into how well we can map them to the Activity Nodes of the Learning Network. This is of key importance for the selection of peer users. Moreover, we asked the designers of the Learning Network to rate the text documents that are selected for the users.

THE CORPUS OF THE LEARNING NETWORK

Fortunately, at the start of the work described, we had a Learning Network at our disposal developed for a study on navigation (Janssen *et al.*, 2007). The domain of this Learning Network is ‘Internet Basics’, a collection of texts, links and tasks that aim to instigate a basic understanding of the Internet. It contains 11 Activity Nodes, each of which introduces a different aspect of the Internet, ranging from ‘web searching’, ‘chatting’ to ‘worms and horses’. The Activity Nodes consist of an introduction, exercises, references to external web pages for further study, and an assessment. The Learning Network matches our two start requirements, i.e. (1) an accessible text corpus, a combination of the Moodle learning environments and external web pages; and (2) the users’ progress could be tracked by the data available from the Activity Node assessments. The corpus was extracted manually. It contained the Moodle pages and external web pages; assessment questions were left out, however. These questions were used to calibrate the model. The Activity Node of an assessment question is obvious and thus could be compared to verify the Activity Node determined with the help of LSA. The language of the corpus is Dutch - references to documents in English were ignored - admitting, though, a considerable English internet vocabulary. The documents were saved as ‘text only’, a quick way to get rid of all non-textual elements. The documents were used as raw input; this means that no further corrections were applied such as removing irrelevant documents, diacritical signs or misspellings. The final corpus was relatively small. It consisted of 327 documents ranging in size from 50 to 23 534 bytes (41 documents smaller than 250 bytes, 50 documents above 3000 bytes). The corpus contained a total of 82 986 words divided over 10 601 terms, 4440 of which occur in at least two documents.

THE CALIBRATION OF THE LSA-PARAMETERS

Having created the corpus, our first action was to calibrate the LSA parameters. A calibration is primarily focused on finding an optimal combination of parameters connected to a model. However, in our case, it is equally important to find a way to define the parameters with a predefined, limited number of steps that can be easily repeated and automated at a later stage. In this way, we ensure that we can apply our model in real practice. An overview of applications with LSA (Haley *et al.*, 2005) reveals that there is no straightforward procedure to determine the LSA parameters. The parameters are influenced by the corpus and the way LSA is applied. We selected the five steps (Giles *et al.*, 2001; Wild, Stahl, Stermsek & Neumann, 2005) that should be the most important: the definition of a correlation measure and method, corpus preprocessing, normalisation, weighting and dimensionality. We did not carry out, however, an exhaustive test with different combinations of parameters. Instead, we started with an initial combination of parameters based on the results reported (Van Bruggen, Rusman, Giesbers & Koper, submitted; Wild *et al.*, 2005), and in each step, we tested one parameter in a limited number of test runs. Each time we continued to the next step, we only used the best result(s) from the previous step.

CORRELATION MEASURE AND METHOD

For our correlation measure, we used cosine similarity. Our method directly follows from our model. We used LSA in two closely related ways. First, we used LSA to identify to which Activity Node(s) the question posed fits best. This information is used to identify peers that are competent in the pertinent topic. Second, we wanted to select the three documents (text fragments really) in the corpus that were most suited to assist the peers in answering the question. We combined the two by selecting the three best correlating documents and by assigning one point to each Activity Node that a document originates from. This resulted in a maximum of three Activity Nodes that the question relates to. We used the result of the mapping on the Activity Nodes to select the parameter combination with which to continue. In our case, the questions, 16 in total, were chosen from the original assessment questions of the Learning Network. Therefore - in principle - each question should map to one known Activity Node.

Preprocessing the corpus can consist of stopping (removing ‘meaningless’ words) and stemming (reducing terms to their semantic stem). Because we did not have access to a stemming application for Dutch, we only considered stopping. Moreover, given the size of our corpus, we decided to follow a recommendation by Van Bruggen *et al* (submitted) to create our own stop lists based on the term frequency in the corpus. The stop list consisted of the terms that covered 33% (22 terms) and 50% (91 terms) respectively of the overall term frequencies with the exception of terms that were judged corpus specific. By way of comparison, we also used a ‘general’ Dutch stop list (Oracle Text Reference: Release 9.2, 2002). For our corpus, this resulted in a reduction of 188 terms. Finally, in each run (until the actual dimensionality step), we chose to limit the number of singular values (i.e. the number of dimensions) to 40% of the sum of the singular values (Wild *et al.*, 2005). Next, as previously reported, our corpus showed quite a spread in document lengths, while at the same time the number of documents per Activity Node proved limited. Therefore, we decided to use *normalisation*. It makes the norm of each document vector equal to one. This has the effect that documents with the same semantic content are ranked equal in the question query. Next, we applied the three available types of *Global Weighting* and finally, in the last step, we determined the best value for the *dimensionality* by comparing the initial value of 40% of the sum of the singular values to 30 and 50%.

A SIMULATION OF THE MODEL

After having studied the Learning Network and with a view to the simulation, we formulated a new set of 16 questions, each connected to one Activity Node. The questions were once again mapped on the Activity Nodes, and the results were compared with their known Activity Nodes. Please note that, this time, only the parameter combination that performed best in the calibration was applied. Next, we asked two of the designers of the Learning Network to rate, on a 5-point scale (Figure 5.3), the suitability of the text fragments selected through the application of LSA. Obviously, a question may go beyond the content discussed in the Activity Nodes. In such cases, the text documents identified by LSA have little bearing on the question; they can only serve to start off a discussion. Therefore, we instructed the designers to assess the suitability of the documents identified *relative to* the available text. This means that also, a document that only starts off the discussion of a question should be rated high in case there is no better alternative available. In addition, in case of a low rating, we asked the designers to indicate a better alternative from within the corpus.

Results and discussion

The first part of our study aimed to determine the LSA parameters in a fixed, limited number of steps and a limited number of test runs. We achieved the following results. First, (Figure 5.4) we compared three stopping approaches: 33 and 50%, and a general Dutch stop list (Runs 1–3). We were able to identify correctly the Activity Nodes of 5, 11, 11 questions respectively. Second, as a result of this, we continued with normalisation for the 50% and the Dutch stop list (Runs 4 and 5). The number of correctly recognised Activity Nodes remained 11. However, the questions with a single match increased, in particular in Run 5 (Dutch stop list). We kept normalisation, continued with the Dutch stop list and compared global weights ‘inverse document frequency’, ‘logarithm’ and ‘entropy’ (Runs 6–8). This time, the results improved to 12, 14 and 15. For the last step, the dimensionality, we continued with the setting of Run 8 to Run 9 (30% singular values) and Run 10 (50% singular values). The overall results remained the same. The number of 100% recognitions increased by one. Finally, we carried out one additional run, which we had not planned beforehand; we used the 50% stop list in order to check if this would improve our results. The other parameters followed the settings of Run 9. The result was good (15 out of 16) but not an improvement.

Overall, the results are encouraging. First of all, - at least for this corpus - it seems possible to determine such a combination of parameters that an important requirement of our model can be fulfilled: the mapping of a question to the appropriate Activity Node and, on the basis of this information, the ability to select appropriate peers. Second, the results suggest that the approach taken to calibrate the parameters in a fixed setting with a limited number of test runs is sound. Nevertheless, one should be open to retrace one’s steps, in particular, if the results are very close (as in our normalisation step) and improvements develop insufficiently. Because Runs 9 and 10 had identical results, both were kept for the simulation.

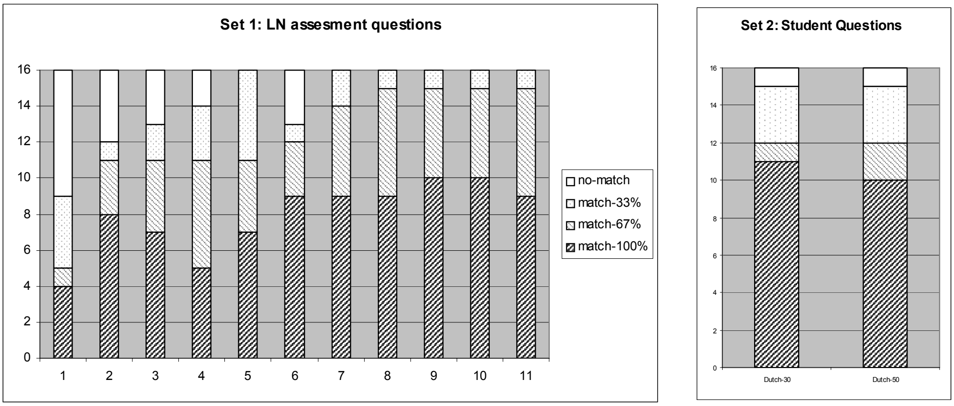


Figure 5.4 The mapping of the questions on the Activity Nodes: Assessment questions in the calibration runs (left); final questions (right)

Having completed the calibration, we devoted the second part of our study to simulating part of the model. We created 16 questions that we felt students may well have asked, mapped them on the Learning Network and invited two of the designers of the test Learning Network to rate the suitability of the proposed text fragments with respect to the questions. First, the model identified the correct Activity Node for 12 out of the 16 questions (Figure 5.4). Case one (the settings of Run 9) did slightly better in the 100% recognition category. For this case (Figure 5.5), subsequently, the designers rated the supplied text fragments. Of the 16 questions,

1. 6 (38%) and 4 (25%) respectively had at least one relevant text fragment (Rating 4 or 5);
2. 1 (6%) and 2 (13%) respectively had a text fragment that was of some use; and
3. 9 (56%) and 10 (62%) respectively had no suitable text fragments connected to them.

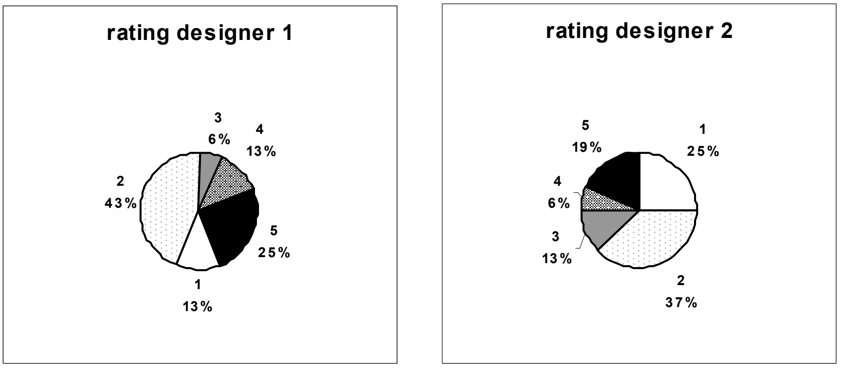


Figure 5.5 The rating by designer 1 and 2 of the suggested text fragments on a 5-point scale: 1 is not useful, 5 is useful.

The results of the mapping are worse than in the calibration, but are still quite accurate with a recognition of 75%. The suitability of the text fragments looks far less accurate; approximately 40% of the questions receive one or more fragments rated 3 or above. However, they do answer our expectations very well, given the conditions we work with:

1. We chose to only forward the first three fragments in order not to overload the users. Obviously, we thus run the risk that relevant fragments are left out. FAQO (cf. Introduction), for instance, returns a top 10 and, indeed, answers 4–10 do give a relevant contribution.
2. The corpus is relatively small; this lowers the likelihood to find a relevant text for each question. Designer 2 confirmed that for 6 out of 10 questions (with a text rating of only 1 or 2), he could not identify a better alternative. In a real implementation, one can stepwise improve the likelihood of finding a relevant text by adding the answers of solved questions to the corpus.
3. Finally, as with Expertfinder and Yenta (cf. Introduction), our intention is not so much to identify *the* answer. Our focus is on questions that are not readily answered by simply looking up the Learning Network contents. But we do want to give the ad hoc communities a solid starting point to the extent that the corpus makes that feasible.

Conclusion

In this paper, we described a model that intends to alleviate the support task of tutors. The model does so by invoking the knowledge and skills of fellow students, who jointly form an ad hoc, transient community. We described how we calibrated LSA for an existing Learning Network. Subsequently, and for the same Learning Network, we checked with a simulation whether the model is fit for experimentation with students. In our opinion, the results are promising. For

75% of the questions, we were able to identify to which Activity Node they belonged; for approximately 40% of the questions, we could suggest one or more text fragments that could be useful when formulating an answer. Moreover, we were able to arrive at our results in a systematic way. The same steps can be followed for a new corpus or if the changes to an exiting corpus are relatively small, the known settings can be reapplied in just one additional run. Important characteristics of the procedure followed are that (1) it is relatively straightforward; there are no experts needed to apply it; and (2) it can be automated to a very large extent. Furthermore, the requirements to use the model are limited. They are restricted to having an accessible text corpus and accessible learner progress information. In a final system the first requirement, for instance, can be realised by adopting the widely accepted IMS-CP standard (IMS-CP, 2004).

Obviously, there are a number of issues to be considered. First, the model has only been applied to questions that exactly match one Activity Node. It is fair to expect that, in real practice, some of the questions will cover not just one but more Activity Nodes. This may complicate the recognition and thus dilute the results. Next, as shown by some of the results, the approach is sensitive to the size (and content) of the available corpus. We do not know (yet) what the minimum size of a corpus should be. We also still have to determine a working combination of weights of the suitability ranking (*tutor competency*, *content competency, availability,* and *eligibility*). These issues, however, do not lend themselves to simulation and should be addressed in empirical tests.

The results indicate that the model is ready for use in experiments with students. A first experiment is planned for the second part of 2006. Here we will also investigate and optimise the community formation conditions discussed in the Introduction. Ultimately, the experiment is meant to investigate our main hypothesis, to show that the task of staff in answering questions can be significantly alleviated by following our peer-tutoring model.

Acknowledgements

The authors thank the designers of the Learning Network Internet Basics, in particular, José Janssen and Colin Tattersall, for their support, and Jan van Bruggen for his indispensable comments and advice on the use of LSA. The authors' efforts were funded by the Educational Technology Expertise Centre (OTEC) Technology Development Programme in the project ASA (http://www.learningnetworks.org) and by the European Commission in TENCompetence (IST-2004-02787) (http://www.tencompetence.org).

Chapter 6

A Learner Support Model Based on Peer Tutor Selection: Experimental Results

Van Rosmalen, P., Sloep, P.B., Brouns, F., Kester, L., Berlanga, A., Bitter, M. and Koper, R. (submitted). A Learner Support Model Based on Peer Tutor Selection - Experimental Results.

Abstract

Tutors have only limited time to support students. In this paper, we discuss a model that addresses the question of how to help students answer content-related questions. A small group of students is created, which consists of the student who asked the question and peers who should be able to answer it. Criteria used to compose the group are the content of the question in relation to the knowledge and skills of the peers. The model supports the collaboration with text fragments selected from the study materials. We will introduce the model and briefly discuss the results of the calibration and a simulation of the model. Finally, we will discuss the outcome of an experiment with two groups of approximately 50 students, who used the model for a period of 8 weeks. The results indicate that the students positively value the model and that it is possible to solve a substantial number of their questions.

Introduction

In modern learning settings, students typically spend a significant amount of time learning online. The advent of the knowledge economy and the individualization of our society are two leading factors that underpin their increasing demand for flexibility: students want to be able to study at the place, time and pace of their own choosing (logistic flexibility); also, students are unwilling to submit themselves to pre-planned, rigid programmes, but want their prior competences honoured and their specific study plans catered for (subject matter flexibility). However, as in traditional, on-site settings, students will have questions on where to start, how to proceed, how to understand and apply the available study material, or they will want to have their contributions assessed. In this paper, we will concentrate on one element of this challenge, to wit, answering questions related to the content studied. For a tutor, this is considered a time-consuming and disruptive task (De Vries *et al.*, 2005). Yet, learning may improve if learners can ask questions and receive timely and relevant feedback (Howell, 2003). To address this issue, we have developed a model of how best to organize this with the help of peer-tutoring and we have instantiated this in software so as to be able empirically to test the model.

Our model seeks to solve content-related questions by involving peers (peer tutoring) in answering them. To that end, we identify appropriate and available students as well as documents, and bring these together in a so-called *ad hoc, transient* community. Such a community is *ad hoc* in that its only purpose is to solve a particular question; it is transient in that it vanishes the moment the question has been solved. The model distinguishes (Table 6.1) six main steps of which the second step depends on a language technology called Latent

Semantic Analysis (LSA). In the following section we will introduce the current implementation; next we will quickly go into the results of the calibration and a simulation of a prototype of the model. The main focus of the present article is an exploration of the results of a field experiment with two groups of approximately 50 students, who use the model in the context of a Learning Network on the topic of 'Internet Basics'.

For a proper perspective on the present article, bear in mind that it is part of a series. Kester *et al.* (2007) examine in details the theoretical aspects of learning in communities. Van Rosmalen *et al.* (2006) discuss the use of LSA, Van Rosmalen *et al.* (2008) describe the technological aspects of the model and discuss how to tweak its parameters. Finally, see Koper *et al.* (2005) for the wider context of our model: a Learning Network, i.e. a self-organised, distributed system of lifelong learners, Activity Nodes and competences, designed to facilitate learning. The present article completes the design and development cycle of the model.

Table 6.1: The main steps of the model.

Pre-condition	A Learning Network with a set of Activity Nodes and a set of users with their profiles indicating their progress with regard to the Activity Nodes
Main steps	1. *Anne* poses a question. 2. The *system* determines: a. the most relevant text fragments; b. the appropriate Activity Nodes; c. the most suitable students. 3. The *system* sets up a wiki with the question, the text fragments and guidelines. 4. The selected *students* receive an invitation to assist. 5. *Anne and the peer-students* discuss and formulate an answer in the wiki. 6. If answered (or after a given period of time) *Anne* closes the discussion and rates the answer.
Post-condition	The answer is stored.

Model implementation

A prototypical software application has been developed to test the model. Through the virtual learning environment Moodle (http://www.moodle.org), the students are exposed to a Learning Network, its Activity Nodes and a question module (AskCQ) (Figure 6.1) that organises and structures the question answering process.

Uw vragen

klik hier om een nieuwe vraag te stellen

Status	Uw Vraag	De Antwoord-Wiki	Vraag afronden
Klaar	Er zijn verschillende mogelijkheden om op internet te komen heb ik zojuist gelezen. Nu heb ik al ads...	Naar de Antwoordwiki	Afgerond
Klaar	wat is een browser. Vraag-Antwoord-module kunt u vragen stellen over de inhoud van de cursus. Uw vr...	Naar de Antwoordwiki	Afgerond
Bezig	Internet biedt allerlei mogelijkheden om te communiceren. Via email kunnen berichten verstuurd worde...	Klik hier voor de antwoord-wiki	Klik hier om deze vraag af te sluiten

Uw antwoorden

Status	De Vraag	De Antwoord-Wiki
Klaar	Ik heb net gelezen dat er ook chatgroepen juist populair zijn vanwege netsletten.Kan ik die al herke...	Naar de Antwoordwiki
Bezig	wat is het voordeel van de opera browser ten opzichte van de microsoft variant. Welke browser kan ik...	Naar de Antwoordwiki
Bezig	wat is een internet browser?, HTML, Flash, XML, Java, Exporer	Naar de Antwoordwiki
Klaar	Laatst was ik aan het chatten en in de chatbox waar ik in zat hadden mijn mede-chatters allemaal moo...	Naar de Antwoordwiki

Figure 6.1 Part of the interface of the AskCQ module with (1) 'Uw vragen' (your questions) with a link to pose new questions and an overview over the posed questions: 'Status' (Klaar = ready; Bezig = busy), 'Uw vraag' (your question), 'De Antwoordwiki' (a link to the wiki), 'Vraag afronden' (End a question: Afgerond = rated; or a link to the rating); (2) 'Uw antwoorden' (your answers) with an overview of answer given: 'Status' (Klaar = ready; Bezig = busy), 'De vraag' (the question), 'De Antwoord-Wiki' (a link to the wiki).

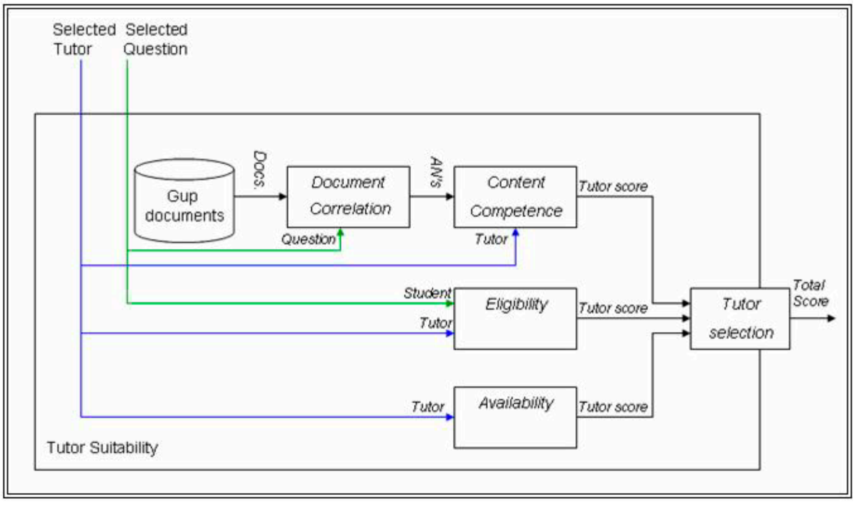


Figure 6.2 The design interface. 'Gup documents' to build the corpus and to set the LSA parameters; 'Document correlation' to set the document matching method; 'Content competency', 'Eligibility', 'Availability' and 'Tutor selection' to set the peer selection parameters.

The model covers three phases. In the design-phase, its operating context is defined. All texts of the Learning Network are captured and put into a corpus for processing; also, all parameters, the LSA and the peer selection parameters, are set. The designer has a specific interface (Figure 6.2) at his disposition to create the corpus and to experiment with and set the LSA and peer selection parameters (see the next phase for details). In principle, the design phase is operative only when launching and updating the Learning Network.

The question-phase starts whenever a student poses a question (e.g. “when I register for a particular chat room, does my registration allow me to use several pseudonyms?”). First, GTP (General Text Parser; Giles *et al.*, 2001), an LSA implementation, maps the question into the collection of text fragments in the corpus. The GTP module returns correlations between the question and text fragments. Text fragments always come from a unique Activity Node. Hence, a high correlation between the question and some text fragment also implies a high correlation between the question and a specific Activity Node. Thus having identified the Activity Nodes that are relevant for the question, ATL (A Tutor Locator; De Jong *et al.*, 2007) selects 2 peer-students most suited to give an answer. The selection is based on their competency on those Activity Nodes that GTP found to correlate highest with the question. This is called the peer’s *content competency*. Actually, ATL uses a weighted sum of four criteria. Besides content competency, it brings into the mix *tutor competency*, *availability* and *eligibility* (Van Rosmalen *et al.*, 2008).

Finally, in the answer phase a wiki is created that includes the question and the three text fragments that in the question phase GTP identified as correlating best with the question (Figure 6.3). The wiki is populated with the 2 peer-students who have been invited and have agreed to help. Peers and the question-asking student discuss possible answers and, hopefully, arrive at a satisfactory one. Parenthetically, we chose to work with three documents as a compromise between too few to be helpful and too many to be all read by the supporting peers.

Figure 6.3 Part of the WIKI: (1) "Vraag" (the question); (2) Three hyperlinks to the selected text fragments 'tekst 1', 'tekst 2' and 'tekst 3' (3) wiki-interface Toon (Show), Bewerk (Edit), Links (Links) and Geschiedenis (History).

Calibration and a first simulation

To ensure the model's viability, we calibrated the LSA-parameters, and simulated and tested two of its key aspects. First, we checked how good LSA was at identifying the topic of a question (i.e. to which Activity Nodes a question belongs) and at selecting text fragments useful for answering the question. Second, we checked if the peer selection criteria met our expectations. The domain of the Learning Network we used is 'Internet Basics', a collection of texts, links and tasks that aim to instigate a basic understanding of the Internet (Janssen *et al.*, 2007). It contains 11 Activity Nodes (content modules), each of which introduces a different aspect of the Internet. The Activity Nodes consist of an introduction, exercises, references to external web pages for further study, and an assessment.

For the simulation, we formulated a set of 16 test questions, each related to exactly one Activity Node. The prototype identified the correct Activity Node for 12 out of the 16 questions (75%). Moreover, two of the authors of the Learning Network in question evaluated the fit of the 3 text fragments proposed by the system. They indicated that for 7 of the questions, one or more of the text fragments were indeed useful in answering those questions. The authors also indicated that 5 of the 16 questions posed were beyond the scope of the contents of the Activity Nodes studied. Taking this into account, the score is 7 questions with useful text fragments out of a total of 11 (about 60%, for details, see Van Rosmalen, 2006).

To test the validity of the peer selection criteria, we created 5 student accounts (Table 6.2) and assigned a set of test values to the parameters of the peer selection formula. Here we will only illustrate how they worked, for details see Van Rosmalen *et al.* (2008). By way of test, student L_1 twice 'asked' one of the 16 questions mentioned above. The question was related to Activity Node 2. The first time the student asked the question, the peer-student with the highest rank was selected, as expected. Furthermore, the results of the test showed, that we could balance the selection of peers by taking availability and eligibility into account. For the first question the value of eligibility favoured student L_2 over student L_3, i.e., it prioritized the selection of a peer-student in the same study-phase. (Note: L_2 and L_3 have content competency 1 and availability 0.5. However, only L_2 and L_1 finished Activity Node 1, therefore L_2 has a higher eligibility than L_1). However, if we pose the same question again a compensation mechanism becomes operative due to the decreased availability of Student L_2. (Note: Because of his being chosen to answer the first question the availability of L_2 will become 0).

Table 6.2 Content competency of student L_1 - L_5 for Activity Node 1 and Activity Node 2, and their availability score.

	L_1	L_2	L_3	L_4	L_5
Content competency Activity Node 1	1 (= successfully completed)	1	0.3	0 (= not started)	0
Content competency Activity Node 2	0.3 (= in progress)	1	1	0	0
Availability (at the start)	0.5 (= moderately available)	0.5	0.5	0.5	0.5

Materials and methods

Having designed and developed a model that fitted our goals and having calibrated it in order for it to function optimally, we proceeded to carry out an experiment. To that end, we made available for 8 weeks a course in the Learning Network on Internet Basics, the same course that was used to calibrate the model. The course is a free (no tuition) course. No credits were given for its completion. The number of study hours is about 22. 111 Students were recruited from both students and staff in our organisation. One of them withdrew after one week. All 110 remaining students were treated as novices when starting the course. For the experiment, the students were divided at random in two groups. Looking at the group characteristics in detail, the groups were very similar with regard to age, gender, computer experience and previous education. Computer experience varied from having little (38) to much (42) and only few students having very little (3) or very much (4) experience (23 non-responders). Males and females were present in almost equal numbers (46 versus 49, 15 non-responders). The modal age was between 46 and 55 (32), trailing off to 25 between the ages of 56 and 65, and 18 between the ages of 36 and 45, with 12 under 35 (5 of which under 25) and 5 over 65 (18 non-responders). Participants educated at the tertiary level (58) vastly outnumbered participants having had up to secondary education (25) or primary education only (3) (24 non-responders or differently educated).

In the experimental group, we used the following weights for the parameters: 1.0 *content competency*, 0.5 *availability* and 0.5 *eligibility*. The model prescribes that the content competency should always be positive. Therefore to avoid a “cold-start” problem, i.e. having no suitable students at the beginning, if necessary the system automatically switched to the settings of the control group. In the control group, we only ensured that the questions would be divided evenly between the students. This meant 1.0 *availability* and all other weights zero (for a detailed description of the parameters and criteria see Van Rosmalen *et al.*, 2008). For each question we invited the two students with the highest scores. If within 48 hours no one responded the two next best students were invited.

All students received general instructions on the use of the Learning Network and a specific instruction on how to use the AskCQ-module for all their content-related questions. They received a global explanation of the objectives of the experiment and the suggestion to use the AskCQ-module at least twice, if appropriate. In addition, three times during the 8 weeks the Learning Network was running, a newsletter was distributed with information regarding the course and also a notice on the AskCQ-module. Staff-tutors did not assist answering content-related questions at all. Their role was limited to rating the results of each question-answer pair after completion of the experiment.

Finally, of the total of 110 students, 78 students were active: 40 in the experimental group and 38 in the control group; the remaining 32 students showed no or very limited activity. This means they did not complete any of the 11 Activity Nodes. The average numbers of Activity Nodes completed was for each group 6. The total number of students that successfully completed the course was 25 for the experimental group and 24 for the control group.

HYPOTHESES

We tested three related hypotheses:

- The model helps to solve a substantial number of the content-related questions posed by students, without invoking any staff support (hypothesis *A*). A substantial number is about 50%, as in our view this is the minimum percentage sufficient to justify the investment needed to develop this kind of system. In specific cases though, e.g. if discussions between students have a very high priority, the percentage may of course be set lower.
- Ad hoc transient communities for which peers were selected on the basis of a combination of the four criteria proposed (i.e. including the LSA-based content competency) outperform communities whose peers were selected with workload balancing in mind only (hypothesis *B*). This hypothesis concerns the effectiveness of the LSA-based peer selection system.
- Text fragments with which the wikis were seeded helped the peer-tutors to answer the questions asked by their peers (hypothesis *C*). This hypothesis is about the usefulness of seeding the wikis with text fragments identified with LSA.

Hypothesis A is tested directly, simply by counting the number of questions solved successfully. Whether a question is solved successfully, is assessed by the student who posed the question and by two expert tutors. Hypothesis *B* has been tested by administering different peer-student selection approaches to an experimental and a control group of students; group membership was decided by random selection. Peer tutors in the experimental group were selected on the basis of content competency, determined with LSA, availability and eligibility; peer tutors in the control group were selected on the basis of availability only. Dependent variables measured are the number of invitations accepted, the time to answer a question, the number of questions answered, and the quality of the answers given. Hypothesis C has been tested in a limited sense only. The wikis of both the experimental and control group were seeded with text fragments in order to avoid compounding the effect of the way in which the groups have been composed with the availability of text fragments. Therefore, the effect of the availability of text fragments on, say, the quality of the answers given could not be tested. In order still to assess their usefulness we looked at the sources the students used to answer the question (did they use the fragments at all?) and at the extent to which they perceived the text fragments as useful.

DATA COLLECTION

We collected four types of data. The former two, logging data and student ratings, have been collected during the experiment; the latter two, staff-tutor ratings and evaluation data, have been collected after the experiment:

- Logging data. The progress data of the students, i.e. for each question: question time; invitation-accepted time; number of invitations accepted; answer-accepted time; value of the tutor suitability and the value of the underlying criteria; the main Activity Node of the question.
- Student ratings. For each question, the students that accept the invitation each rate their own peer-tutor suitability; the question poser indicates the main Activity Node of the question and rates the answer received.
- Staff-tutor ratings. At the end of the course, two staff-tutors rated the answers of all closed questions-answer pairs; for these pairs they also specified the Activity Node from which the question derived in their opinion.
- Evaluation data. At the end of the course the students received a questionnaire on the usability aspects of the system. The questionnaire included among other things questions on the usefulness of the supplied text fragments; on how students supplied their answers (prior knowledge, an Activity Node or another source); on whether students perceived other positive outputs such as getting to know each other or understanding the learning material better; and on whether students appreciated the overall approach and use of the system.

Results

HYPOTHESIS A: QUESTIONS SOLVED

During the 8 weeks of the course, a total of 101 questions were posed, 59 in the experimental group and 42 in the control group (see Table 6.4, rows 1a-1c for more details). According to the question posers, the number of successfully answered questions was 42 (71%) for the experimental group and 19 (45%) for the control group (Table 6.3, row 1). However, these figures do not suffice to conclude that hypothesis A has been confirmed.

First of all, the student judgments should be in line with the staff-tutors' rating. After all, the students could have been satisfied too easily. The two staff-tutors rated all questions, including the questions started but not yet rated by the students (Table 6.3, rows 2-4). The overall agreement between the tutors on solved versus not-solved questions (i.e. "solved" = rating 4 and 5 and "not solved" = rating 1 and 2) is high: 83% (62 out of 75) or 73% (64 out of 88 if we also include the rating 3 "solved/not solved"). If we combine the judgment of the students and the tutors, by counting a question as solved if at least two of the three ratings are 4 or above, the number of questions solved is approximately

the same as the number indicated by the students (Table 6.3, row 5). So student opinion do not differ much from expert (staff) opinions.
Second, irrespective of an overall agreement between students and staff, there should only be very few 'false-positives'. A false-positive is an answer that according to the student is right but actually is wrong. Too many false-positives are a threat to the quality of education. Based on the ratings, we identified 8 questions that required further analysis. Careful reading of each of the questions showed that none of the answers was a genuine false-positive:

- 5 questions were irrelevant and/or closed by the question poser before the peers could help.
- 2 answers received a rating of 3 by both staff-tutors. In both cases, however, one could easily argue that the question was fairly well answered.
- Finally, one remaining question was not articulated well, making it difficult to judge whether the answer was adequate or not.

Table 6.3 Question-answer details.

		Experimental	Rated	Control	Rated
1	Questions solved: Question poser	42 (42/59=71%)	53	19 (19/42=45%)	29
2	Questions solved: Tutor 1	34	53	16	29
3	Questions solved: Tutor 2	38	53	17	29
4	Question* not closed but solved according to Tutor 1 and 2	2	4	4	6
5	Questions solved (integrated score) at least 2 agree	44 (44/59=75%)	57	22 ((22/42=52%)	35

** Note: 6 of the 10 questions started, but not rated, did contain an answer, 4 were not really started.*

HYPOTHESIS B: THE MODEL TREATMENT

For hypothesis B we looked if the experimental group outperformed the control group on responsiveness (the number of invitations required and the answer time) and quality (the number of solved questions, the level of the answer ratings) of the answers.

Before going into details, a first inspection suggests that the students in the experimental group participated more actively in question-answering. They posed more questions (though not significantly), solved more questions (see above), answered faster, and fewer invitations were required (Table 6.4, rows 1, 6 and 2 respectively). Their answers were also rated higher (Table 6.4, row 7). There was only one apparent anomaly, the control group (Table 6.4, rows 4-5) had a higher overall involvement. This, however, actually is to be expected as in the control group the algorithm attempts to achieve an optimal spread of the workload.

Table 6.4 Overview of the results.

		Experimental	Control	Total
1	The number of questions posed*	59	42	101
1a	- closed (so rated);	53	29	82
1b	- in discussion	4	6	10
1c	- failed i.e. the invited peer-tutors did not react to or refused the invitation	2	7	9
2	The number of questions with 2 invitations loop	12	21	33
3	The number of students that posed one or more questions	26	21	47
4	The number of students that assisted in answering one or more questions	29	36	65
5	The total number of students actively involved (posing or answering)	31	37	68
6	Average time to resolve a question (days)	5.6	9.6	7.0
7	Average answer rating of the rated questions (5-points scale, 1 is not answered - 5 is fully answered) by the question poser	4.0	3.4	3.8

** For 4 out of the 9 questions not started, invitations were sent out on the very last day that the course was available.*

The following tables show the results of a detailed analysis on responsiveness, i.e. the number of invitations required (derived from Table 6.4, row 2) and the answer time (Table 6.4, row 6). For both the number of invitations required (Table 6.5) and the time to answer (Table 6.6) the experimental group scores significantly better.

Table 6.5 Number of invited students per question: 2 implies one invitation loop, 4 two loops.
The chi-square = 9.81; df = 1; asymp sig = 0.002.

		Experimental	Control	Total
Number of invited students	2	47 (79.7%)	21 (50.0%)	68 (67.3%)
	4	12 (20.3%)	21 (50.0%)	33 (32.7%)
Total		59 (100.0%)	42 (100.0%)	101 (100.0%)

Table 6.6 Answer time (in hours).
Mann-Whitney U = 436.500; asymp. sig. = 0.001.

	Group	N	Mean Rank	Sum of Ranks
Answer time in hours	Experimental	53	35.24	1867.50
	Control	29	52.95	1535.50
	Total	82		

The picture is confirmed when looking in detail at the quality, i.e. the number of question solved (Table 6.7). We already noted that the experimental group solved more questions, further analysis shows that difference to be significant.

Moreover, when looking in detail at the rating of the questions, the two groups not just differ with respect to questions solved and not-solved (Table 6.7). The experimental group, also proves to be very clear about its judgement (Table 6.8). A large part of their ratings (60%) is in the segment of absolutely reject (rating 1) and absolutely accept (rating 5).

Table 6.7 Solved Questions. The chi-square = 6.90; df = 1; asymp sig = 0.009.

		Experimental	Control	Total
Questions	Not solved	17 (28.8%)	23 (54.8%)	40 (39.6%)
	Solved	42 (71.2%)	19 (45.2%)	61 (60.4%)
Total		59 (100%)	42 (100%)	101 (100%)

Table 6.8 Ratings of the answers.

		Experimental	Control	Total
Rating Question Poser	1	6 (11.3%)	6 (20.7%)	12 (14.6%)
	2	5 (9.4%)	4 (13.8%)	9 (11.0%)
	4	16 (30.2%)	10 (34.5%)	26 (31.7%)
	5	26 (49.1%)	9 (31.0%)	35 (42.7%)
Total		53 (100.0%)	29 (100.0%)	82 (100.0%)

So there is strong evidence that the experimental group outperforms the control group. However, it still needs to be shown that the selection algorithm chooses its peer tutors for the reasons we assume. A critical condition for this to be the case is that the question posed should be mapped to the relevant Activity Node (as the result of this mapping is used to determine the suitable peer-students). The staff-tutors were asked to indicate the main Activity Node for 71 questions (9 of the questions were not rated at all, because nobody had yet responded to them, 20 questions were rated as being out of scope with regard to the content of the Learning Network). The results for both tutors were almost identical. They differed in opinion only on 4 questions. In 62% of the cases the judgment of the tutors was identical to the one calculated by the system. This result imparts confidence on the algorithm, particularly if one takes into account that in practice it points to more than one Activity Node.

HYPOTHESIS C: SUPPORTING THE STUDENT IN ANSWERING

To establish to what extent the text fragments are of use in answering the questions, we collected the following results through the questionnaire. First of all, the majority of the respondents (34 of 50) valued the text fragments with a rating of 4 or above (Table 6.10). The resources used to answer are given in Table 6.9.

Table 6.9 The resources used to answer a question, the respondents could select one or more choice (n = 51; 1 skipped this question, 5 were not involved as peer-tutor).

		Experimental	Control	Total
Resources used	Text fragments	5	9	14
	Course content	14	11	25
	Prior-knowledge	17	12	29
	Others (unspecified)	11	9	20

Table 6.10 The appreciation of the text fragments (n = 50; 2 skipped this question, 5 were not involved as peer-tutor).

		Experimental	Control	Total
Text fragment rating (1=useless ; 5=good)	2	1	4	5
	3	10	1	11
	4	11	12	23
	5	3	8	11
Total		25	25	50

It is interesting to notice that the control group tends to appreciate the text fragments more; also they mention more often that they actually used the text fragments (both differences not significant). An explanation could be that the control group is more dependent on the text fragments since they have been selected at random. Also there is the in-built tension that a good selection strategy likely diminishes the need for text fragments.

GENERAL OBSERVATIONS

Finally, through the questionnaire we received feedback on the usability and general acceptance of the model. The questionnaire was completed by 57 of the 110 students (52%) fairly evenly divided over the two groups, 29 (experimental) and 28 (control) respondents respectively. For both groups, almost all respondents agreed that answering a question is a good investment of time (25 experimental; 22 control). Positive responders could motivate their answer by indicating one or more reasons on a list. Two reasons were selected most often:

- “I am aware that other students also have questions” (24 students), and
- “It improved my knowledge and understanding” (29 students).

The overall usefulness (26 experimental; 17 control) and usability (22 experimental; 16 control) received a positive rating i.e. 4 or above on a 5-point scale. The figures, however, show that the respondents of the experimental group are more positive. This was confirmed when asked who would like to see this question-answering approach offered in other courses too. The students in the experimental group were significantly more interested, i.e. 25 students of the experimental group answered positive against 16 of the control group (chi-square = 5.177; df = 1; asymp sig = 0.023).

Discussion and conclusions

The tests of our three hypotheses lead to the conclusion that our system can successfully be used to help answering content questions students may have. For the experimental group the number of questions solved is clearly above 50% and there were no false-positives. The number of solved questions could have even been higher, had we not abruptly stopped the experiment after 8 weeks. We also unambiguously showed that the experimental group outperforms the control group both with regard to responsiveness and quality. Only the result for the last hypothesis, about the usefulness of the text fragments, is less clear. Though students seem to appreciate the text fragments, they make only limited use of them.

Despite these promising results, a number of limitations to the experiment have to be considered. The experiment ran with a fixed group and for a fixed period. This situation is different from our target situation, with an ‘unending’ course and lifelong learners starting and finishing at any time. Another concern is the limited complexity of the contents studied, which were typically at the beginner level. This was confirmed indirectly as a substantial part of the responders indicated that they used prior knowledge to answer the questions. Results may be different with more demanding topics, the questions may then be too difficult to answer. Finally, students were aware that they participated in an experiment. This likely will have boosted their responsiveness. In the reality of a long-lasting Learning Network with many Activity Nodes, policies to ensure sufficient participation may be required (Berlanga *et al*., in press).

Also the particular implementation of the model deserves further attention. The wiki proved to be a tool that was unknown to the students. They used it much in the same way as a forum, contributing in turn and not editing the texts of others, while we had hoped them to become involved in a collaborative writing process. The students also ‘complained’ that they were not properly informed once a question was resolved and, even more important, that it was not possible to continue a contact through the system. Both we consider very serious issues because an important additional objective of this system is to assist lifelong learners in becoming (self-)organised into communities (Kester *et al*., 2007). Therefore it is important that contacts are well-established and can be followed up if desired. Finally, students mentioned they would have liked to be able to study all question-answer pairs, not only the ones they contributed to. Whether or not this is a good idea is not obvious. It may improve the question-answering efficiency but it will likely lessen the need for and the benefits of a discussion between students.

Further research is needed to address the issues raised and also to get a better insight into the effects of different values for the parameter settings of the model. So far, however, we believe that we showed that the system developed offers a promising line for efficient and effective support for e-learning in general and lifelong learning in particular.

Acknowledgements

The authors' efforts were partly funded by the European Commission in TENCompetence (IST-2004-02787) (http://www.tencompetence.org). The authors thank André de Jong, who built the current release of the prototype, Tally Schmeits, who assisted in the final debugging and the queries to collect the data, and José Janssen for her valuable help with the analysis of the data.

Chapter 7

General Discussion

Introduction

Nowadays, continuous development of knowledge and skills is a key requirement for all professionals no matter their profession or position. The European Commission expressed this in one of its latest funding calls (FP7-ICT-2007-1, 2006) in the following words: “In today’s society individuals and organisations are confronted with an ever growing load and diversity of information and content, and with increasing demands for knowledge and skills”. One of the proposed solutions called for “responsive environments for technology-enhanced learning that motivate, engage and inspire learners, and which can be embedded in the business processes and human resource management systems of organisations.” The advent of the knowledge economy and the individual demands of lifelong learners are two factors that underpin an increasing demand for flexibility: students want to be able to study at the time, place and pace of their own choosing; furthermore, students are reluctant to submit themselves to pre-planned, rigid programmes, but want their prior competences honoured and their specific study plans catered for. Nevertheless, as in traditional settings, students will have questions on where to start, how to proceed, how to understand and apply the available study material, or they will want to have their contributions assessed.

In this thesis we made an attempt to address these challenges. In the first part, we discussed a model aimed at meeting the increasing demands of learners to have a wide variety of learning activities at their disposal which, in line with contemporary pedagogical models, adapt to their individual needs. We investigated how to support staff in designing these. In the second part, we discussed a model that aimed at supporting staff in supervising and giving guidance to students using these learning activities. In this chapter, we will discuss the results of our work and reflect on our findings. We will start with an overview and discussion of the results. This shall lead us to point out the scope and limitations of our work, practical implications and an agenda for further research.

Review of the results

Authoring Adaptation

Our first topic in this thesis was an exploration on how to support the design role of staff in adaptive systems. Chapter 2 set the stage for our model. We introduced the aim of our system and the requirements behind it, i.e. it should: (1) support active and adaptive e-learning; (2) be open with regard to different types of learning as well as to new components, e.g. agents; (3) support the user in an efficient way. To meet these criteria we argued for a combination of standards and an open architecture with dedicated agents. After having presented an overview of tools, technologies and methods relevant to our aim, we presented our framework. It builds on a set of five e-learning standards (the main one being IMS-LD) to assure the required openness, but it also heavily relies on agents to enable part of its functionality. It distinguishes itself in that it integrates and supports two approaches towards adaptation to the learner's needs, i.e. design time and runtime adaptation. At design time an author can inspect, adapt or create a learning design and use it in multiple courses. At runtime a tutor or agent can interpret the learning design and students' progress and subsequently take appropriate adjustments, e.g. make suggestions to learners, while a course is in progress. The viability of the framework was demonstrated through a small mock-up with Edubox (an EML-based e-learning environment; EML being the predecessor of IMS-LD) with two agents connected to it.

Chapter 3 focused on the question that is central to the first part of this thesis, i.e. using the framework proposed, how to support authors in their design of adaptive e-learning. Authors have two options to create adaptive e-learning. They can specify the adaptations required directly in IMS-LD or they can make use of the adaptation facilities of the agents and specify the data required by them. For authors to be able to use the system there are a number of conditions that need to be fulfilled. The two most important ones are that, firstly, the available adaptation options have to be transparent to the authors, i.e. it has to be clear to them what these options do (and why) and what is required to use them. Secondly, authoring should be facilitated with tools and guidelines that enable authors to take full advantage of these options. To cope with these requirements, the authors received a combination of tools and documentation including a description of the aLFanet life cycle model for adaptation with explanations of the adaptation features on offer, a 'concept learning' template, and IMS-LD and IMS-QTI authoring tools and manuals.

Our hypothesis was that this standards-based framework and the combination of tools and documentation, indeed facilitated and simplified the authors' task of designing adaptive e-learning. The evaluation followed a design research approach. In three successive rounds, a group of authors was observed while using the system. The results of each round were used to feed into the next development stages of the system. The results acquired through the evaluation were mixed. The overall opinion of the authors was one of dissatisfaction; they complained that the tools were too complex to work with and that it was too difficult to make full use of the adaptivity offered. Nevertheless, there were also a number of positive outcomes. The use of the system showed that it was possible to support adaptive e-learning by using standards and a set of services. The tools for editing IMS-LD and IMS-QTI were a (be it small) improvement on editing the underlying XML directly. The authors appreciated the use of a template, the guidelines and the adaptation features, even if they did not meet the required level of usability.

Facilitating Support
Our second study was a detailed elaboration of a model that intends to address one particular example of a typically time-consuming task of staff, i.e. answering content related questions. In Chapter 4 we defined four requirements for our model:

- It had to alleviate the support task for the tutor.
- It had to involve a substantial part of the student population.
- It had to be able to support the students in their task.
- It had to be portable.

Based on a cursory search for ways to answer content related questions we opted for the unique combination of setting up a small group of selected peers to give the answer with the help of documents selected from the course. We used LSA to identify the most relevant text fragments and the Activity Nodes (topics) related to a question. The selection of the peer students providing support is based on a weighted sum of four criteria:

- *content competency,* indicating if a learner has successfully finished the Activity Node(s) relevant for the question;
- *tutor competency,* indicating past performance in the role of peer-tutor;
- *availability*, a combination of actually being available and how many questions this peer has previously answered (workload balance);
- *eligibility*, a similarity measure relating the progress of the peer-student with the student posing the question.

In Chapter 5 we focussed on a key aspect of our model, i.e. the use of LSA. We prepared a corpus from an existing Learning Network on ‘Internet Basics’ and calibrated the LSA-parameters. We determined the value of the parameters with the help of 16 questions selected from the assessment section of the Activity Nodes of the Learning Network. In the optimal combination of parameters, LSA correctly identified 15 out of 16 questions. We double-checked by formulating 16 new questions. This time LSA identified 12 questions correctly. In addition, the designers of the Learning Network rated the selected text fragments. They indicated that 5 questions were beyond the scope of the Learning Network, and that LSA suggested useful text fragments for 7 out of the 11 questions left.

In Chapter 6 we described and discussed the results of a field test with a prototype of the model. We set up a course in the Learning Network on Internet

Basics, the same course that was used to calibrate the model. The course is a free of charge (no tuition) course that lasted 8 weeks. Students were invited from among students and staff of our organisation. They were divided at random into two groups. In the experimental group, we used the model to select the students to help answering a question. In the control group, we only made sure that the questions would be allotted evenly to the students. In the experiment we tested three related hypotheses:

A. The model solves at least 50% of the content-related questions posed by students, without invoking any staff support.
B. The groups that are composed by the model outperform groups whose members have been selected with workload balancing in mind only.
C. The text fragments selected by the system help peers to answer the questions posed to them.

To verify our hypotheses we looked at a combination of logging data, student ratings, staff ratings, and data from a questionnaire. We obtained the following results:

A. In the experimental group 75% of the questions were solved, against 52% in the control group.
B. The experimental group significantly outperformed the control group. The number of invited students per question was significantly lower; the answer time was significantly lower; and the number of questions solved was significantly larger.
C. The data for the last hypothesis were more ambiguous. A big majority of the respondents valued the text fragments although their actual usage was limited.

Finally, it is important to note that the general response to the system was positive. Almost all respondents to the questionnaire agreed that the time it takes to answer a peer's question is time well spent.

The scope and limitations of the research

The two models proposed have a different scope and therefore different limitations. Our first study was an *exploratory* study, in the relatively unpaved territory of adaptivity and standards, on how to support the design role of staff in adaptive systems. Our second study was a *detailed* exploration on a model that intends to alleviate one particular example of a time-consuming task of staff, i.e. answering content related questions.

ALFanet is (one of) the first e-learning environments developed on a set of five e-learning standards to provide adaptation during the full life cycle of the e-learning process. Being one of the first to explore the combination of five standards (including the then brand-new IMS-LD specification) within the context of an adaptive system obviously gave rise to a lot of unexpected challenges. As a result a large number of research questions became heavily intertwined. The project had to deal with technical issues, i.e. standards not 'prepared' to work with other standards; functional questions, i.e. how to apply these standards within the functionality required; and usability questions, i.e. how to enable designers, tutors and learners to make the most effective use of the tools while at the same time guaranteeing a system committed to a complex set of standards and a variety of adaptive learning scenarios. As a result, the study we undertook has clear limitations. Two general challenges were met fairly well: the standards were integrated and the system did offer a set of adaptive features. However, on the challenge most important for our study, i.e. the usability of the authoring process and tools, the verdict is still out and there is room for significant improvement. The expertise required to operate the current tools is not commonly available and is not likely to emerge on a large enough scale. The use of a template and a catalogue of adaptive scenarios were judged as useful by the authors but not sufficiently translated into the tools. To assure further uptake, future research and development should focus on how to clearly articulate the design choices and to translate the constraints and requirements imposed by these choices directly into the authoring tools to minimize complexity and to take advantage of information that can be derived automatically.

The question-answering model studied here, had a clear and relatively restricted focus. Therefore its limitations are much more tangible than for the first model. While the results are clear and point into the direction of a useful and valuable approach, there are a number of concerns. The first and most obvious limitation of the setup lies in the choices made. As we have seen in the literature review there are many options to address the problem of answering questions. What option prevails depends very much on the role and responsibility we perceived for the learner and to what extent we wanted to rely on various technologies. The other limitations are related to the characteristics of the experiment we carried out, its fixed group and fixed duration. The experimental situation deviates from our target setting, with lifelong learners starting and finishing at any time of their liking. Also we abstained from tutor intervention and installed no specific supporting policies. Abstracting away from the experiment, it may be too optimistic to rely on the self-regulatory powers of students and their willingness to invest time in others without tangible rewards. There should be an exit strategy in case of problems (e.g. un-solved questions or inappropriate behaviour) and also some policies may be required to assure sufficient participation. Another concern is the complexity of the contents studied. The topics of our Learning Network were typically at the beginners' level. The results of the experiment may have been different with more demanding topics. Moreover, the corpus derived from the Learning Network was relatively small and this may have affected the performance of the LSA software. However, probably the most important limitation of the setup at this moment does not affect our conclusions, it points to a missed opportunity. We did not enable students to remain in contact with each other after their engagement in answering a question. An important, additional objective for the model then could be to assist lifelong learners in becoming (self-) organised in communities.

Practical implications

In the introduction we recalled that further development and deployment of technology enhanced learning has to face a number of obstacles. Firstly, in order to meet the increasing demands of learners, staff have to develop and plan a wide and complex variety of learning activities that, in line with contemporary pedagogical models, adapt to the learners' individual needs. Secondly, staff have only limited time to support learners. In this thesis, we discussed two models each addressing one of those needs. From the experiments done we can derive the following practical implications:

The first study made clear, at system level, that it is possible to build an adaptive, standards-based system. This should open up the possibility of development of services that are transferable to more widely-used systems. Secondly, the authors thought the use of a template useful. Templates can encapsulate design knowledge that is otherwise difficult to incorporate; they can facilitate the design of activities otherwise difficult to create. The combination of services and templates can alleviate the design task of staff. In both cases the underlying assumption is that the requirements to use the adaptive features are transparent and well translated into tools minimizing the complexity for the authors and making optimal use of context specific constraints. An example of this is for instance COLLAGE (Hernández-Leo *et al.*, 2006) where existing patterns are used to help authors in the process of creating their own (collaborative) Learning Designs.

For the second model one may firstly infer that in a structured process with a well-defined task the students are willing and competent to support each other. Moreover, they indicated that supporting each other was a good investment of their time. A selection mechanism that takes into account the competence, availability and eligibility of the participants further increased their willingness to participate and the quality of the answers. Besides, our specific case study invites us to look at other critical student support activities and to investigate whether they can be structured and supported following the same principles.

Secondly, at the technology side, the use of LSA appeared to be quite successful. It was possible with a modest investment, in terms of preparing a corpus and setting the parameters, to support the critical steps in the relatively complex task of selecting the right peer-students. This area too could benefit from further in-depth exploration. There are many tasks in technology enhanced learning that can take advantage of LSA or other language technologies ranging e.g. from the analysis of essays to the analysis of e-portfolios, interactions or content.

Further research

The studies reported in this thesis give rise to a number of questions for further investigation. Our first study showed the potential value of the combination of standards and adaptivity but, being of an exploratory kind, did not arrive at any final, transferable results. At the start of the project, key elements such as IMS-LD were still in their conception phase. However, meanwhile significant experience has been and still is gathered with the use of IMS-LD and other standards in projects such as TELCERT (http://www.open group.org/telcert/), UNFOLD (http://www.unfold-project.net:8085/UNFOLD), ELEGI (www.elegi.org) and TENCompetence (www.tencompetence.org). Since then, more user-friendly editors have emerged that take into account the background and knowledge authors may have (Sampson, Karampiperis & Zervas, 2005; Paquette et al., 2006) and make use of existing patterns (Hernández-Leo et al., 2006). Nevertheless, one important part of our ambition still requires further exploration. It is the combination of standards and adaptive services. In the aLFanet system we explored services such as the recommending of appropriate learning material based on the automatic identification of students with lack of knowledge or for extended learning for those with high interest levels (Boticario & Santos, 2007). This kind of services in combination with adaptations specified directly in IMS-LD can facilitate and dramatically simplify the authors' task of designing adaptive e-learning.

Our second study leaves open the question of how the results vary under different conditions. The following conditions may be relevant:

- Complexity of the domain. The responsiveness and the quality of the answers may be influenced by the complexity of the domain and therefore how difficult it is to answer correctly.
- Student population. The effect of having an 'ever-lasting' course with students starting and finishing at any time instead of having a fixed group for a fixed period.
- Policies. The effect of policies, e.g. showing high-rated peer-tutors; offering bonus study points or other rewards for active and high-rated peer-tutors;

limiting the right to ask questions depending on the number of answers given.

- Selection criteria. Both the weighting and the criteria themselves can be further explored. The eligibility criteria could for instance be extended or replaced by matching the learner's interest.

Another strand is to explore different kinds of tasks. At the moment we have concentrated on content-related questions. However, there are other types of questions that are relevant, e.g. advice on 'what to study' and on 'what topic to continue with' and other types of tasks such as supporting the writing process of essays.

Finally, there is the important issue of how to enable the formation of communities. The system as it is now creates temporary, small communities (ad hoc transient community). One may use such communities as instruments for letting students actively acquaint each other and in this way to motivate them and make it easier to continue the contact also in other situations. Thus lifelong learners may stop being lone-learners and become (self-) organised into communities (Kester *et al.*, 2007). The current implementation and also the short length of the experiment prohibited the students to continue their contacts. An extension of the current implementation and a prolonged study to see whether the initial contacts help to become self-organised, therefore is an important continuation of this research.

References

References

Anderson, T. (2004). Teaching in an Online Learning Context. In T. Anderson & F. Elloumi (Eds.), *Theory and Practice of Online Learning* (pp. 271-294). Athabasca: Athabasca University. Retrieved September 1, 2004, from http://cde.athabascau.ca/online_book/ch11.html.

Ayala, G. (2003). Towards lifelong learning environments: agents supporting the collaborative construction of knowledge in virtual communities. In B. Wasson, S. Ludvigsen & U. Hoppe (Eds.), *Designing for change in networked learning environments.* (pp. 141-149). Dordrecht: Kluwer Academic Publishers.

Bacsich, P., & Ash, C. (2000). Costing the Lifecycle of Networked Learning: Documenting the costs from conception to evaluation. *Association for Learning Technology Journal (ALT-J), 8*(1), 92-102.

Barrera, C., Boticario, J., Gaudioso, E., Santos, O., Rodriguez, A., Hoke, I., Boy, J., Schmidt, J., Fuentes, C., Carrion, J., Arana, C., Poelmans, P., Mofers, F., Passier, H., Moutinho, A., Barros, F., Canada de Abreu, R., & Ferreira, L. (2005). *D66 - Evaluation Results. Alfanet Project Deliverable,* Madrid 2005.

Bartolic-Zlomislic, S., & Bates, A. W. (1999). Assessing the costs and benefits of telelearning: a case study from the University of British Columbia. *Network of Centers of Excellence (NCE)-Telelearning Project report.* Retrieved November 8, 2005, from http://research.cstudies.ubc.ca/nce /index.html.

Berlanga, A., Sloep, P.B., Kester, L. Brouns, F., Van Rosmalen, P., & Koper, R. (in press). Ad Hoc Transient Communities: Towards Fostering Knowledge Sharing in Learning Networks. *International Journal Learning Technologies.*

Boticario, J. G., Gaudioso, E., & Hernandez, F. (2000). Adaptive Navigation Support and Adaptive Collaboration Support in WebDL. In *Lecture Notes in Computer Science (LNCS). Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-based Systems Trento, Italy, August 2000, 1892* (pp. 51-61). Berlin - Heidelberg: Springer Verlag.

Boticaro, J. G., & Santos, O. C. (2007). An open IMS-based user modelling approach for developing adaptive learning management systems. *Journal of Interactive Media in Education (*Adaptation and IMS Learning Design. Special Issue, ed. Daniel Burgos*), 2007/2*, from http://jime.open.ac.uk/ 2007/02.

Brouwers, M., Brouns, F., Van Rosmalen, P., Sloep, P.B., Kester, L., & Koper, R. (2006). ASA Tutor Locator. Retrieved 10 April, 2006, from http://sourceforge.net/projects/asa-atl.

Brusilovsky, P. (2001). Adaptive Hypermedia. *User Modelling and User-Adapted Interaction, 11*, 87-110.

Buchholz, S., & Daelemans, W. (2001). Complex answers: a case study using a www question answering system. *Natural Language Engineering, 7*(4), 301–323.

Caron, J. (1999). Design for the FAQ Organizer Application in Internet Dreams/CSCI 7000. Retrieved 14 July, 2004, from http://my.unidata.ucar.edu/content/staff/caron/collab/InternetDreams.pdf.

Caron, J. (2000). Applying LSA to Online Customer Support: A Trial Study. Retrieved 18 April, 2004, from http://my.unidata.ucar.edu/content/staff/caron/faqo/faqoPaper1.pdf.

Carrión, M. J., Fuentes, C., Rodrigo, M., Barrera, C., Catalina, C., Gaudioso, E., Boticario, J.G., Rodríguez, A., Santos, O., Martens, H., Mofers, F., Passier, H., Stoyanov, S., Vogten, H., & De Abreu, R.C. (2004). *General System Architecture & Design, ALFanet/ IST-2001-33288 Deliverable D21*. Madrid: SAGE.

De Croock, M., Mofers, F., Van Veen, M., Van Rosmalen, P., Brouns, F., Boticario, J., Barrera, C., Santos, O., Ayala, A., Gaudioso, E., Hernández, F., Arana, C., & Trueba, I. (2002). D12 - State-of-the-art. Alfanet Project Deliverable. Heerlen: Open Universiteit Nederland. Retrieved January, 2004, from http://hdl.handle.net/1820/94.

De Jong, A., Brouns, F., Van Rosmalen, P., Sloep, P., Kester, L., & Koper, R. (2006). GUP GTP Usability Prototype. Retrieved 10 April, 2006, from http://sourceforge.net/projects/gup.

De Jong, A., Brouwers, M., Brouns, F., Van Rosmalen, P., Kester, L., & Sloep, P.B. (2007). ATL: A Tutor Locator. Retrieved May, 2007, from http://hdl.handle.net/1820/960.

De Vries, F., Kester, L., Sloep, P., Van Rosmalen, P., Pannekeet, K., & Koper, R. (2005). Identification of critical time-consuming student support activities in e-learning. *Research in Learning Technology (ALT-J), 13*(3), 219-229, from http://dspace.ou.nl/bitstream/1820/397/4/CALT_A_137631.pdf.

Dumais, S. (2003). Data-driven approaches to information access. *Cognitive Science, 27*, 491–524

Ellis, A. L., Longmire, W.R., & Wagner, E.D. (1999). *Managing Web-Based Training: How to Keep Your Program on Track and Make It Successful*. Arlington: American Society for Training & Development.

Fantuzzo, J. W., Riggio, R. E., Conolly, S., & Dimeff, L. A. (1989). Effects of reciprocal peer tutoring on academic achievement and psychological adjustement: a component analysis. *Journal of Educational Psychology, 81*, 173-177.

Felder, R. M., & Silverman, L. K. (1988). Learning and teaching styles in engineering education. *Engineering education, 78*, 674-681.

Foltz, P. W., Laham, D. & Landauer, T. (1999). Automated essay scoring: applications to educational technology. *World Conference on Educational Multimedia, Hypermedia and Telecommunications, 1*, 939–944.

Foner, L. N. (1997). Yenta: A Multi-Agent, Referral-Based Matchmaking System. In W. L. Johnson (Ed.), *Proceedings of the First International Conference on Autonomous Agents in Marina del Rey, California* (pp. 301–307). New York, NY.: ACM Press.

FP7-ICT-2007-1. (2006). EU Call for Proposals - Information and Communication Technologies. Retrieved 10 January 2007, from http://cordis.europa.eu/fp7/dc/index.cfm?fuseaction=UserSite.CooperationDetailsCallPage&call_id=11.

Fuentes, C., Carrión, M., Arana, C., Boticario, J., Barrera, C., Van Rosmalen, P., Schmidt, J., Hoke, I., Barros, F., & Canada, R. (2005). *D8.2 - Public Final Report. Alfanet Project Deliverable*, Madrid 2005.

Gaston, J. (2003). Interactive Agents: responding to student queries “naturally”. *Presentation at EDUCAUSE 2003, Los Angeles.* Retrieved September 10, 2004, from http://mysiteagent.com/educause.

Giles, J. T., Wo, L., & Berry, M.W. (2001). GTP (General Text Parser) Software for Text Mining. Retrieved 10 January, 2005, from http://www.cs.utk.edu/~berry/papers02/GTPchap.pdf.

Graesser, A., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D. & Person, N. & the Tutoring Research Group. (2000). Using Latent Semantic Analysis to evaluate the contributions of students in AutoTutor. *Interactive Learning Environments, 8*, 149–169.

Griffiths, D., Blat, J., Garcia, R., Vogten, H., & Kwong, KL. (2005). Learning Design Tools. In R. Koper & C. Tattersall (Eds.), *Learning Design - A Handbook on Modelling and Delivering Networked Education and Training* (pp. 109-135). Berlin-Heidelberg: Springer-Verlag.

Guzdial, M. (1997). Information ecology of collaborations in educational settings: influence of tool. In *Proceedings of the CSCL’97*. Toronto, ON, Canada, 10-14 December.

Haley, D. T., Thomas, P., De Roeck, A. & Petre, M. (2005). A research taxonomy for latent semantic analysis-based educational applications. Technical report no. 2005/09. Retrieved 10 April, 2006, from http://computing-reports.open.ac.uk/index.php/content/download/187/1136/file/TR2005_09.pdf.

Hambrecht, & Co. (2000). Corporate eLearning: Exploring A New Frontier. Retrieved March, from http:// www.internettime.com/itimegroup/articles/hambrecht.pdf.

Hernández-Leo, D., Villasclaras-Fernández, E. D., Asensio-Pérez, J. I, Dimitriadis, Y., Jorrín-Abellán, I. M., Ruiz-Requies, I.,& Rubia-Avi, B. (2006). COLLAGE: A collaborative Learning Design editor based on patterns. *Educational Technology & Society, 9*(1), 58-71.

Howell, K. (2003). Question Generation and Answering Systems R&D for Technology-Enabled Learning Systems Research Roadmap. Retrieved November, 2004, from http://www.thelearningfederation.org/qa.pdf.

IMS-AccessForAll. (2004). IMS AccessForAll. Information Model, Best Practice and Implementation Guide. Version 1 Final Specification IMS Global Learning Consortium Inc. Retrieved August 30, 2005, from http://www.imsglobal.org/accessibility/index.html.

IMS-CP. (2003). IMS Content Packaging. Information Model, Best Practice and Implementation Guide. Version 1.1.3 Final Specification IMS Global Learning Consortium Inc. Retrieved May 26, 2004, from http://www.imsglobal.org/content/packaging/.

IMS-CP. (2004). IMS Content Packaging v1.1.4 IMS Global Learning Consortium Inc. Retrieved April 10, 2006, from http://www.imsglobal.org/content/packaging/index.html

IMS-LD. (2003). IMS Learning Design. Information Model, Best Practice and Implementation Guide. Version 1.0 Final Specification IMS Global Learning Consortium Inc. Retrieved May 26, 2004, from http://www.imsglobal.org/content/learningdesign/.

IMS-LIP. (2001). IMS Learner Information Package, Information Model, Best Practice and Implementation Guide. Version 1.0 Final Specification IMS Global Learning Consortium Inc. Retrieved January 22, 2004, from http://www.imsglobal.org/profiles/.

IMS-Metadata. (2001). IMS Meta-Data. Information Model, Best Practice and Implementation Guide. Version 1.2.1 Final Specification IMS Global Learning Consortium Inc. Retrieved January 22, 2004, from http://www.imsglobal.org/content/meta-data/.

IMS-QTI. (2003). Question and Test Interoperability. Information Model, Best Practice and Implementation Guide. Version 1.2.1 Final Specification IMS Global Learning Consortium Inc. Retrieved January 19, 2004, from http://www.imsglobal.org/qti/.

Janssen, J., Tattersall, C., Waterink, W.,Van den Berg, B., Van Es, R., Bolman, C., & Koper, R. (2007). Self-organising navigational support in lifelong learning: how predecessors can lead the way. *Computers & Education, 49*(3), 781-793.

Jennings, N. R., Sycara, K., & Wooldridge, M. (1998). A roadmap of agent research and development. *Autonomous Agents and Multi-Agent Systems, 1*, 7 – 38.

Kester, L., Sloep, P., Brouns, F., Van Rosmalen, P., De Vries, F., De Croock, M., & Koper, R. (2006). Enhancing Social Interaction and Spreading Tutor Responsibilities in Bottom-Up Organized Learning Networks. In P. Kommers, P. Isaías & A. Goikoetxea (Eds.), *Proceedings of the International Conference Web Based Communities 2006, 26-28 February 2006, San Sebastian, Spain* (pp. 472): IADIS.

Kester, L., Sloep, P., Van Rosmalen, P., Brouns, F., Koné, M., & Koper, R. (2007). Facilitating Community Building in Learning Networks Through Peer-Tutoring in Ad Hoc Transient Communities. *International Journal on Webbased Communities, 3*(2), 198-205.

Kester, L., Rosmalen, P. v., Sloep, P., Brouns, F., Koné, M., & Koper, R. (2007). Matchmaking in Learning Networks: Bringing Learners Together for Knowledge Sharing *Journal of Interactive Learning Environments, 15*(2), 117-126.

Kollock, P. (1998). Design Principles for Online Communities. *PC Update, 15*(5), 58-60.

Kollock, P., & Smith, M. (1996). Managing the Virtual Commons: Cooperation and Conflict in Computer Communities. In S. Herring (Ed.), *Computer-Mediated Communication: Linguistic, Social, and Cross-Cultural Perspectives* (pp. 109-128). Amsterdam: John Benjamins.

Koper, E. J. R. (2001). Modelling Units of Study from a Pedagogical Perspective: the Pedagogical Metamodel behind EML. Retrieved 2004, from http://eml.ou.nl/ introduction/docs/ped-metamodel.pdf.

Koper, E. J. R. (2003). Combining reusable learning resources and services to pedagogical purposeful units of learning. In A. Littlejohn (Ed.), *Reusing Online Resources: A Sustainable Approach to eLearning,* (pp. 46 - 59). London: Kogan Page.

Koper, E. J. R. (2004). Use of the Semantic Web to Solve Some Basic Problems in Education: Increase Flexible, Distributed Lifelong Learning, Decrease Teacher's Workload. *Journal of Interactive Media in Education. Special Issue on the Educational Semantic Web. ISSN:1365-893X, 6* (2004). Retrieved September 2004, from www-jime.open.ac.uk/2004/6.

Koper, E. J. R., & Sloep, P. B. (2002). *Learning Networks connecting people, organizations, autonomous agents and learning resources to establish the emergence of effective lifelong learning. RTD Programma into Learning Technologies 2003-2008. More is different.....* Heerlen: Open Universiteit Nederland.

Koper, E. J. R., & Tattersall, C. (2005). *Learning Design - A Handbook on Modelling and Delivering Networked Education and Training*. Berlin-Heidelberg: Springer-Verlag.

Koper, R., Giesbers, B., Van Rosmalen, P.,Sloep, P., Van Bruggen J., Tattersall, C., Vogten, H., & Brouns F. (2005). A design model for Lifelong Learning Networks. *Interactive Learning Environments, 13*(1-2), 71-92.

Koper, R. (2006). The TENCompetence Domain Model. *TENCompetence Project Report.* Retrieved 10 July, 2006, from http://dspace.ou.nl/handle/1820/649.

Landauer, T., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes, 25*, 259-284.

Leshin, C., Pollock, J., & Reigeluth, Ch. (1992). *Instructional Design Strategies and Tactics*. Englewood Cliffs, New Jersey: Educational Technology Publications.

Lin, F. O. (2005). *Designing Distributed Learning Environments with Intelligent Software Agents*. London: Information Science Publishing.

Meij, J. (2002). *Dealing with the Data Flood: Mining Data, Text and Multimedia*. The Hague: STT/Beweton.

Merrill, M.D. (2000). First principles of instruction. Paper presented at the *International Conference of the Association for Educational Communications and Technology (AECT)*, Denver, CO.

Michaelson, R. (1999). Web-based group work. In *Proceedings of the 10th Annual CTI-AFM Conference, August.* (pp. 58–64). East Anglia: CTI-AFM Publications.

Mizoguchi, R., Sinitsa, K., & Ikeda, M. (1996). Task Ontology Design for Intelligent Educational/Training Systems. *Position Paper for ITS '96 Workshop on Architectures and Methods for Designing Cost-Effective and Reusable ITSs, Montreal.* Retrieved 2004, from http://advlearn.lrdc.pitt.edu/its-arch/papers/mizoguchi.html.

Murray, T. (1999). Authoring Intelligent Tutoring Systems: An analysis of the state of the art. *International Journal of Artificial Intelligence in Education, 10*, 98-129.

Olivier, B. (2003). From EML to LD. *Presentation at CETIS Learning Design Workshop, Glasgow.* Retrieved October 2003, from http://www.cetis.ac.uk/members/pedagogy/files/LDworkshopOCT/ FromEMLtoLD_BO.

Oracle-Text-Reference. (2002). Release 9.2. Part Number A96518-86501. Dutch (nl) Default Stoplist. Retrieved 10 April, 2006, from http://www.lc.leidenuniv.nl/awcourse/oracle/text.920/a96518/astopsup.htm#37481.

Paiva, A. (1996). Communicating with Learner Modeling Agents. Position Paper for ITS '96 Workshop on Architectures and Methods for Designing Cost-Effective and Reusable ITSs, Montreal.

Paquette, G., Léonard, M., Lundgren-Cayrol, K., Mihaila, S., & Gareau, D. (2006). Learning Design based on Graphical Knowledge-Modelling. *Educational Technology & Society, 9*(1), 97-112.

Paramythis, A., Loidl - Reisinger, S., & Kepler, J. (2004). Adaptive Learning Environments and e-Learning Standards [Electronic Version]. *Electronic Journal of eLearning, EJEL*, 2 (1). Retrieved June 1, 2005 from http://www.ejel.org/volume-2/vol2-issue1/issue1-art11-paramythis.pdf.

Person, N. K., Gholson, B., Craig, S.D., Hu, X., Stewart, C.O., & Graesser A.C. (2001). HURAA: an interactive web-based agent that optimizes information retrieval in a multi-media environment. In C. Montgomerie & J. Viteli (Eds.), *Proceedings for ED-MEDIA 2001: World Conference on Educational Multimedia, Hypermedia and Telecommunications* (pp. 1476–1481). Norfolk, VA: Association for the Advancement of Computing in Education.

Preece, J., Nonneke, B., & Andrews, D. (2004). The top five reasons for lurking: Improving community experiences for everyone. *Computers in Human Behavior, 20*, 201-223.

Rawlings, A., Van Rosmalen, P., Koper, E.J.R., Rodríguez-Artacho, M.R., & Lefrere, P. (2002). Survey of Educational Modelling Languages (EMLs), Publication CEN/ISSS WS/Learning Technologies. Retrieved 2002, from http://www.cenorm.be /cenorm/businessdomains/businessdomains/informationsocietystandardizationsystem/elearning/learning+technologies+workshop/emlsurv eyv1.pdf.

Salmon, G. (2000). *E-Moderating: The Key to Teaching and Learning Online*. London: Kogan Page.

Sampson D., Karampiperis, P., & Zervas, P. (2005). ASK-LDT: A Web-Based Learning Scenarios Authoring Environment based on IMS Learning Design. *International Journal on Advanced Technology for Learning (ATL), 2*(4), 207-215.

Santos, O. C., Boticario, J.G., & Barrera, C. (2004). A Machine Learning Multi-Agent architecture to provide adaptation in a standard-based Learning Management System. *WSEAS Transactions on Information Science and Applications, 1*(1), 468-473.

Sloep, P. B. (2004). Reuse, portability and interoperability of learning content: Or why an educational modelling language. In R. McGreal (Ed.), *Online education using learning objects* (pp. 128-137). London: Routledge/Falmer.

Sloep, P. B., Kester, L., Brouns, F., Van Rosmalen, P., De Vries, F., De Croock, M., & Koper, R. (2007). Ad Hoc Transient Communities to Enhance Social Interaction and Spread Tutor Responsibilities. In V. Uskov (Ed.), *Proceedings of the Sixth International Conference on Web-based Education WBE 2007* (pp. 548-554). Chamonix, France: Acta Press.

Sloep, P. B., Hummel, H., & Manderveld, J. (2005). Basic design procedures for e-learning courses. In R. Koper & C. Tattersall (Eds.), *Learning Design - A Handbook on Modelling and Delivering Networked Education and Training* (pp. 139 -160). Heidelberg: Springer Verlag.

Tattersall, C., Vogten, H., & Hermans, H. (2005). The Edubox Learning Design Player. In R. Koper & C. Tattersall (Eds.), *Learning Design - A Handbook on Modelling and Delivering Networked Education and Training* (pp. 303-310). Berlin-Heidelberg: Springer-Verlag.

Towle, B. H., M. (2005). Designing Adaptive Learning Environments with Learning Design. In R. Koper & C. Tattersall (Eds.), *Learning Design - A Handbook on Modelling and Delivering Networked Education and Training* (pp. 215-226). Berlin-Heidelberg: Springer-Verlag.

Van Bruggen, J. (2001). *Computer Ondersteund beoordelen van essays (computer-assisted assessment of essays), OTEC Publication 2002/1*. Heerlen: Open Universiteit Nederland.

Van Bruggen, J., Sloep, P., Van Rosmalen, P., Brouns, F., Vogten, H., Koper, R., & Tattersall, C. (2004). Latent semantic analysis as a tool for learner positioning in learning networks for lifelong learning. *British Journal of Educational Technology, 35*, 729-738.

Van Bruggen, J. M., Rusman, E., Giesbers, B. & Koper, R. (submitted). Latent semantic analysis of small-scale corpora for positioning in learning networks. Retrieved 24 February, 2006, from http://hdl.handle.net/1820/561.

Van der Klink, M., Boon, J., Rusman, E., Rodrigo, M., Fuentes, C., Arana, C., Barrera, C., Hoke, I., & Franco, M. (2002). D72 - Initial Market Study, ALFanet/IST-2001-33288, Heerlen: Open Universiteit Nederland. Retrieved 2002, from http://learningnetworks.org/downloads/ alfanet-d72-initial-market-studies.pdf.

Van Es, R., Van der Baaren, J., Van Rosmalen, P., Manderveld, J., Koper, E.J.R., Boticario, J., Barrera, C., Rodriguez, A., & Santos, O. (2003). D31 - Existing Standards Analysis, ALFanet/ IST-2001-33288, Heerlen: Open Universiteit Nederland. Retrieved 2003, from http://learningnetworks .org/downloads/alfanet-d31-existing-standards-analysis.pdf.

Van Es, R., Vogten, H., Martens, H., Stoyanov, S., Van der Baaren, J., Barrera, C., Santos, O., Boticario, J., Hoke, I., Carion, M.J., & Fuentes, C. (2005). D32 - Standards Contribution Report, Alfanet Project Deliverable, Heerlen: Open Universiteit Nederland. Retrieved April, 2005, from http://hdl.handle.net/1820/345.

Van Es, R., & Koper, R. (2006). Testing the pedagogical expressiveness of LD. *Educational Technology & Society, 9*(1), 229-249.

Van Merriënboer, J. J. G. (1997). *Training Complex Cognitive Skills: A four-component Instructional Design Model for Technical Training*. Englewood Cliffs, NJ: Educational Technology Publications.

Van Rosmalen, P., Brouns, F., Tattersall, C., Vogten, H., Van Bruggen, J., Sloep, P., et al. (2005). Towards an open framework for adaptive, agent-supported e-learning. *International Journal of Continuing Engineering Education and Life-long Learning, 15*(3-6), 261-275.

Van Rosmalen, P., Sloep, P., Kester, L., Brouns, F., De Croock, M., Pannekeet, K. & Koper, R. (2008). A learner support model based on peer tutor selection. *Journal of Computer Assisted Learning*, 24, 74-86.

Van Rosmalen, P., Sloep, P., Brouns, F., Kester, L., Koné, M., & Koper, R. (2006). Knowledge matchmaking in Learning Networks: Alleviating the tutor load by mutually connecting learning network users. *British Journal of Educational Technology, 37*(6), 881-895.

Van Rosmalen, P., Vogten, H., Van Es, R., Passier, H., Poelmans, P. and Koper, R. (2006). Authoring a full life cycle model in standards-based, adaptive e-learning. *Educational Technology & Society*, Vol. 9(1), 72-83.

Vassileva, J., Detters, R., Geer, J., Maccalla, G., Bull, S., & Kettel L. (2001). Lessons from deploying I-Help. Workshop on Multi-Agent Architectures for Distributed Learning Environments. In *Proceedings of International Conference on AI and Education*. San Antonio, TX. 3 - 11 May.

Vivacqua, A., & Lieberman, H. (2000). Agents to assist in finding help. In T. Turner & G. Szwillus (Eds.), *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. New York, NY and The Hague, The Netherlands: ACM Press, from http://doi.acm.org/10.1145/332040.332408.

Vogten, H., Tattersall, C, Koper, R. Van Rosmalen, P., Brouns, F., Sloep, P., Van Bruggen, J., & Martens, H. (2006). Designing a learning design engine as a collection of finite state machines. *International Journal on E-Learning (IJEL), 5*(4).

Webber, C., Bergia, L., Pesty, S., & Balacheff, N. (2001). The Baghera project: a multi-agent architecture for human learning. Workshop - Multi-Agent Architectures for Distributed Learning Environments. In *Proceedings International Conference on AI and Education*. San Antonio, Texas.

Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems*. California: Morgan Kaufman Publishers.

Wild, F., Stahl, C., Stermsek, G. & Neumann, G. (2005). Parameters driving effectiveness of automated essay scoring with LSA. In *Proceedings of the 9th International Computer Assisted Assessment (CAA) Conference, Loughborough, UK, July 5, 2005* (pp. 485–494).

Wiley, D. (2004). Scalability and Sociability in Online Learning Environments. Retrieved 1 September, 2004, from http://wiley.ed.usu.edu/docs.

Wooldridge, M., & Jennings, N. (1995). Intelligent agents: theory and practice. *Knowledge Engineering Review, 10*(2), 115–152.

Woudstra, A., Huber, C., & Michalczuk, K. (2004). Call Centers in Distance Education. In T. Anderson & F. Elloumi (Eds.), *Theory and Practice of Online Learning,* (pp. 295-317). Athabasca: Athabasca University. Retrieved September 1, 2004, from www.cde.athabascau.ca/online_book.

Summary

Summary

The central research question of this thesis is how to support staff in the design and deployment of adaptive e-learning. The advent of the knowledge economy and the individual demands of each lifelong learner are two factors that underpin an increasing demand for flexibility in place, time and pace (logistic flexibility) and for learning activities that adapt their content to the learners' individual needs (subject matter flexibility). These increased demands for flexibility put a strain on staff, as it is they who have to deliver it to the students. Our research looks into two different ways of providing flexibility. The first - 'authoring adaptation' - investigates how to support staff in the design of adaptive learning activities. The second - 'facilitating support activities' - looks into how to support staff in supervising and giving guidance to students' learning activities.

Authoring adaptation
Our first study in this thesis is an exploratory study on how to support the design role of staff in developing adaptive e-learning. Chapter 2 sets the stage for our model. We introduce the aim of our system and the requirements behind it, i.e. it should: (1) support active and adaptive e-learning; (2) be open with regard to different types of learning models as well as to new components, e.g. agents; (3) support the user in an efficient way. A review of existing systems commonly used shows that they do not allow the developer to express content and design independently. They are therefore limited in the kind of e-learning they support and restricting the options to make the design process more efficiently. After providing an overview of tools, technologies and methods relevant to our aim, we present our framework. The most important aspects of our framework are that it builds on a set of five e-learning standards to assure the required openness and additionally that it heavily leans on agents (autonomous pieces of software) to enable part of its functionality. Furthermore, it distinguishes itself in that it integrates and supports two approaches towards adaptation to the learner's needs, i.e. both design time and runtime adaptation. The core standard used is IMS-Learning Design. IMS-LD offers a semantic notation to describe an educational scenario in a formal way. At design time a teacher or a design team can inspect, adapt or create a learning design model and use it in multiple courses. At runtime a tutor or agent can interpret a learning design and students' progress and subsequently take action, e.g. make suggestions to learners, while a course is in progress. The viability of the framework is demonstrated through a small mock-up with Edubox (an EML-based e-learning environment; EML being the predecessor of IMS-LD) with two agents connected to it.

Chapter 3 focuses on the question that is central to the first part of this thesis, i.e. using the framework proposed, how to support authors in their design of adaptive e-learning. First, we set the stage by describing the actual system (as it was built stepwise), its components and the type of adaptation they can offer. Authors have two options to make adaptive e-learning. They can specify the adaptations required directly in IMS-LD or they can make use of the adaptation facilities of the agents and specify the data required by them. For authors to be able to use the system there are a number of conditions that need to be fulfilled. The two most important ones are that, firstly, the available adaptation options have to be transparent to the authors, i.e. it has to be clear to them what these options do (and why) and what is required to use them. Secondly, authoring should be facilitated with tools and guidelines that enable authors to take full advantage of these options. To cope with these requirements, the authors received a combination of tools and documentation including a description of the aLFanet life cycle model for adaptation (figure 9.1) with explanations of the adaptation features offered, a 'concept learning' template, and IMS-LD and IMS-QTI authoring tools and manuals.

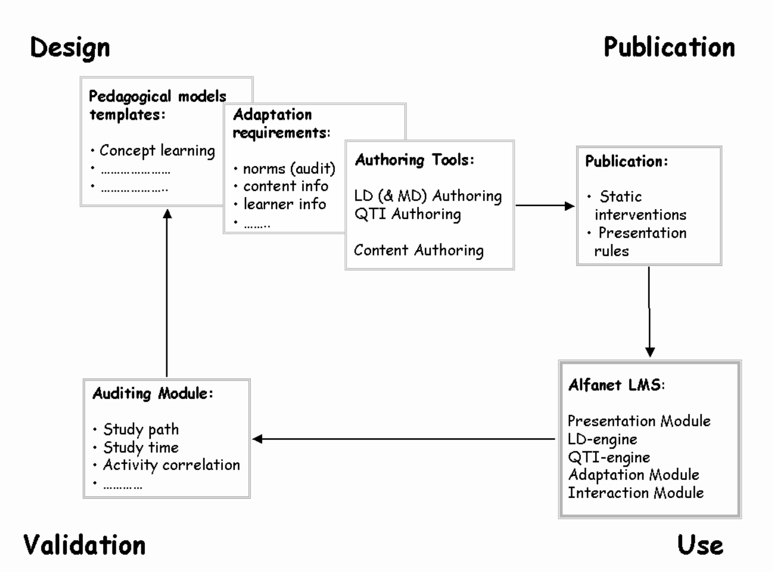


Figure 9.1 An overview of the aLFanet four step life cycle model.

Within this setting, our hypothesis is that this standards-based framework and the combination of tools and documentation, indeed facilitates and simplifies the authors' task of designing adaptive e-learning. The evaluation follows a design research approach. In three successive rounds a group of authors is observed while using the system. The results of each round are used to drive the further development of the system. The results acquired through the evaluation are mixed. The overall opinion of the authors was one of dissatisfaction with complaints that the tools were too complex to work with and that it was too difficult to make full use of the adaptivity offered. Nevertheless, there are also a number of positive outcomes. The use of the system shows that it is possible to support adaptive e-learning by using standards and a set of services. At the authoring side, the tools for editing IMS-LD and IMS-QTI are a (small) improvement on editing the underlying XML directly. The authors deem the use of a template, the guidelines and the adaptation features as being useful, even if they do not meet the required level of usability.

Facilitating Support Activities
Our second study is a detailed study on a model that intends to address one particular example of a typically time-consuming task of staff, i.e. answering content-related questions. In Chapter 4 we elaborate upon the requirements of the model, review existing approaches and - underpinned by these findings – detail the model and discuss its first implementation. We define four requirements for our model:

- It has to alleviate the support task for the tutor.
- It has to involve a substantial part of the student population.
- It should be able to support the students in their task.
- It should be portable.

A cursory search for ways to answer content related questions reveals a wide choice of solutions, ranging from groupware, help-desks to virtual assistants. In addition, we consider agents as a possibility for conceiving distributed learning applications and, because of its potential relevance to assist in answering questions to LSA, an example of a language technology. For our model, we opted for the unique combination of setting up a small group of selected peers to give the answer with the help of documents selected from the course.

The main steps of the model are given in Table 9.1. Of particular interest is step 2: with the help of LSA we determine the text fragments most relevant to the question. Text fragments always come from a unique Activity Node. Hence, once having the most relevant text fragments, we also know the Activity Node(s) that is (are) relevant for the question. In step 2c the model selects the most suitable students to assist. Their selection is based on a weighted sum of four criteria:

- *content competency,* indicating if a learner has successfully finished the Activity Node(s) relevant for the question;
- *tutor competency,* indicating past performance in the role of peer-tutor;
- *availability*, a combination of actually being available and how many question this peer has previously answered (workload balance);
- *eligibility*, a similarity measure relating the progress of the peer-student with the student posing the question.

A test with the selection procedure shows that it is possible to spread the selection of the students depending on the weights of the criteria.

Table 9.1 The main steps of the model.

Pre-condition	A Learning Network with a set of Activity Nodes and a set of users with their profiles indicating their progress with regard to the Activity Nodes
Main steps	1. *Anne* poses a question. 2. The *system* determines: a. the most relevant text fragments; b. the appropriate Activity Nodes; c. the most suitable students. 3. The *system* sets up a wiki with the question, the text fragments and guidelines. 4. The selected *students* receive an invitation to assist. 5. *Anne and the peer-students* discuss and formulate an answer in the wiki. 6. If answered (or after a given period of time) *Anne* closes the discussion and rates the answer.
Post-condition	The answer is stored.

In Chapter 5 we focus on a key aspect of our model, i.e. the use of LSA. A successful usage of language technologies such as LSA very much depends on the corpus, its preparation and the parameters applied. We prepare a corpus from an existing Learning Network on ‘Internet Basics’ and calibrate the LSA-parameters with a simple depth-first strategy. This means we start with an initial setting derived from literature and then optimise the main parameters one by one. We determined the value of the parameters with the help of 16 questions selected from the assessment section of the Activity Nodes of the Learning Network. Therefore – in principle – each question can be mapped to a single known Activity Node. In the optimal combination of parameters, LSA correctly identifies 15 out of 16 questions. We double check our findings by formulating 16 new questions we feel students may ask. This time LSA identifies 12 questions correctly. In addition for this case the designers of the Learning Network rate the selected text fragments. The designers indicate that LSA suggests useful text fragments for 7 questions, and that 5 questions are beyond the scope of the Learning Network, giving a total of 7 out of 11 with useful suggestions.

Having designed and developed a model that fits our goals and having calibrated it in order to function optimally, our next step is a field test. In Chapter 6 we describe and discuss the results of an experiment with a prototype of the model. For the experiment, we set up a course in the Learning Network on Internet Basics, the same course that was used to calibrate the model. The course is a free of charge (no tuition) course that lasts 8 weeks. Students are invited from among students and staff of our organisation. They are divided at random into two groups. In the experimental group, we use the model to select the students to help answer a question. In the control group, we only make sure that the questions will be allotted evenly to the students. In the experiment we test three related hypotheses:

A. The model should solve at least 50% of the content-related questions posed by students, without invoking any staff support.
B. The groups that are composed by the model should outperform groups whose members have been selected with workload balancing in mind only.
C. The text fragments selected by the system help peers to answer the questions posed to them.

To verify our hypotheses we look at a combination of logging data, student ratings, staff ratings and data from a questionnaire with the following results:

A. The experimental group solves 75% of the questions, against 52% of the control group.
B. The experimental group outperforms the control group. This means the number of invited students per question is significantly lower; the answer time is significantly shorter; and the number of questions solved is significantly larger.
C. The results of the last hypothesis are less clear. The far majority of the respondents value the text fragments although their actual usage is limited.

Finally, it is important to note that the general response to the system is positive. Almost all respondents in the questionnaire agree that the time it takes to answer a peer's question is time well spent. Two types of reasons were selected:

- "I am aware that other students also have questions";
- "It improved my knowledge and understanding".

This is confirmed by the outcome that most indicate they want to use this question-answering approach in other courses.

Conclusions

The two most important conclusions of the study are:

(1) The standards-based framework with its adaptive services and the connected tools may stimulate a wider implementation of adaptive e-learning. However, before it can do so, the requirements of the adaptive features have to be fully transparent and the tools should indeed support the authors. This can only be achieved by taking into account the authors' knowledge and experience and by making an optimal use of design specific constraints.
(2) The experiment with the question-answering model shows that it is possible satisfactorily to solve a substantial number of questions students have by involving their peers. In a well-structured process the students are willing

and competent to support each other, moreover they indicate that supporting each other is a good investment of their time. A selection mechanism that takes into account the competence, availability and eligibility of the participants clearly increases their willingness to participate and the results.

For the authoring we propose to further investigate the role adaptive services may have and to explore a variety of pedagogically sound templates (or patterns). In both cases one should, in order to be successful, not merely look at the functional aspects but equally well concentrate on the authoring aspects. For our question-answering model we distinguish three different directions for further research. First, there is the need to investigate how the results may vary under different conditions such as altered complexity of the domain, different population characteristics, policies (e.g. a reward mechanism), and different selection criteria. Second, one should explore different kinds of tasks e.g. peer-support on other types of questions such as ‘what to study’, ‘how to write a study paper’, or questions on administrative issues. Finally, there is the important issue that such a question-answering system may be used to stimulate community formation. The system as it is implemented now creates so-called ad hoc transient communities, small groups whose task it is to answer a question and disappear once the answer is in. Now that students know each other through this device they may become motivated to maintain contact, also in other situations. This way lifelong learners will be helped to (self-)organize themselves into communities.

Samenvatting

Samenvatting

De centrale vraag in dit proefschrift is op welke wijze docenten ondersteund kunnen worden bij het ontwerpen en begeleiden van gepersonaliseerde leeromgevingen (adaptief e-learning). De opkomst van de kenniseconomie en de individuele eisen van 'lifelong learners' zijn twee gegevens die het toegenomen belang benadrukken van de flexibiliteit van onderwijsaanbod in plaats, tijd en tempo (logistieke flexibiliteit) en voor leeractiviteiten die zich inhoudelijk aanpassen aan de individuele wensen van de student (inhoudelijke flexibiliteit). Deze eisen zetten een hoge, extra druk op de docenten aangezien zij het zijn die aan deze eisen tegemoet moeten komen. Dit onderzoek bestudeert twee verschillende wijzen van aanpak om een bijdrage te leveren aan de gewenste flexibiliteit. De eerste studie – 'het ontwerp van gepersonaliseerde leeromgevingen' – onderzoekt op welke wijze docenten ondersteund kunnen worden bij het ontwerp van adaptieve leeractiviteiten. De tweede studie – 'het faciliteren van ondersteuningsactiviteiten' – bestudeert op welke wijze docenten ondersteund kunnen worden bij het geven van begeleiding en ondersteuning aan studenten.

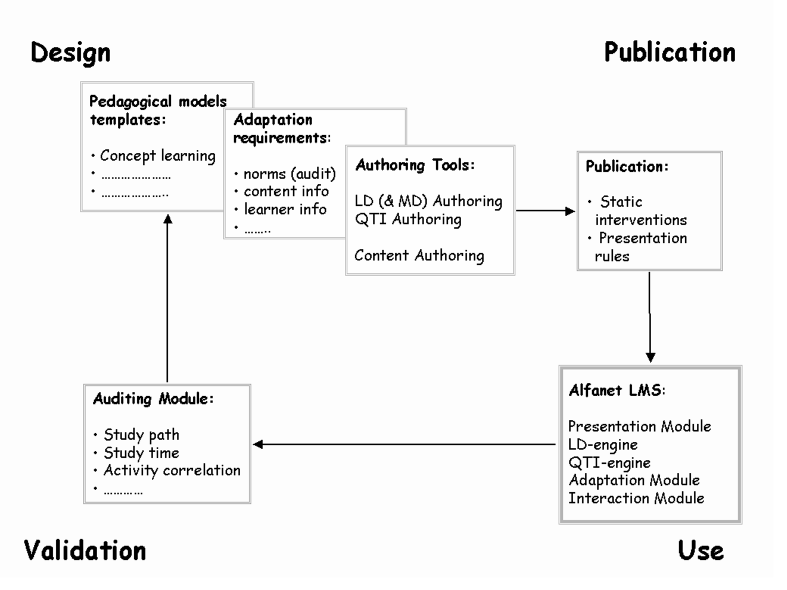
Het ontwerp van gepersonaliseerde leeromgevingen
De eerste studie in dit proefschrift is een exploratief onderzoek naar de wijze waarop de ontwerptaak van docenten ondersteund kan worden. Hoofdstuk 2 bespreekt de achtergrond van dit onderzoek. We introduceren het doel van ons systeem en de onderliggende eisen, dat wil zeggen het moet: (1) actief en adaptief e-learning ondersteunen; (2) open zijn met betrekking tot het inzetten van verschillende onderwijskundige modellen en nieuwe componenten; (3) de gebruiker op efficiënte wijze ondersteunen. Een review van bestaande, algemeen gebruikte systemen laat zien dat het in deze systemen niet mogelijk is om inhoud en ontwerp gescheiden te houden. Hierdoor zijn deze systemen beperkt in welke type e-learning ze ondersteunen en in de mogelijkheden om het ontwerpproces efficiënt te ondersteunen. Na een overzicht van relevante tools, technologieën en methoden presenteren wij ons raamwerk. De belangrijkste eigenschappen van ons raamwerk zijn het gebruik van vijf e-learning standaarden, dit om de vereiste openheid te borgen, en het gebruik van agents (autonome stukken software) om een deel van de functionaliteit te realiseren. Verder onderscheidt het zich doordat het twee aanpakken voor adaptiviteit integreert en ondersteunt, dat wil zeggen 'design-time' en 'runtime' adaptiviteit. De belangrijkste standaard die gebruikt wordt, is IMS-Learning Design. IMS-LD biedt een semantische notatie om een onderwijskundig scenario op formele wijze te beschrijven. Een docent of een ontwerpteam kan tijdens de ontwerpfase een learning design model maken, aanpassen of inspecteren en het desgewenst gebruiken in meerdere cursussen. Tijdens de

uitvoering van een cursus kan een begeleider het model en de voortgang van de studenten bekijken en waar nodig actie ondernemen, bijvoorbeeld een suggestie geven aan studenten. De technische haalbaarheid van het raamwerk wordt aangetoond door een mock-up van Edubox (een op EML gebaseerde e-learning omgeving; EML is een voorloper van IMS-LD) met daaraan gekoppeld twee agents.

Hoofdstuk 3 concentreert zich op de probleemstelling van het eerste deel van dit proefschrift, dat wil zeggen gegeven het beschikbare raamwerk: op welke wijze kunnen auteurs ondersteund worden in het ontwerp van adaptief e-learning. We beginnen met een bespreking van het daadwerkelijk gebruikte systeem (zoals het stapsgewijs ontwikkeld is), de componenten en de type adaptiviteit die ze ondersteunen. Auteurs hebben twee mogelijkheden om adaptief e-learning te ontwikkelen. Ze kunnen de gewenste adaptiviteit rechtstreeks in IMS-LD uitdrukken of gebruik maken van de faciliteiten van de agents en de daarvoor noodzakelijk gegevens invoeren. Voordat auteurs gebruik kunnen maken van het systeem moet er echter aan een aantal voorwaarden voldaan worden. De twee belangrijkste zijn dat de beschikbare adaptatie-mogelijkheden transparant moeten zijn, dat wil zeggen het moet duidelijk zijn wat de adaptiviteit doet, hoe ze werkt (en waarom) en wat er nodig is zodat ze gebruikt kan worden; èn het auteursproces moet ondersteund worden met tools en richtlijnen om ze daadwerkelijk te kunnen gebruiken. Om aan deze voorwaarden te voldoen hebben de auteurs tot hun beschikking een combinatie van:

- tools en manuals (IMS-LD en IMS-QTI auteurstools);
- documentatie, waaronder een beschrijving van het aLFanet life-cycle model voor adaptatie (Figuur 10.1) met uitleg van de beschikbare adaptatie mogelijkheden;
- een ‘concept learning’ template.

De hypothese die we onderzoeken is of dit op standaarden gebaseerde raamwerk en de combinatie van tools en documentatie inderdaad de taak van de auteur faciliteert en eenvoudig maakt om adaptief e-learning te ontwerpen. De evaluatie volgt een design-research benadering. We volgen een groep auteurs in drie opeenvolgende rondes tijdens het gebruik van het systeem. De bevindingen van elke ronde worden gebruikt voor de verdere ontwikkeling van het systeem. De resultaten van de evaluatie laten een wisselend beeld zien. In het algemeen is het oordeel van de auteurs negatief, ze zijn niet tevreden over het gebruik van de tools. De tools zijn te complex om mee te werken en het is te moeilijk om volledig gebruik te maken van de geboden adaptiviteit. Desalniettemin zijn er ook een aantal positieve resultaten. Het gebruik van het systeem laat zien dat het mogelijk is adaptief e-learning te ondersteunen door middel van het gebruik van standaarden en een verzameling services. De geboden tools voor het bewerken van IMS-LD en IMS-QTI zijn een (kleine) verbetering in vergelijking met het rechtstreeks bewerken van de onderliggende XML. Ook het gebruik van het template, de bijbehorende instructies en de geboden opties voor adaptiviteit worden positief beoordeeld hoewel ze duidelijk niet gebruikersvriendelijk genoeg zijn.



Figuur 10.1 Een overzicht van het aLFanet 'vier stappen life cycle model'.

Het faciliteren van ondersteuningsactiviteiten

De tweede studie is een detailstudie over een model dat een specifiek voorbeeld van een arbeidsintensieve taak van een docent probeert te verlichten, te weten het beantwoorden van inhoudelijke vragen. In hoofdstuk 4 werken we de eisen van dit model uit, bespreken we bestaande benaderingen en – hierdoor geïnspireerd – bediscussiëren we de eerste implementatie. We definiëren vier eisen voor ons model:

- Het moet de ondersteuningstaak van de docent verlichten;
- Het moet een substantieel deel van de studenten er actief bij betrekken;
- Het moet de studenten ondersteunen in hun taak;
- Het moet portable zijn.

Een globaal onderzoek naar beschikbare wijzen van aanpak om vragen te beantwoorden laat een breed spectrum van mogelijke oplossingen zien, variërend van groupware, helpdesks tot virtuele assistenten. Hiernaast nemen we ook het gebruik van agents in overweging als mogelijkheid om gedistribueerde systemen mee te ontwerpen en, vanwege het vermogen om het beantwoorden van vragen te ondersteunen LSA (Latente Semantische

Analyse), een voorbeeld van taaltechnologie. Voor ons model kiezen we voor de unieke combinatie van het inschakelen van een kleine groep studenten, om het antwoord te geven, ondersteund door documenten geselecteerd uit de cursus.

De belangrijkste stappen van het model staan beschreven in Tabel 10.1. Stap 2 is speciaal van belang: met behulp van LSA selecteren we uit de cursus de belangrijkste tekstfragmenten met betrekking tot de gestelde vraag. Teksten behoren altijd tot een unieke Activity Node. Met andere woorden, zodra we de belangrijkste tekstfragmenten geselecteerd hebben, kennen we ook de Activity Node(s) die relevant is (zijn) voor de desbetreffende vraag. In stap 2c selecteert het model de studenten die het meest geschikt zijn om te helpen. Deze selectie wordt gebaseerd op een gewogen som van de volgende vier criteria:

- *'content competency'* geeft aan of een student de Activity Node(s), die relevant is (zijn) voor de vraag, succesvol heeft afgesloten;
- *'tutor competency'* geeft de prestatie aan van de student in zijn rol als peer-tutor;
- *'availability'* is een combinatie van daadwerkelijk beschikbaar zijn samen met de werkdruk tot op heden, dat wil zeggen hoeveel vragen een student al heeft beantwoord;
- *'eligibility'*, is een gelijkheidsmaat die de studievoortgang van een student t.o.v. de vraagsteller aangeeft.

Een test met de selectiemethode laat zien dat het mogelijk is om de keuze van de studenten te spreiden afhankelijk van de keuze van de gewichten van de criteria.

Tabel 10.1 De hoofdstappen van het model.

Preconditie	Een Leernetwerk met een verzameling Activity Nodes en een groep gebruikers met profielen waarin hun voortgang aangegeven is met betrekking tot de Activity Nodes.
Hoofdstappen	1. *Anne* stelt een vraag. 2. Het *systeem* bepaalt: a. de relevantste tekst fragmenten; b. de juiste Activity Nodes; c. de geschiktste studenten. 3. Het *systeem* creëert een wiki met de vraag, de tekstfragmenten en richtlijnen. 4. De geselecteerde *studenten* ontvangen een uitnodiging om te helpen. 5. *Anne en de medestudenten* bediscussiëren en formuleren een antwoord in de wiki. 6. Als de vraag beantwoord is (of na een vooraf bepaalde tijd) sluit *Anne* de discussie en beoordeelt het antwoord met een rating.
Postconditie	Het antwoord wordt opgeslagen.

In hoofdstuk 5 behandelen we een hoofdonderdeel van ons model, dat wil zeggen het gebruik van LSA. Een succesvolle toepassing van taaltechnologie zoals LSA hangt in belangrijke mate af van het corpus, zijn bereiding en de waarde van de parameters. We stellen een corpus samen uit een bestaand Leernetwerk in het onderwerp 'Introductie Internet' en calibreren de LSA-parameters met een eenvoudige depth-first strategie. Dat willen zeggen dat de initiële parameterwaarden afgeleid zijn uit de literatuur en dat de parameterwaarden vervolgens één voor één geoptimaliseerd zijn. We bepalen de optimale waarden met behulp van 16 originele toetsvragen uit het toetsdeel van de Activity Nodes van het Leernetwerk. Met andere woorden - in principe – kan iedere vraag op precies een vooraf bekende Activity Node afgebeeld worden. LSA beeldt 15 van de 16 vragen correct af bij de optimale combinatie van parameterwaarden. We controleren onze bevindingen door 16 nieuwe vragen te formuleren, zoals die ook door studenten gevraagd zouden kunnen worden. In dit geval herkent LSA 12 vragen. Bij deze vragen bekijken we ook de tekstfragmenten die LSA aanraadt. De ontwikkelaars van de cursus geven aan dat LSA bij 7 vragen nuttige fragmenten aanraadt en dat 5 vragen buiten de leerstof van het Leernetwerk vallen, met andere woorden bij 7 van de 11 vragen is er een bruikbaar advies.

Onze volgende stap, na het ontwerp en de ontwikkeling van het beoogde model, is een experiment. In hoofdstuk 6 beschrijven en bediscussiëren we de resultaten van een experiment met een prototype van het model. Om het experiment uit te kunnen voeren stellen we de cursus Leernetwerk Introductie Internet ter beschikking, dezelfde cursus als gebruikt voor de calibratie van het model. De cursus is een gratis cursus met 8 weken doorlooptijd. De deelnemers bestaan uit staf en studenten van onze organisatie. Ze zijn willekeurig verdeeld over twee groepen. In de experimentele groep gebruiken we het model om de studenten uit te kiezen die assisteren bij het beantwoorden van een vraag. In de controle groep zorgen we er alleen voor dat de vragen evenredig verdeeld worden over de studenten. In het experiment testen we drie hypotheses:

A. Het model moet minstens 50% van de inhoudelijke, door de studenten gestelde vragen oplossen, zonder inmenging van de docent.
B. De groepen die door het model zijn samengesteld moeten het significant beter doen dan de groepen die alleen samengesteld zijn op basis van een eerlijke verdeling van de vragen.
C. De geselecteerde tekstfragmenten ondersteunen de studenten bij het beantwoorden van de gestelde vragen.

Om de hypotheses te toetsen kijken we naar een combinatie van data loggings, ratings door studenten en staf, en de resultaten van een slotenquête. Dit geeft de volgende resultaten:

A. De experimentele groep lost 75% van de vragen op tegenover 52% in de controle groep.
B. De experimentele groep doet het significant beter dan de controle groep. Het aantal studenten dat uitgenodigd moet worden per vraag is significant lager; de tijd nodig om een antwoord te geven is significant korter en het aantal vragen dat opgelost wordt, is significant hoger.
C. De uitkomsten voor de laatste hypothese zijn minder eenduidig. Het overgrote deel van de respondenten is positief over de tekstfragmenten, maar het daadwerkelijk gebruik is beperkt.

Tot slot nog een belangrijke waarneming: het algemeen oordeel van de gebruikers over het model is positief. Bijna alle respondenten van de enquête zijn het er over eens dat de tijd die het kost om een vraag te beantwoorden goed besteed is. Ze geven hiervoor de volgende twee redenen aan:

- “Ik zie dat andere studenten ook vragen hebben”;
- “Het heeft mijn kennis en inzicht verbeterd”.

Het beeld wordt bevestigd door het gegeven dat de meeste respondenten aangeven dat ze dit model om vragen te beantwoorden ook in andere cursussen willen gebruiken.

Conclusies

De twee belangrijkste conclusies van deze studie zijn:

(1) Het op standaarden-gebaseerde raamwerk met zijn adaptieve services en de bijbehorende tools heeft de mogelijkheid in zich om een bredere implementatie van gepersonaliseerde leeromgevingen te ondersteunen. Echter, voor het zover is, moeten de onderliggende eisen van de geboden adaptiviteit volledig transparant zijn en moeten de beschikbare tools de auteurs daadwerkelijk ondersteunen. Dit kan alleen bereikt worden door de kennis en ervaring van de auteurs in acht te nemen en door optimaal gebruik te maken van de restricties die volgen uit het ontwerp.

(2) Het experiment met het vraag-antwoord model laat zien dat het mogelijk is om op bevredigende wijze een substantieel aantal van de vragen van studenten op te lossen door hun medestudenten in te schakelen. Studenten zijn, in een goed gestructureerd proces, bereid en competent om elkaar te ondersteunen en niet alleen dat, ze geven ook aan dat ze het een goede investering van hun tijd vinden om elkaar te helpen. Een selectiemethode die rekening houdt met de ‘content competency’, ‘availability’ en ‘eligibility’ verhoogt duidelijk de bereidheid om te helpen en de uiteindelijke resultaten.

Bij het ontwerp van gepersonaliseerde leeromgevingen stellen we met name voor om nader te kijken naar de bijdrage die adaptieve services kunnen bieden en om het gebruik van een palet van onderwijskundig onderbouwde templates (of patterns) te onderzoeken. In beide gevallen is het voor een succesvolle bijdrage cruciaal om niet alleen naar de functionele aspecten te kijken maar evenzeer aandacht te besteden aan de auteursaspecten. Voor het vraag-antwoord model onderscheiden we drie verschillende richtingen voor verder onderzoek. Allereerst is het noodzakelijk om te kijken hoe de resultaten beïnvloed worden door verschillende condities zoals een andere complexiteit van het studiedomein, andere eigenschappen van de studentpopulatie, ondersteunende maatregelen (zoals het geven van studiepunten), en andere selectie criteria. Als tweede kunnen verschillende soorten taken onderzocht worden, bijvoorbeeld peerondersteuning voor vragen zoals 'wat zal ik nu gaan bestuderen', 'hoe schrijf ik een essay' of administratief georiënteerde vragen. Tot slot ligt er een belangrijke onderzoeksvraag op het gebied van groepsvorming. Het systeem, zoals het nu geïmplementeerd is, stelt voor elke vraag zogeheten 'ad hoc transient communities' samen. Dit zijn kleine groepen met de taak een vraag te beantwoorden. Deze groepen verdwijnen zodra de vraag opgelost is. Op deze wijze leggen studenten contacten met elkaar die ze ook in andere situaties kunnen gebruiken. 'Lifelong learners' kunnen zo ondersteund worden om zichzelf te organiseren in groepen.

Acknowledgement

Acknowledgement

Writing a thesis besides a set of critical assignments and a second job is definitely a challenge. It is like “Around the world in 80 days”. It will be obvious that without a lot of support and without a very good cooperation with the people of the projects from which this thesis emerged, this thesis would not have been possible.

With the serious risk of forgetting people (I apologize in case) I will try to express my thanks to everyone who supported me. First of all, my two promotors. Rob Koper, who insisted I should write my thesis and subsequently offered me the opportunity and his support, and Peter Sloep, who for more than two years each week kept me on track, shared with me his views on my work, and gave his indispensable help to improve on the quality of my papers.

Special thanks to Mieke Haemers who assisted me in numerous things ranging from supporting the organization of my second experiment, to proof reading of all my output and assisting in the production of this thesis, to keeping up the good spirit and helping me survive my OTEC-jungle (keeping a record of some of my most prominent statements). To Francis Brouns who always was there to assist me in my understanding and in getting my wonderfully stable tools and software up and running. Without her help, the calibration, simulation, and experiment with the second model would not have been possible. To Hubert Vogten for his mental and creative support, mental because he joined me in almost all aLFanet meetings, also the challenging ones, but also being like me a ‘second-chance’ phd-student, who shared his positive view on our world many times with me. He will now have to keep the ‘light on’ himself while finalizing his thesis. Creative because he contributed (often unknown) to my work in the many on- and off-topic discussions we had. And to Eric Kluijfhout, Eric who just worked extra on our shared responsibility TENCompetence-WP1, besides all the work he already did, when I was running out of time in particular in the last year.

Many thanks also to all the people in the aLFanet project team: Cristina Arana, Carlos Fuentes, Jesús González Boticario, Carmen Barrera, Olga Santos, Juergen A. Schmidt, Ingeborg Hoke, Elsa Escala, Adalberto Moutinho, Francisco Barros, Roberto Canada, Hubert Vogten, Harrie Martens, René van Es, Patricia Poelmans, Frans Mofers, Harrie Passier, Slavi Stoyanov, John van der Baaren, Leo Wagemans and many others. A wonderful group of people and a project team which had many and intense discussions on what to do and how to continue, but also a group that ended with great mutual respect for each other.

Many thanks to the ASA project team who helped me in countless discussions to shape the ideas behind the system of the question-answering experiment: Peter Sloep, Francis Brouns, Liesbeth Kester, Marlies Bitter, Adriana Berlanga, Kees Pannekeet, Marcel de Croock, Malik Koné, Sibren Fetter, Maurice Brouwers and in particular to André de Jong who in his special way created the final version of the software of the question-answering model.

Many thanks to Jan Hensgens my colleague for many years in RIKS and later at AURUS KTS, who regularly gave me a valuable second opinion and who also kept me aligned with the other side of the coin, i.e. highly innovative projects with objectives to be achieved in a relatively short period of time.

Many thanks also to current and past members of the Development team: Jan van Bruggen for his advices and cooperation on the use of LSA; Colin Tattersall, José Janssen and Tally Schmeits for their help and advice in the question-answering simulation and experiment; Harrie Martens, who was together with Hubert, responsible for our ICT-part in aLFanet; René van Es for his work in the aLFanet project and the many ideas we shared while carpooling, and Marco Kalz for our many discussions on and off topic.

And many thanks to my family and friends. In particular my parents who opened, already decades ago, my gate to the university and therewith to this thesis; and last but not least Yvonne, Sasja and Rik, for their genuine support, their critical remarks, and their sincere unbelief expressed about my world and its challenges.

Curriculum Vitae

Curriculum Vitae

Peter van Rosmalen has been active in educational technology and knowledge management since the early eighties for the financial sector, government and for large industrial companies in Denmark and the Netherlands. He started his career at Delft University as one of the pioneers of using computers in education. From Delft he moved to one of the first commercial companies in the Netherlands aimed at designing and developing courseware and courseware tools at a large scale. In 1986 he was asked for a position by a large savings bank in Denmark to assist in the further development of their newly established computer based training department. At the end of 1989 he returned to the Netherlands to the Research Institute for Knowledge Systems of which he became co-director in 1999. In addition, from 1998-1999, he worked part time for the University Maastricht at the Maastricht McLuhan Institute as project manager ELECTRA II and senior advisor Lifelong Learning. In 2000 he was co-founder and director of a company in e-learning and knowledge management. He joined OTEC in 2003. He started in the position of researcher, since 2005 he has been appointed as assistant professor.

Throughout his career he has been involved in many large innovation projects for industry, government and finance. Moreover, he has been active as a researcher and as a project manager for a large number of Dutch and European R&D projects on topics such as authoring tools for intelligent courseware (MATIC – EU Delta Program), simulations (SAM – EU 3rd framework), e-universities and computer supported cooperative learning (ELECTRA II and MODEM – 4th framework), adaptive e-learning (aLFanet – 5th framework), language learning (TUM – Dutch Ministry of Education), personal competence development (TENCompetence – 6th framework). Most recently, he successfully took the lead in acquiring a project focused on the use of language technologies for e-learning (LTfLL – 7th framework).

His current work focuses on question-answering and community formation with a focus on establishing effective and efficient learner and learning related interactions and the use of language technologies in e-learning.

SIKS Dissertatiereeks

SIKS Dissertatiereeks

====
1998
====

1998-1 Johan van den Akker (CWI)
DEGAS - An Active, Temporal Database of Autonomous Objects

1998-2 Floris Wiesman (UM)
Information Retrieval by Graphically Browsing Meta-Information

1998-3 Ans Steuten (TUD)
A Contribution to the Linguistic Analysis of Business Conversations within the Language/Action Perspective

1998-4 Dennis Breuker (UM)
Memory versus Search in Games

1998-5 E.W.Oskamp (RUL)
Computerondersteuning bij Straftoemeting

====
1999
====

1999-1 Mark Sloof (VU)
Physiology of Quality Change Modelling; Automated modelling of Quality Change of Agricultural Products

1999-2 Rob Potharst (EUR)
Classification using decision trees and neural nets

1999-3 Don Beal (UM)
The Nature of Minimax Search

1999-4 Jacques Penders (UM)
The practical Art of Moving Physical Objects

1999-5 Aldo de Moor (KUB)
Empowering Communities: A Method for the Legitimate User-Driven Specification of Network Information Systems

1999-6 Niek J.E. Wijngaards (VU)
Re-design of compositional systems

1999-7 David Spelt (UT)
Verification support for object database design

1999-8 Jacques H.J. Lenting (UM)
Informed Gambling: Conception and Analysis of a Multi-Agent Mechanism
for Discrete Reallocation.

====
2000
====

2000-1 Frank Niessink (VU)
Perspectives on Improving Software Maintenance

2000-2 Koen Holtman (TUE)
Prototyping of CMS Storage Management

2000-3 Carolien M.T. Metselaar (UVA)
Sociaal-organisatorische gevolgen van kennistechnologie;
een procesbenadering en actorperspectief.

2000-4 Geert de Haan (VU)
ETAG, A Formal Model of Competence Knowledge for User Interface Design

2000-5 Ruud van der Pol (UM)
Knowledge-based Query Formulation in Information Retrieval.

2000-6 Rogier van Eijk (UU)
Programming Languages for Agent Communication

2000-7 Niels Peek (UU)
Decision-theoretic Planning of Clinical Patient Management

2000-8 Veerle Coup, (EUR)
Sensitivity Analyis of Decision-Theoretic Networks

2000-9 Florian Waas (CWI)
Principles of Probabilistic Query Optimization

2000-10 Niels Nes (CWI)
Image Database Management System Design Considerations, Algorithms and Architecture

2000-11 Jonas Karlsson (CWI)
Scalable Distributed Data Structures for Database Management

====
2001
====

2001-1 Silja Renooij (UU)
Qualitative Approaches to Quantifying Probabilistic Networks

2001-2 Koen Hindriks (UU)
Agent Programming Languages: Programming with Mental Models

2001-3 Maarten van Someren (UvA)
Learning as problem solving

2001-4 Evgueni Smirnov (UM)
Conjunctive and Disjunctive Version Spaces with Instance-Based Boundary Sets

2001-5 Jacco van Ossenbruggen (VU)
Processing Structured Hypermedia: A Matter of Style

2001-6 Martijn van Welie (VU)
Task-based User Interface Design

2001-7 Bastiaan Schonhage (VU)
Diva: Architectural Perspectives on Information Visualization

2001-8 Pascal van Eck (VU)
A Compositional Semantic Structure for Multi-Agent Systems Dynamics.

2001-9 Pieter Jan 't Hoen (RUL)
Towards Distributed Development of Large Object-Oriented Models,
Views of Packages as Classes

2001-10 Maarten Sierhuis (UvA)
Modeling and Simulating Work Practice
BRAHMS: a multiagent modeling and simulation language for
work practice analysis and design

2001-11 Tom M. van Engers (VUA)
Knowledge Management:
The Role of Mental Models in Business Systems Design

====
2002
====

2002-01 Nico Lassing (VU)
Architecture-Level Modifiability Analysis

2002-02 Roelof van Zwol (UT)
Modelling and searching web-based document collections

2002-03 Henk Ernst Blok (UT)
Database Optimization Aspects for Information Retrieval

2002-04 Juan Roberto Castelo Valdueza (UU)
The Discrete Acyclic Digraph Markov Model in Data Mining

2002-05 Radu Serban (VU)
The Private Cyberspace Modeling Electronic
Environments inhabited by Privacy-concerned Agents

2002-06 Laurens Mommers (UL)
Applied legal epistemology; Building a knowledge-based ontology of
the legal domain

2002-07 Peter Boncz (CWI)
Monet: A Next-Generation DBMS Kernel For Query-Intensive Applications

2002-08 Jaap Gordijn (VU)
Value Based Requirements Engineering: Exploring Innovative E-Commerce Ideas

2002-09 Willem-Jan van den Heuvel(KUB)
Integrating Modern Business Applications with Objectified Legacy Systems

2002-10 Brian Sheppard (UM)
Towards Perfect Play of Scrabble

2002-11 Wouter C.A. Wijngaards (VU)
Agent Based Modelling of Dynamics: Biological and Organisational Applications

2002-12 Albrecht Schmidt (Uva)
Processing XML in Database Systems

2002-13 Hongjing Wu (TUE)
A Reference Architecture for Adaptive Hypermedia Applications

2002-14 Wieke de Vries (UU)
Agent Interaction: Abstract Approaches to Modelling, Programming and Verifying Multi-Agent Systems

2002-15 Rik Eshuis (UT)
Semantics and Verification of UML Activity Diagrams for Workflow Modelling

2002-16 Pieter van Langen (VU)
The Anatomy of Design: Foundations, Models and Applications

2002-17 Stefan Manegold (UVA)
Understanding, Modeling, and Improving Main-Memory Database Performance

====
2003
====

2003-01 Heiner Stuckenschmidt (VU)
Ontology-Based Information Sharing in Weakly Structured Environments

2003-02 Jan Broersen (VU)
Modal Action Logics for Reasoning About Reactive Systems

2003-03 Martijn Schuemie (TUD)
Human-Computer Interaction and Presence in Virtual Reality Exposure Therapy

2003-04 Milan Petkovic (UT)
Content-Based Video Retrieval Supported by Database Technology

2003-05 Jos Lehmann (UVA)
Causation in Artificial Intelligence and Law - A modelling approach

2003-06 Boris van Schooten (UT)
Development and specification of virtual environments

2003-07 Machiel Jansen (UvA)
Formal Explorations of Knowledge Intensive Tasks

2003-08 Yongping Ran (UM)
Repair Based Scheduling

2003-09 Rens Kortmann (UM)
The resolution of visually guided behaviour

2003-10 Andreas Lincke (UvT)
Electronic Business Negotiation: Some experimental studies on the interaction between medium, innovation context and culture

2003-11 Simon Keizer (UT)
Reasoning under Uncertainty in Natural Language Dialogue using Bayesian Networks

2003-12 Roeland Ordelman (UT)
Dutch speech recognition in multimedia information retrieval

2003-13 Jeroen Donkers (UM)
Nosce Hostem - Searching with Opponent Models

2003-14 Stijn Hoppenbrouwers (KUN)
Freezing Language: Conceptualisation Processes across ICT-Supported Organisations

2003-15 Mathijs de Weerdt (TUD)
Plan Merging in Multi-Agent Systems

2003-16 Menzo Windhouwer (CWI)
Feature Grammar Systems - Incremental Maintenance of Indexes to Digital Media Warehouses

2003-17 David Jansen (UT)
Extensions of Statecharts with Probability, Time, and Stochastic Timing

2003-18 Levente Kocsis (UM)
Learning Search Decisions

====
2004
====

2004-01 Virginia Dignum (UU)
A Model for Organizational Interaction: Based on Agents, Founded in Logic

2004-02 Lai Xu (UvT)
Monitoring Multi-party Contracts for E-business

2004-03 Perry Groot (VU)
A Theoretical and Empirical Analysis of Approximation in Symbolic Problem Solving

2004-04 Chris van Aart (UVA)
Organizational Principles for Multi-Agent Architectures

2004-05 Viara Popova (EUR)
Knowledge discovery and monotonicity

2004-06 Bart-Jan Hommes (TUD)
The Evaluation of Business Process Modeling Techniques

2004-07 Elise Boltjes (UM)
Voorbeeldig onderwijs; voorbeeldgestuurd onderwijs, een opstap naar abstract denken, vooral voor meisjes

2004-08 Joop Verbeek(UM)
Politie en de Nieuwe Internationale Informatiemarkt, Grensregionale politiële gegevensuitwisseling en digitale expertise

2004-09 Martin Caminada (VU)
For the Sake of the Argument; explorations into argument-based reasoning

2004-10 Suzanne Kabel (UVA)
Knowledge-rich indexing of learning-objects

2004-11 Michel Klein (VU)
Change Management for Distributed Ontologies

2004-12 The Duy Bui (UT)
Creating emotions and facial expressions for embodied agents

2004-13 Wojciech Jamroga (UT)
Using Multiple Models of Reality: On Agents who Know how to Play

2004-14 Paul Harrenstein (UU)
Logic in Conflict. Logical Explorations in Strategic Equilibrium

2004-15 Arno Knobbe (UU)
Multi-Relational Data Mining

2004-16 Federico Divina (VU)
Hybrid Genetic Relational Search for Inductive Learning

2004-17 Mark Winands (UM)
Informed Search in Complex Games

2004-18 Vania Bessa Machado (UvA)
Supporting the Construction of Qualitative Knowledge Models

2004-19 Thijs Westerveld (UT)
Using generative probabilistic models for multimedia retrieval

2004-20 Madelon Evers (Nyenrode)
Learning from Design: facilitating multidisciplinary design teams

====
2005
====

2005-01 Floor Verdenius (UVA)
Methodological Aspects of Designing Induction-Based Applications

2005-02 Erik van der Werf (UM))
AI techniques for the game of Go

2005-03 Franc Grootjen (RUN)
A Pragmatic Approach to the Conceptualisation of Language

2005-04 Nirvana Meratnia (UT)
Towards Database Support for Moving Object data

2005-05 Gabriel Infante-Lopez (UVA)
Two-Level Probabilistic Grammars for Natural Language Parsing

2005-06 Pieter Spronck (UM)
Adaptive Game AI

2005-07 Flavius Frasincar (TUE)
Hypermedia Presentation Generation for Semantic Web Information Systems

2005-08 Richard Vdovjak (TUE)
A Model-driven Approach for Building Distributed Ontology-based Web Applications

2005-09 Jeen Broekstra (VU)
Storage, Querying and Inferencing for Semantic Web Languages

2005-10 Anders Bouwer (UVA)
Explaining Behaviour: Using Qualitative Simulation in Interactive Learning Environments

2005-11 Elth Ogston (VU)
Agent Based Matchmaking and Clustering - A Decentralized Approach to Search

2005-12 Csaba Boer (EUR)
Distributed Simulation in Industry

2005-13 Fred Hamburg (UL)
Een Computermodel voor het Ondersteunen van Euthanasiebeslissingen

2005-14 Borys Omelayenko (VU)
Web-Service configuration on the Semantic Web; Exploring how semantics meets pragmatics

2005-15 Tibor Bosse (VU)
Analysis of the Dynamics of Cognitive Processes

2005-16 Joris Graaumans (UU)
Usability of XML Query Languages

2005-17 Boris Shishkov (TUD)
Software Specification Based on Re-usable Business Components

2005-18 Danielle Sent (UU)
Test-selection strategies for probabilistic networks

2005-19 Michel van Dartel (UM)
Situated Representation

2005-20 Cristina Coteanu (UL)
Cyber Consumer Law, State of the Art and Perspectives

2005-21 Wijnand Derks (UT)
Improving Concurrency and Recovery in Database Systems by Exploiting Application Semantics

====
2006
====

2006-01 Samuil Angelov (TUE)
Foundations of B2B Electronic Contracting

2006-02 Cristina Chisalita (VU)
Contextual issues in the design and use of information technology in organizations

2006-03 Noor Christoph (UVA)
The role of metacognitive skills in learning to solve problems

2006-04 Marta Sabou (VU)
Building Web Service Ontologies

2006-05 Cees Pierik (UU)
Validation Techniques for Object-Oriented Proof Outlines

2006-06 Ziv Baida (VU)
Software-aided Service Bundling - Intelligent Methods & Tools for Graphical Service Modeling

2006-07 Marko Smiljanic (UT)
XML schema matching -- balancing efficiency and effectiveness by means of clustering

2006-08 Eelco Herder (UT)
Forward, Back and Home Again - Analyzing User Behavior on the Web

2006-09 Mohamed Wahdan (UM)
Automatic Formulation of the Auditor's Opinion

2006-10 Ronny Siebes (VU)
Semantic Routing in Peer-to-Peer Systems

2006-11 Joeri van Ruth (UT)
Flattening Queries over Nested Data Types

2006-12 Bert Bongers (VU)
Interactivation - Towards an e-cology of people, our technological environment, and the arts

2006-13 Henk-Jan Lebbink (UU)
Dialogue and Decision Games for Information Exchanging Agents

2006-14 Johan Hoorn (VU)
Software Requirements: Update, Upgrade, Redesign - towards a Theory of Requirements Change

2006-15 Rainer Malik (UU)
CONAN: Text Mining in the Biomedical Domain

2006-16 Carsten Riggelsen (UU)
Approximation Methods for Efficient Learning of Bayesian Networks

2006-17 Stacey Nagata (UU)
User Assistance for Multitasking with Interruptions on a Mobile Device

2006-18 Valentin Zhizhkun (UVA)
Graph transformation for Natural Language Processing

2006-19 Birna van Riemsdijk (UU)
Cognitive Agent Programming: A Semantic Approach

2006-20 Marina Velikova (UvT)
Monotone models for prediction in data mining

2006-21 Bas van Gils (RUN)
Aptness on the Web

2006-22 Paul de Vrieze (RUN)
Fundaments of Adaptive Personalisation

2006-23 Ion Juvina (UU)
Development of Cognitive Model for Navigating on the Web

2006-24 Laura Hollink (VU)
Semantic Annotation for Retrieval of Visual Resources

2006-25 Madalina Drugan (UU)
Conditional log-likelihood MDL and Evolutionary MCMC

2006-26 Vojkan Mihajlovic (UT)
Score Region Algebra: A Flexible Framework for Structured Information Retrieval

2006-27 Stefano Bocconi (CWI)
Vox Populi: generating video documentaries from semantically annotated media repositories

2006-28 Borkur Sigurbjornsson (UVA)
Focused Information Access using XML Element Retrieval

====
2007
====

2007-01 Kees Leune (UvT)
Access Control and Service-Oriented Architectures

2007-02 Wouter Teepe (RUG)
Reconciling Information Exchange and Confidentiality: A Formal Approach

2007-03 Peter Mika (VU)
Social Networks and the Semantic Web

2007-04 Jurriaan van Diggelen (UU)
Achieving Semantic Interoperability in Multi-agent Systems: a dialogue-based approach

2007-05 Bart Schermer (UL)
Software Agents, Surveillance, and the Right to Privacy: a Legislative Framework for Agent-enabled Surveillance

2007-06 Gilad Mishne (UVA)
Applied Text Analytics for Blogs

2007-07 Natasa Jovanovic' (UT)
To Whom It May Concern - Addressee Identification in Face-to-Face Meetings

2007-08 Mark Hoogendoorn (VU)
Modeling of Change in Multi-Agent Organizations

2007-09 David Mobach (VU)
Agent-Based Mediated Service Negotiation

2007-10 Huib Aldewereld (UU)
Autonomy vs. Conformity: an Institutional Perspective on Norms and Protocols

2007-11 Natalia Stash (TUE)
Incorporating Cognitive/Learning Styles in a General-Purpose Adaptive Hypermedia System

2007-12 Marcel van Gerven (RUN)
Bayesian Networks for Clinical Decision Support: A Rational Approach to Dynamic Decision-Making under Uncertainty

2007-13 Rutger Rienks (UT)
Meetings in Smart Environments; Implications of Progressing Technology

2007-14 Niek Bergboer (UM)
Context-Based Image Analysis

2007-15 Joyca Lacroix (UM)
NIM: a Situated Computational Memory Model

2007-16 Davide Grossi (UU)
Designing Invisible Handcuffs. Formal investigations in Institutions and Organizations for Multi-agent Systems

2007-17 Theodore Charitos (UU)
Reasoning with Dynamic Networks in Practice

2007-18 Bart Orriens (UvT)
On the development an management of adaptive business collaborations

2007-19 David Levy (UM)
Intimate relationships with artificial partners

2007-20 Slinger Jansen (UU)
Customer Configuration Updating in a Software Supply Network

2007-21 Karianne Vermaas (UU)
Fast diffusion and broadening use: A research on residential adoption and usage of broadband internet in the Netherlands between 2001 and 2005

2007-22 Zlatko Zlatev (UT)
Goal-oriented design of value and process models from patterns

2007-23 Peter Barna (TUE)
Specification of Application Logic in Web Information Systems

2007-24 Georgina Ramírez Camps (CWI)
Structural Features in XML Retrieval

2007-25 Joost Schalken (VU)
Empirical Investigations in Software Process Improvement

====
2008
====

2008-01 Katalin Boer-Sorbán (EUR)
Agent-Based Simulation of Financial Markets: A modular, continuous-time approach

2008-02 Alexei Sharpanskykh (VU)
On Computer-Aided Methods for Modeling and Analysis of Organizations

2008-03 Vera Hollink (UVA)
Optimizing hierarchical menus: a usage-based approach

2008-04 Ander de Keijzer (UT)
Management of Uncertain Data - towards unattended integration

2008-05 Bela Mutschler (UT)
Modeling and simulating causal dependencies on process-aware information systems from a cost perspective

2008-06 Arjen Hommersom (RUN)
On the Application of Formal Methods to Clinical Guidelines, an Artificial Intelligence Perspective