DISSERTATION

e-ULE

e-Usable Learning Environment

Reconciling student needs and lecturer requirements

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Doktors der technischen Wissenschaften unter der Leitung von

Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Thomas Grechenig
E 183
Institut für Rechnergestützte Automation

eingereicht an der Technischen Universität Wien
Fakultät für Technische Naturwissenschaften und Informatik
von

Dipl. Ing. Larissa Naber
e9125588
1070 Wien, Kaiserstraße 101/1/8

Wien, am 16. Juli 2003

Larissa Naber

Kurzfassung

e-ULE – *e*-Usable Learning Environment

Vereinbarkeit der Bedürfnisse von Studenten und Lektoren

Diese Arbeit stellt *e*-ULE (*e*-Usable Learning Environment), eine universitäre Lehr- und Lernumgebung vor.

e-ULE berücksichtigt die Bedürfnisse von Studenten und Lektoren gleichermaßsen. Dabei sollen die Studierenden mit elektronischen Lernmaterialien versorgt werden, die reale Vorteile gegenüber der Ausbildung mit konventionellen Methoden bieten. Die Erstellung dieser Materialien darf natürlich nicht zu Lasten der Lektoren gehen. Idealerweise sollte ein *e*-ULE unterstüzter Arbeitsablauf dem Vortragenden mehr Zeit verschaffen – Zeit, die für Forschung oder Diskussion mit Studierenden genutzt werden kann.

Einfache Handhabung steht bei *e*-ULE im Vordergrund. Allerdings darf die Benutzerfreundlichkeit nicht zu Lasten der Funktionalität gehen. Deshalb stützt sich *e*-ULE auf folgende Paradigmen:

- Fokus auf universitäre Lehre
- Unterstützung für den gesamten akademischen Arbeitsablauf
- Geeignet für (fast) alle Studiengebiete
- Geeignet für Nicht-Informatiker
- Geeignet für Nicht-Pädagogen
- Unterstützung für alle studentischen Lernphasen

- Unterstützung für verschiedene studentische Lernstile
- Unterstützung für Studenten mit Behinderungen
- Freie (Open-Source) Software

Die konsequente Verfolgung dieser Paradigmen unterscheidet *e*-ULE von anderen Lernumgebungen. *e*-ULE richtet sich ausschließlich an ein universitäres Umfeld und ist weder für die Erstellung von Kindergarten-Lernsoftware noch für Bedienungsanleitungen für Wendeltreppeneinbau geeignet. Die umfassende Arbeitsablaufunterstützung deckt alle Bereiche des wissenschaftlichen Unterrichts von Literatursuche, über Inhaltserstellung bis zu Lehrveranstaltungsverwaltung ab. *e*-ULE ist nicht auf ein schmales Feld eingeschränkt, sondern soll einem möglichst breiten Publikum wie. z.B. Mathematikern, Theologen oder Veterinärmedizinern zugute kommen. Aus diesem Grund sind keine besonderen Datenverarbeitungskenntnisse vorausgesetzt. Um Personen mit geringen Vorkenntnissen in Didaktik entgegenzukommen, bietet *e*-ULE auch Unterstützung beim Entwurf neuer Lehrveranstaltungen.

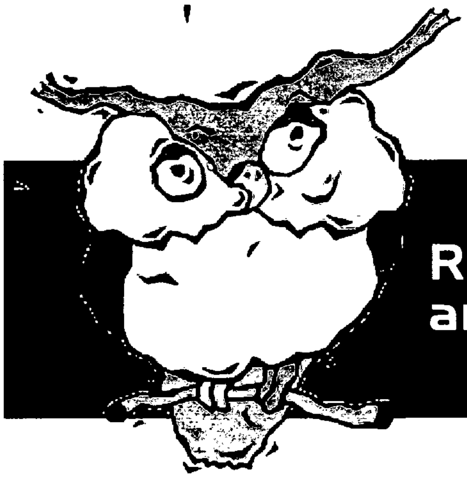
e-ULE benutzt das WWW nicht als ein weiteres Transportmedium zum Verbreiten konventioneller Lernmaterialien, sondern stellt Materialien bereit, welche die Studierenden in allen Phasen des Lernens von der ersten Orientierung, über die Prüfungsvorbereitung bis zum Nachschlagen unterstützen. Die Inhalte eines *e*-ULE Projekts können auch in einer Form angeboten werden, die für Studierende mit visuellen oder motorischen Einschränkungen geeignet ist.

Da die Kostenfrage für österreichische Universitäten von entscheidender Bedeutung ist, ist *e*-ULE als freie (opensource) Software angelegt, genaugenommen repräsentiert *e*-ULE ein meta-opensource Projekt, weil es auf einer vielzahl bekannter opensource Software, wie z.B. Apache, Mozilla und Cocoon aufbaut.

Teile dieser Arbeit wurden in (Naber and Köhle, 2003), (Naber and Köhle, 2002b) und (Naber and Köhle, 2002a) veröffentlicht.

ϵ-ULE

ϵ - Usable Learning Environment



Reconciling student needs and lecturer requirements

Dipl. Ing. Larissa Naber

Wien am 16. 7. 2003

Contents

List of Figures

List of Tables

Foreword – *e*-ULE in a nutshell

Everything should be made as simple as possible, but not simpler.
Albert Einstein

This work proposes *e*-ULE (*e*-Usable Learning Environment), a system for the creation of *e*-nhanced learning materials at university level.

e-ULE aims to reconcile student and lecturer needs and to provide the former with real value, not found in conventional learning materials without placing more burden on the latter. Ideally *e*-ULE's workflow support should actually free up time for the lecturer – time that could be spent on research or on discussion with students.

Simplicity in tools is important, but functionality must not be sacrificed to simplicity. Thus the four main paradigms of *e*-ULE are :

- Focus on university level material
- Support for academic workflow
- Usable in most (all?) scientific fields
- Usable by computer non-experts
- Usable by non-educationalists
- Support for various student learning stages
- Support for different student learning styles
- Support for handicapped student needs
- Free software

These set *e*-ULE apart from other *e*-learning environments. *e*-ULE is only geared towards university level, you can neither build kindergarten applications nor training courses to fend off Alzheimer's disease. The comprehensive workflow support covers all aspects of academic courses from literature research, over content creation to course administration. *e*-ULE is not limited to a small scientific field, mathematicians should benefit as well as theologians or veterinarians. As the systems shall be used by the widest possible range of lecturers, nothing more than basic computing skills are necessary to use it. To help those with limited educational expertise, *e*-ULE will also give hints on course design.

On the students side *e*-ULE does not use the WWW as yet another way of transporting word processor files, but will provide the student with materials that are useful in all learning stages from orientation, over exam preparation to reference. *e*-ULE also provides content in a format suitable to students with visual or motoric impairment.

As costs – or more precisely the lack thereof – are important to Austrian universities *e*-ULE is built entirely from free and opensourced software, in fact it is a meta opensource project. One aspect of *e*-ULE is to show how far one can reach by using existing opensource building blocks. In this place I would like to thank the following organisations and their members for their excellent work:

- Apache Software Foundation
- GNU Project
- Mozilla Organisation

My thanks also go to other individuals involved in opensource software development.

Parts of this work have been published under (Naber and Köhle, 2003), (Naber and Köhle, 2002b) and (Naber and Köhle, 2002a).

How to read this thesis

Any sufficiently complex material worthy of academic study tends to defy the hierarchical structure imposed by the medium thesis. *e*-ULE is no exception. Depending on the readers role (student,lecturer,technician) and inclination several pathes are possible. To lighten the blight of the reader this thesis has been equipped with hypertext and multipath features.

For students

For lecturers

For technicians

Why?

How?

Chapter 1

Introduction

Teaching is not a lost art, but the regard for it is a lost tradition.
Jacques Barzun

1.1 Austrian academic situation

All is not well in the Austrian academic world. Never designed as educational institutions, but research centres involved in the teaching of the next generation researchers, universities now bear the burden of educating the masses for the industry.

Austrian universities are doing a good job in producing highly qualified, independent, selfreliant graduates. The system may or may not be cost-effective, the graduates may not be needed in Austria, but the fact that the most qualified leave the country proofs that there is market for these graduates.

However, the ongoing changes in the educational field increasingly trapped universities between a rising number of students and an ever diminishing budget. This is not about to change in the next years. Still, the industry continuous to pressures the universities to "produce" more market ready students in less time, thus forcing them to teach more topics in a reduced time frame.

There are three possible ways to change this unsatisfying status quo

1. Change university education

2. Change the industry

3. Produce alternative ways of higher education

Unfortunately the latter two belong to realm of high politics. Only "changing university teaching" (partly) lies in the hands of the universities. To change university teaching, one has first to assess what university teaching comprises, and what is wrong with it.

As Barzun already pointed out in the 1950ies, the regard for teaching is a lost tradition. University teaching is necessary to attract students, but frequently regarded as a burden. One wonders if Page Smith has point when he claims that

> It can be said unequivocally that good teaching is far more complex, difficult, and demanding than mediocre research, which may explain why professors try so hard to avoid it.

Moreover, lecturers who take up teaching are punished, as the system does not reward teaching as an academic pursuit. Every minute spent on teaching and the creation of teaching materials is a minute not spent on research, thus leading to a reduced paper count which in turn might have a negative effect on contract extension or promotion. The disregard for university teaching is not a typical Austrian or even European problem the same is true for the USA (Felder, 2003).

The growing number of "Fachochschulen", which provide more practically orientated education and lay more emphasis on quality of teaching, will force universities to rethink their concepts of teaching. This will take time. In the meantime the universities are faced with a more serious and immediate problem: with the reintroduction of study fees beginning with the winter term 2001, the universities will be forced to reconsider the position of students in the new light of "paying customer". Moreover, they have to face an ever increasing number of working students, who do not fit the typical student profile.

Not surprisingly universities have recently been looking for alternative ways of information transfer to replace the classical lecture. Without much consideration for teacher or student needs the universities see *e*-learning as the solution of choice, although there seems to be no clear understanding of what *e*-learning is to comprise.

It is about time to separate effect from cause, to ascertain the real problem, to ask an inconvenient question:

1.1.1 If *e*-learning is the answer, what was the problem?

Problem 1 – Information transfer

When we look at lectures at Austrian universities we find some that are so overrun that students will actually fight for a place in the lecture hall. On the other hand we see lectures where only about 3 to 4 % of the students enrolled actually attend the lecture.

Obviously this is a real problem: neither situation is well suited for passing on information. In the first case the lecture is reduced to the true meaning of the word[1]: the lecturer passes information to his students by reading the information in question aloud. A lecture hall packed with a thousand students perched on every flat (or not so

[1]Middle English, act of reading, from late Latin lectura, from Latin lectus, past participle of legere, a discourse given before an audience or class especially for instruction.

flat) surface surely is not a place to inspire a lively discussion of the latest development in this or that scientific field. In the second case one might argue that it is a waste of resources to even give a lecture that is attended by only 3% of the students. The intrinsics of information transfer by lecture are detailed in section 2.1.

Technology can solve part of this problem. The Internet can help in distributing information. However, an *e*-nhanced learning environment can only be successful if it fulfils students' and lecturers' needs alike. The student needs to be supported in various stages of learning, whereas the lecturer can't afford to spent more time on generating lecture support materials. A tool is needed that helps students without additionally burdening the lecturer.

Problem 2 – Regard

As pointed out above, lecturing is not well esteemed in the scientific community. Nobody cares if you let matters slide while doing complex research, while holding back on research to provide first rate education is considered harmful for your reputation.

Technology can not solve this problem, only reform could do that. In the meantime, technology can lighten the blight of the stressed out lecturer.

The learning environment authoring system must be geared towards a typical lecturer, requiring no undue amount of IT or pedagogical skills. It should support the academic workflow by catering to tasks like literature research and integration, and collaborative editing in large groups (e.g. together with students).

Problem 3 – Time is money?

University budgets are always tight. Only a small number of institutes can raise money from companies. Most have to rely on the regular budget and research grants. Again the need to research to get another research grant interferes with the need to teach.

Even if the *e*-learning software is free, the creation of *e*-courses or web based lectures is very time (= money) consuming and requires considerable skills on part of the creator. When looking at major players in the university field (compare: MIT is putting all their lectures online (MIT, 2003)) one realises that is multi-million euro / dollar effort, involving hundreds of professionals.

So, should *e*-learning be left to the big players, or can small universities or even individual lecturers *e*-nhance their lectures without driving themselves mad or swamping their students in well meant, but useless *e*-materials? (See (Mitchell et al., 2001), (Montelpare and Williams, 2000) and (Watson, 2001) for first hand accounts on the problems of introducing *e*-learning).

This can be solved by technology: in compliance with the meagre university budgets an *e*-learning system should be open sourced and therefore cost-free. Relying on already available opensource projects can significantly reduce production time and cost. Ease-of-use and a well defined feature set selected especially for the university environment can help reduce production time. Ideally, the time to generate *e*-nhanced materials should be equal or less to the time spent on creating conventional lecture notes/materials. Time (and cost cutting) methods include: students work, collaboration and recycling.

1.2 Lessons to be learned from abroad

Looking overseas, one can see that things are quite different there – different, not necessarily better! The fact that the US are a net importer of academics proofs that their system is not efficient. The biggest influence on university teaching is posed by study fees. The fees to be paid for university education in the USA are considerable.

US universities sport media departments that will help with the creation of multimedia content, teaching excellence centres that will help in whipping up a rock solid curriculum or in enhancing the rhetoric skills of the lecturers. This trend was started by the ivy-league establishments, but is now taking over in less renown colleges as well. All these quality of service assurance methods are comparatively fresh additions to US universities mostly linked to the ever rising study fees. To benefit from this "university teaching on steroids" you either have to be filthy rich, a sport ace or sell your soul to your bank. Very few students benefit from a scholarship. Basically, quality university teaching is available – for a price.

But study fees do not only have social consequences. The need to keep students paying does lead to a lowering of standards. If too many students fail a test, the standards will be lowered to avoid an overly loss of income. Thus education is for sale and the US educational system is in constant decline. Up to now the US economy could compensate by importing qualified personnel from all over the world. Now the situation becomes critical as major export countries like India or China are clamping down on this educational closing sale. Even Europe has finally seen the need to keep qualified graduates inside Europe. America is headed for the educational "Big Bang".

It is clear that the American way is not the way to go. Austria will have to find its own way. There is no way around the basic fact that education is important and expensive. Moreover, somebody got to pay. The only question is how the money is best spent? Reforms are necessary to give us quality research and quality university teaching. But in the meantime we might take a leave from the American book and help the university lecturer in his work, otherwise Richard Felder's quote

> College teaching may be the only skilled profession for which no preparation or training is provided or required.(Felder, 2003)

might be all too true.

1.3 Summary

Yes, *e*-learning can make an impact on Austrian higher education, but it can not solve all of our problems. Properly introduced *e*-learning can free up personnel resources and increase student motivation, but any deployed solution must help students and lecturers alike. A system that burdens the lecturer will lead to diminished research and thus a decline in reputation of the university, whereas a system that does not actively help the students will not be perceived as benefit by the students, but as a very bad return of investment. Losing the students' support will mean a decrease of support for the university in the whole population and another cut in funding.

Chapter 2

e-nhancing university teaching

> **A lecture is a process by which the notes of the professor become the notes of the students without passing through the minds of either.**
>
> *R. K. Rathburn*

This chapter deals with *e*-nhanced university teaching, that is, all methods which rely on electronic materials as way a of information distribution. It takes a look at the omnipresent lecture format of todays university courses and the problems associated with them and other possible ways of information transfer. It further examines the reasons for skipping lectures and the student views on the "ideal course". The chapter concludes with taxonomy of *e*-learning systems and tries to answer the question if *e*-learning is useful or not.

2.1 Lectures – an outdated way of information transfer?

Despite the technological blessings of the the 20th and 21st century (Xerox machines, Internet, video, ...) much of our university systems is not stuck in the middle ages (which would be preferably as it offered small groups and lively discussion) but in the late 19th and early 20th century, when higher education first became available to a larger group of people.

The lecture format with its historic roots in the public reading of rare texts has evolved to the primary means of university teaching. Increasing student numbers result in more students per course and thus enforce the use of even more lecture format courses (Garrison and Anderson, 2000).

Using lecture as primary way of information transfer is tedious for both parts: Lecturers have to cover nearly the same ground year after year, and students suffer from the often less than perfect pedagogical and rhetoric skills (One cannot necessarily expect a top scientist to have first rate rhetoric or pedagogical skills as well). While it is true, that a well designed lecture is beneficial to the students' progress, many students claim that their lecturer lacked rhetoric and didactic skills, rendering this aspect of lectures virtually useless.

A large amount of lecture centred courses can also discriminate against working or handicapped students, who might find it difficult to attend. Students unapt to learn by listening often find lectures cumbersome.

A predominant use of the lecture format causes valuable resources (time | space | personnel) to be given up to the sheer act of information transfer, information that in many cases could be transferred in a more efficient manner, thus freeing up resources for more in-depth discussion and exploration of the topics.

University teaching is basically about getting students to ask the right questions, frame problems, and find solutions themselves, which is rarely achieved by lecture format courses. Along the lines of Tsichritzis (Tsichritzis, 1999), I believe that university teaching in the "post xerox age" should concentrate less on information transfer, but on aiding the student in the process of knowledge [1] acquisition.

To facilitate the creation of knowledge a shift from the lecture format is necessary. This shift could be well supported by digital media and *e*-learning provided that they offer students more than conventional lectures without additionally burdening the lecturer. A shift to digital media, however should not be confused with so called *e*-learning in the form of downloadable Powerpoint™presentations plus moderated newsgroups which are the latest hype, but of little benefit to the students.

In a recent study (Naber, 2002b) we found that the students want more "quality time" with their teachers and support in answering specific questions, or the exploration of advanced topics. The teacher shall not be replaced by an *e*-learning environment, but in *e*-nhancing the lecture by shifting the main burden of information transfer to other – digital – media, lecture time is freed to discussions and question & answer sessions. The results are summarised in section 2.2.

2.2 Survey of lecture non-attendance

One of the first steps in the design of our *e*-learning environment was to look at the puzzle of lecture attendance. Lecture attendance vastly differs with university. While many students hardly see a lecture hall from the inside, others (e.g. students of medicine) get up at 5 o'clock in the morning to get place in the lecture hall. In fall 2002 the vice chancellor of the notoriously overcrowded Vienna University of Economics and Business Administration made the news by renting a cinema to solve the lack of lecture halls.

We made a survey of lecture attendance among students and interviewed them

[1] Ackoff's 4 step ladder (Ackoff, 1981) of data - information - knowledge - wisdom.

about their ideas about an *e*-learning environment. The survey was done via the web, the interviews were done on a one-to-one basis.

2.2.1 Case study design and methodology

The web survey was prompted by the radically different percentages in lecture attendance among the different fields of study. A quick survey among friends and colleagues found that asking 'Why do you attend the lecture' is a rather moot question, not likely to elicit some meaningful response.

So in the actual survey we asked 'Why did you avoid certain lectures'. Obviously students tend to have rather strong thoughts about lectures they did not attend, which lodges them into memory.

The participants were selected to be representative for the average Austrian student, (different universities, programs, gender, age, social and cultural background, progress in their studies, full-time/part-time).

Of the harvested questionnaires 68 were statistically relevant. Nine of the students who filled out the questionnaire were interviewed further. Later interviews on *e*-learning also included questions about lecture attendance, but were in line with this finding.

2.2.2 The questionnaire

The questionnaire focused on

- compulsory courses (no elective courses)
- without compulsory attendance
- which stressed the lecture aspect (no labs, workshops, seminars ...)
- which were attended less than 30% of the time

Concerning the field of study we settled for a distinction between technical / natural science studies ("technical") and arts / humanities ("non-technical") as we were unlikely to find a sufficient number of participants to do an evaluation based on the field of study.

Furthermore we decided to distinguish between undergraduate and graduate students.

We asked about the estimated number of skipped lectures

- less than 20 %
- 20 - 40 %
- 40 - 60 %

- 60 - 80 %
- more than 80 %

and settled for an attendance rate of <40% as "few" and >40% as "many"

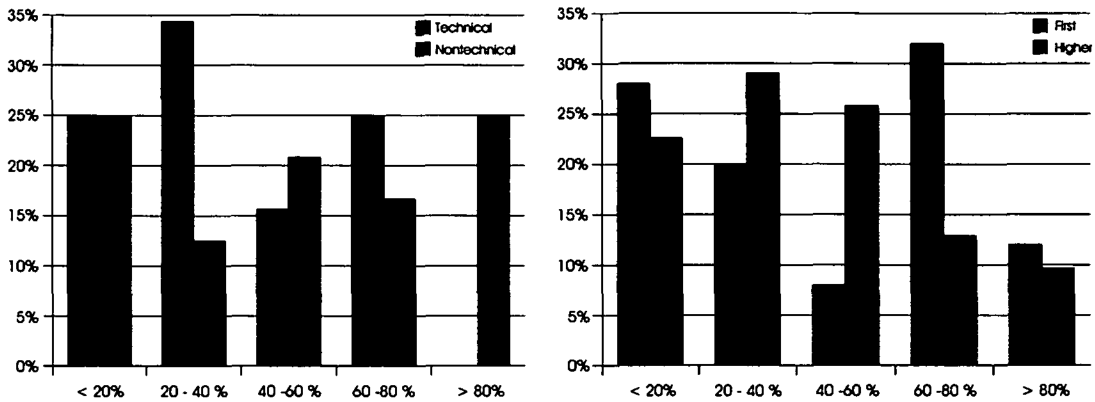


Figure 2.1: Lecture attendance for freshmen versus senior students and technical versus non-technical students.

Then we asked for reasons for not attending a lecture. These reasons we divided into 6 groups:

1. lecturer (L)
2. time (T)
3. contents (C)
4. personal learning style (S)
5. handicaps (H)
6. free-form entry of other reasons (F)

At last we asked to name the main reason, the group that is most likely to cause non-attendance.

2.2.3 Hypotheses

The questionnaire was based on several hypotheses which we will document in this section.

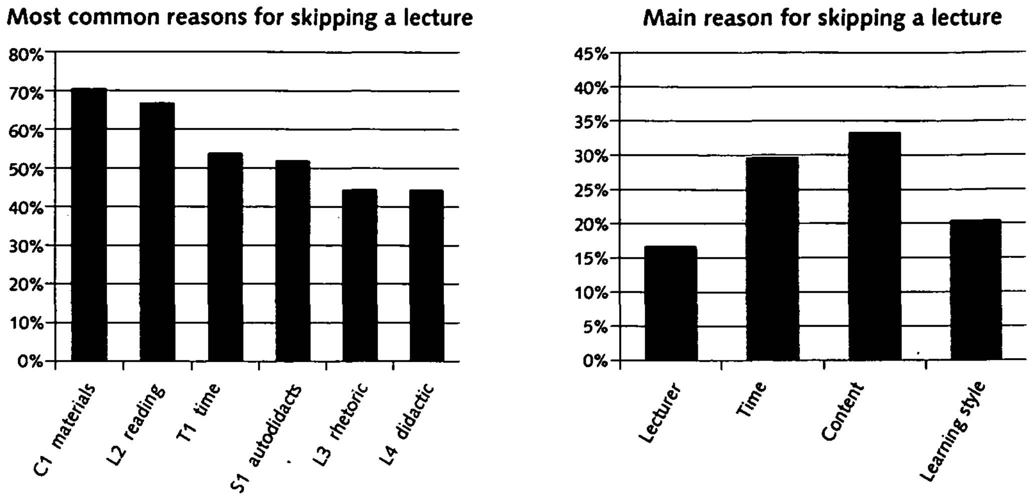


Figure 2.2: The six most frequent reasons for not attending a lecture – the remaining 14 reasons occurred in less than 25% of the responses and the single most likely reason for not attending for all participants.

Lecture notes

hypothesis: Students avoid a lecture if lecture notes are provided.

outcome: confirmed

rationale: Many lectures are a waste of time: the lecturer lacks rhetoric or pedagogic skills and little is to be gleaned from actually attending the lecture. If study material is available, the lecture is likely to be skipped. "Materials are available" was the overall most named reason for avoiding a lecture. It made first or second place in all different categories (undergraduate/graduate, many/few or technical/nontechnical). "Materials are available" also was the overall main reason for avoiding a lecture.

Lecturer

hypothesis: Students avoid a lecture if the lecturer lacks rhetoric, pedagogical or organisational skills.

outcome: confirmed

rationale: Even though "Materials are available" made first place, 3 of the six reasons in the category *lecturer* made it into the top five. However nobody claimed the lecturer as the main reason to skip lectures.

Learning styles

hypothesis: Those who attended few (less than 40%) lectures are likely to prefer learning alone / from books.

outcome: confirmed

rationale: Students who do not profit from listening to lecturers are not likely to attend any. Students who attended few lectures were the only ones to claim "personal learning style" as the main reason for their absence.

Field of study

hypothesis: Technical students tend to avoid lectures

outcome: confirmed

rationale: Technical students are assumed to be autodidacts and do prefer learning from books. 60% of the technical students attended few (less than 40%) lectures. The ratio is reversed for non-technical students, whereas undergraduate or graduate has no effect on the number of lectures attended. Also compare next subsection.

Age/experience

hypothesis: Undergraduates are more inclined to attend lectures

outcome: not confirmed

rationale: Undergraduates were assumed "to do as they are told". The ratio for few (less than 40%) to many (more than 40%) is roughly 50:50 (as it is globally). In fact undergraduates are even less enthusiastic than graduates: nearly 30% attended less than 20% of the lectures. Also compare the previous subsection.

2.2.4 Reasons for not attending lectures

The questionnaire contained 20 questions arranged in six groups. After listing the respective questions for each group, we give an evaluation of the responses for the group. For a graphical comparison of the reasons see figures 2.3 and 2.4.

Group 1: lecturer (L)

The Group *Lecturer* comprised 6 reasons pertaining to the rhetoric, presentation and pedagogic skills of the lecturer.

L1 lectures too fast

L2 follows the word *lecture* to the extreme and reads the book out loud

L3 lacks rhetoric skills

L4 lacks planning/organisational skills

L5 can't get to the point

L6 doesn't interact with students (questions, discussions)

Although the single most named reason for not attending was "materials are available" the sum of complaints about the lecturer made up for nearly 40% of all complaints. However, no group claimed the lecturer as the main reason. Technical students are less tolerant regarding the quality of a lecture as their non-technical colleagues, the same goes for undergraduates compared to graduate students.

Group 2: timing (T)

This group contained the following reasons:

T1 lecture interferes with other lectures

T2 lecture is a single event in certain time frame

T3 long commuting necessary

T4 lecture interferes with job

T5 lecture interferes with private life

The time group is dominated by "Lectures interfere with other lectures". Interference with job or private life came up nearly even, but not so pronounced as interference with other lectures. Reasons T2 and T3 were hardly named. Time was the most named *main reason* for the group *graduate students.* This does not surprise as the number of lecture collisions grows with every term. Time also was the most named reason for the groups "non-technical" and "graduate" students.

Group 3: content of lecture (C)

This group contained the following reasons:

C1 Sufficient material is available (lecture notes, books, etc.)

C2 Content is not relevant for the exam

C3 Topic of the lecture and content of the lecture have nothing in common

"Sufficient material available" utterly dominated this section. The availability of study material obviously is crucial to lecture attendance. No matter how inconvenient or annoying a lecture: if there's no material available it will be attended. The content group made it main reason in the overall group, as well as in technical students, undergraduates and those who attended many lectures.

Group 4: individual learning style (S)

This group contained the following reasons:

S1 I'm an autodidact, I prefer learning on my own

S2 I prefer learning in small groups together with other students

S3 Listening to a lecture doesn't help, I prefer learning from written material.

The learning style group was dominated by reason S1 (autodidact), backed by S3 (learning from books). Learning style was nominated as the main reason by those who attended few lectures (jointly with time).

Group 5: handicapped students (H)

This group contained the following reasons:

H1 I can't follow the lecture (hearing, seeing, ...)

H2 It's to difficult to get to the university

H3 It's not possible to access the lecture hall (lack of elevators, ramps, ...)

The sample was too small a number to allow for a statistics of the main problems for handicapped students. As *e*-learning is often positioned as a benefit for handicapped students, their specific requirements will be canvassed in further interviews.

However, more than 10% of the students claimed to be "unable to follow the lecture" due to the lecturer being too muted, bad/small writing, etc. ...

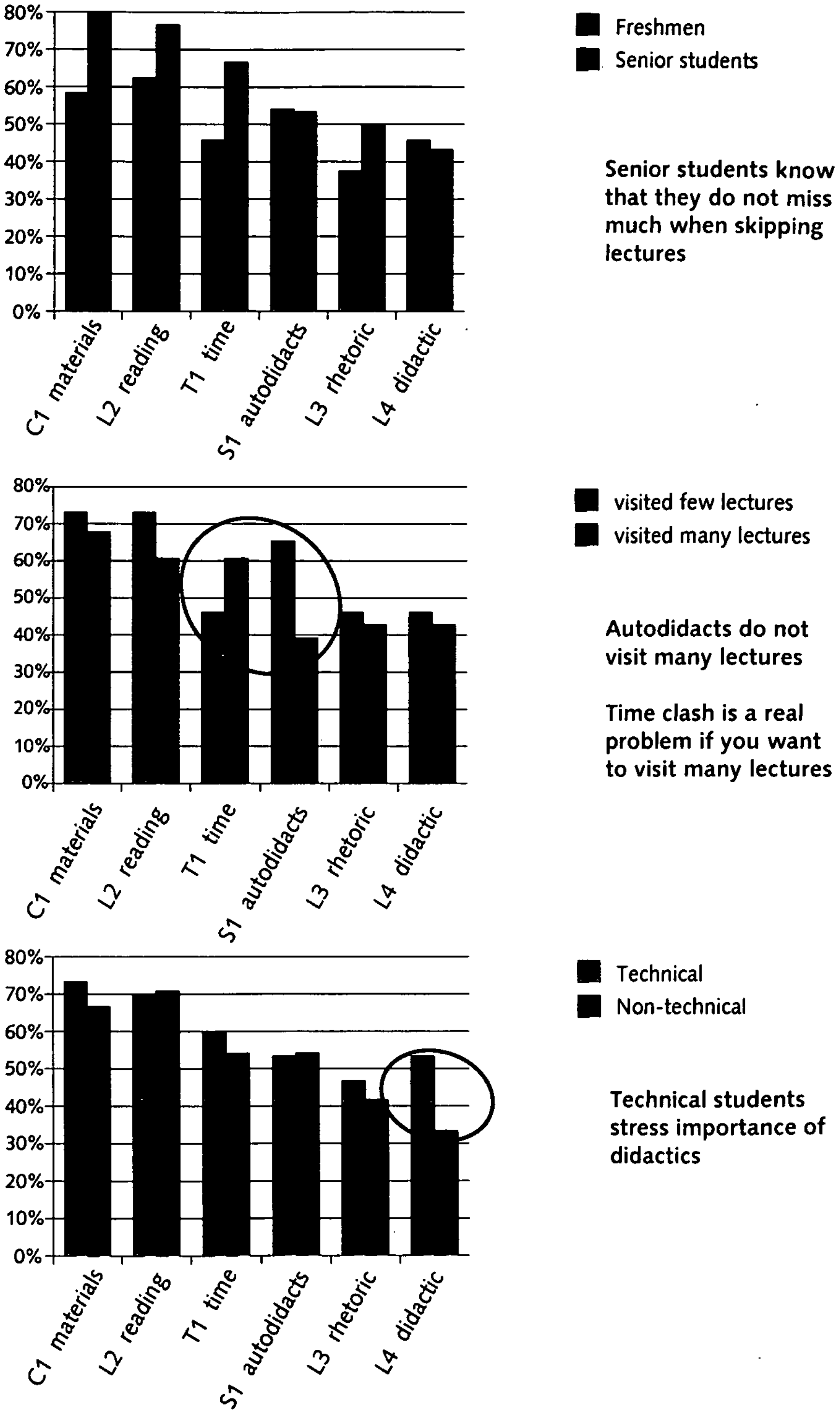
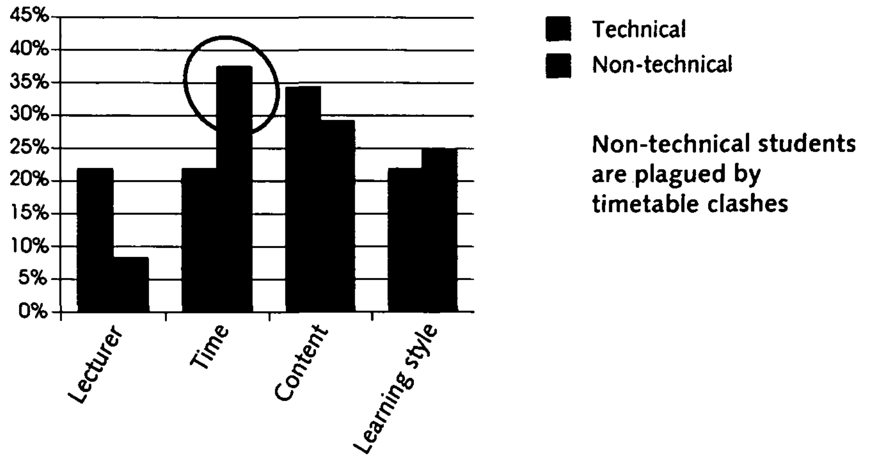
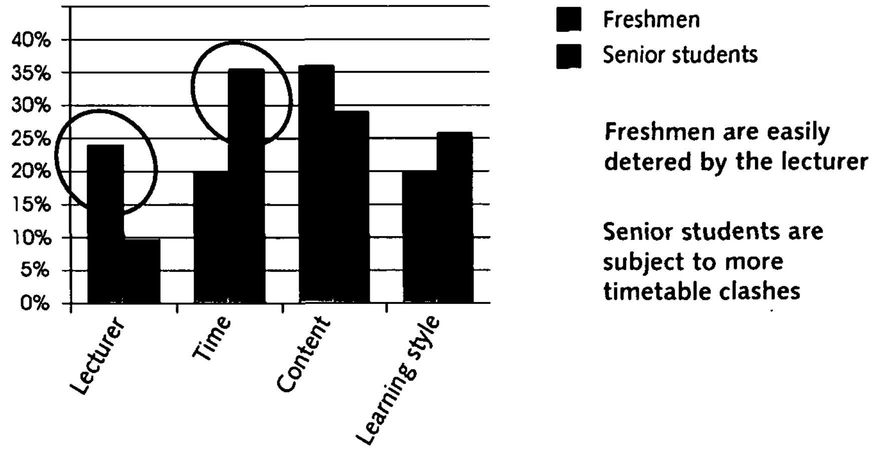


Figure 2.3: The six most frequent reasons for not attending a lecture – the remaining 14 reasons occurred in less than 25% of the responses.





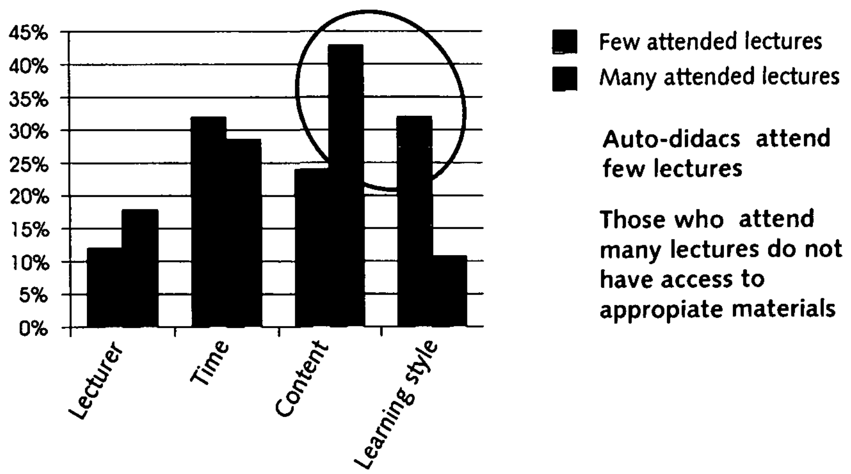


Figure 2.4: The main reasons for not attending a lecture according to the four groups: Lecturer, Time, Content, Learning Style (no one chose handicapped).

Group 6: free-form entry (F)

The free-form entry brought hardly anything new. Most students just put emphasis on one of the other reasons. Several remarked that lectures before 11am should be banned because of cruelty to students. However, three students claimed to be unchallenged and bored by the lecturer who took to long to explain or repeated introductory material.

2.3 What is *e*-learning anyway? – And what is multimedia?

Along with *e*-learning , multimedia can pride itself of being the buzzword of a whole generation of educators. To resolve the eternal puzzle of multimedia one first has to answer the question "What is a medium?". Merriam Webster dictionary defines:

1. Something in a middle position
2. A means of effecting or conveying something
3. A channel or system of communication, information, or entertainment
4. Something (as a magnetic disk) on which information may be stored
5. A condition or environment in which something may function or flourish
6. Material or technical means of artistic expression

In an educational setting one is mostly concerned with the meanings 2,3 and partially 4. Content is represented in various media (content media), which in turn are stored and transported by transport and storage media. The combination of a content and a transport medium does not qualify as multimedia. Only the combination of various content media, like text, pictures, graphs, sound, video or even haptic or olfactory media can be termed multimedia. As hypertext is commonly perceived as the joining of text by hyperlinks, hypermedia therefore is the combination of multimedia and hyperlinks or hypertext with various media.

So can multimedia be an environment in which learning can flourish? There is common superstition stating that multimedia is especially beneficial to learning. For every study claiming a positive effect there is one showing a negative effect and one that could find no effect at all.

Most of the positive effect can be put down to the Hawthorne motivational effect (Mayo, 1933) or novelty/curiosity effects which soon wear off. Another reason for positive correlation is the fact, that time consuming multimedia experiments tend to be carried out by enthusiastic teachers, who would be able to inspire their students anyway.

The negative results are frequently prompted by "media misuse", that is the use of the wrong medium for the content. Not every type of content lends itself to multimedia representation. Classic works of literature provide little that can be enhanced by audio or video. Sure enough, the text can be augmented by photographs of the author or an audio track of a reading from the work or even a video from a theatre production. But all this is worthless if the text is not readable on a monitor or does not come with sufficient (re)search capabilities.

Whereas there is no direct proof that multimedia in fact enhances learning, the use of the right medium for the content is certainly beneficial. The possibility to combine various media in one set will improve chances that the correct medium is used. Not every type of content can be converted to any type of media for representation. The question of choosing the right / the natural medium for the content is complicated by the fact that different people prefer different ways of presentation. However providing the same content in as many ways as possible does not solve the problem. As pointed out by Sumner and Taylor (Sumner and Taylor, 1998) redundant content adds confusion. Redundant information therefore should only be used to ensure accessibility.

Resolving the term e-learning to *electronic learning* one arrives at the conclusion that one has to deal with electronically aided or mediated learning. This would include a language course on compact cassette as well as satellite broadcasted educational television. However, the leading *e* has only became popular with the spread of the Internet and the WWW in special. So when talking about *e*-learning one usually refers to *Internet technology aided learning.* This still leaves the field wide open.

Before the Internet reached every day life, classification of (technology based) learning aids was simple: you could either divide the systems on the time scale, or on a place scale. The time scale saw a division between *synchronous* (lectures, TV/radio broadcast, telephone communication, ...) and asynchronous (video bands, audio discs, CBT, ...) communication. The place scale divided between telemedia that were accessible over a distance (TV/radio broadcasts, telephone communication, ...) and local media which might have been transfered beforehand (CBT, video tapes, audio discs, ...).

The Internet has changed all this. This is the one medium to bridge synchronous and asynchronous media, to be used at a distance, or saved for later local reference, to allow one-to-one or many-to-many communications.

e-learning is often associated with WBT (Web Based Training). This is rather misleading, as the "Web" or WWW is only a small (but growing) part of the Internet. Reducing *e*-learning to WBT would leave out a large number of useful Internet services especially in the communication and broadcasting field.

As pointed out above, the Internet is not a content, but a transport/storage medium. Previous storage and transport media like paper, audio tapes, CDROMs, mail, television or radio are currently being replaced by computer storage and the Internet. Therefore I would like to redefine the term *e*-learning as the *use of the Internet to provide educational content.* The versatility of the medium Internet makes it very difficult to further define the field. A taxonomy of *e*-learning systems is therefore needed (see 2.4).

2.4 Taxonomy of multimedia *e*-learning systems

With the rise of multimedia/hypermedia learning environments numerous taxonomies were developed. Gloor (Gloor and Streitz, 1990) distinguishes four categories based on their *didactic construction*: drill & practice, tutorial, edutainment (learning games) and simulation. Bodendorf (Bodendorf, 1990) focuses on *interaction method* and distinguishes help (learning by hints), passive tutoring (learner controlled), active tutoring (guided learning, program controlled), simulation (discovery learning), games, problem solving and socratic learning (intelligent dialogue). Schulmeister (Schulmeister, 2002) and Ferguson(Ferguson, 1992) propose taxonomies on the basis of learner control. Schulmeister defines 9 classes of increasing *learner control*:

1. Drill & practice
2. Courseware
3. Presentation
4. Kiosk systems
5. Guided Tours
6. Electronic books
7. Hypertext systems
8. Simulations
9. Interactive programs

These classes vaguely correspond to various learning theories described in chapter 3. Figure 2.5 shows major learning theories of the 20th century and multimedia/hypermedia learning tools associated with them. Drill & practice programs test knowledge in a purely question and answer format, whereas Courseware systems include the necessary knowledge blocks to master the exercises. Drill & practise systems as well as courseware are commonly found in language learning tools and K12 (Kindergarten to Grade 12 or graduation) tools.

Presentation, kiosk systems and guided tours focus on the representation of data and the freedom of movement within the content is rather limited. Whereas presentations are mostly used in classroom settings or technology shows, kiosk systems and guided tours are commonly found in museums or on edutainment CDROMs.

Electronic book systems follow the hierarchical book approach while simultaneously integrating various other non text media such as audio or video sources. Hypertext/Hypermedia systems in contrast are network based rather than hierarchically structured and allow to access the materials in various ways.

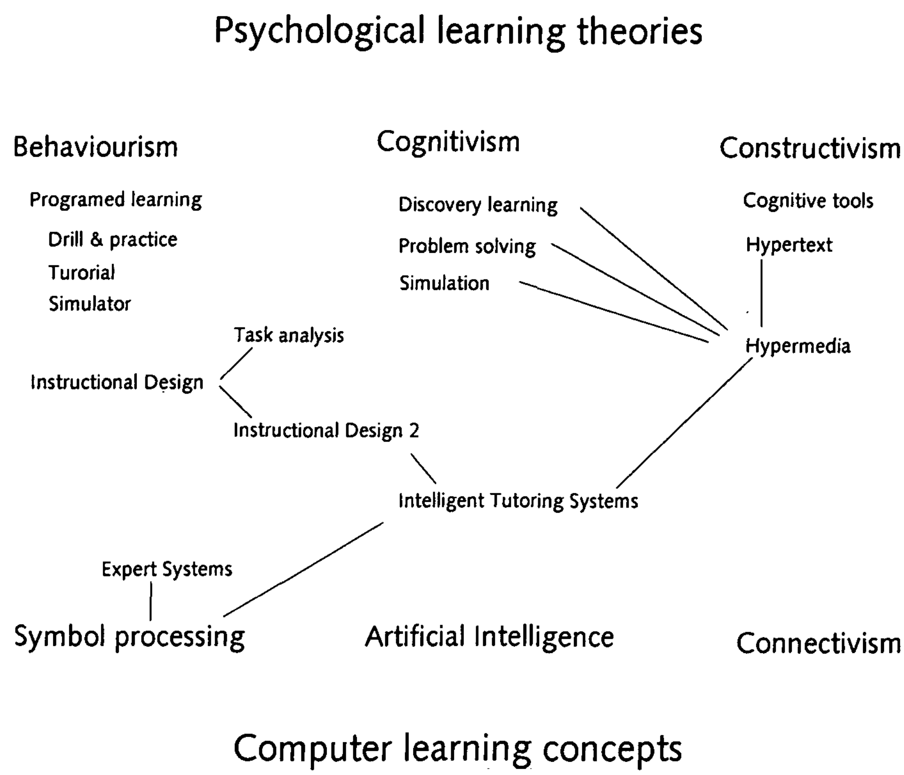


Figure 2.5: Development of multimedia/hypermedia learning environments in relation psychological and computer learning teories.

Simulations allow for explorative learning and are commonly divided into parameterised simulations, which only allow for changes of predefined parameters and microworlds, which can be completely controlled by the learner. Microworlds form the link to interactive systems, which focus on involving the learner into the creation of new content.

The classification of e-learning systems into categories by didactic concept, interaction strategy or learner control bears the danger of over compartising or stereotyping the systems and thereby loosing the big picture. Therefore other categories are presented, which are modelled closer to the educational reality.

- Tool Purpose: generic or specialised?
- Target users (persons generating the materials) / needed skills
- Time frame
- Costs – development costs and TCO (Total Cost of Ownership)

- "Cui bono?" – benefits
- Content centred use of media

2.4.1 Tool Purpose

What is the purpose of the tool? To build a highly specialised learning environment like a simulation on international agrarian marketing a very flexible and powerful tool is necessary, like a programming language. Flexible tools have nearly unlimited possibilities but high skill demands.

On the other hand a hypermedia learning environment is very predictable and very generic. A hypermedia system for atomic physics has very much the same features as one on the influence of Italian writers on the Elizabethans or a system dealing with Cagne's theories on child development. A tool for building educational hypermedia text books therefore can be very narrow featured indeed. Any functionality not provided by this tool can be circumvented by including external references to contents built by specialised systems.

Trying to build tools suitable for generating everything from edutainment applications for kindergarten kids, K12 and university level educational applications to business training is therefore likely to fail. The resulting product will be far too complicated for the casual user, but too restraining for professionals. Unfortunately this jack-of-all-trades approach fascinates developers and buyers alike as it promises "build once sell to all" and "buy one use for every thing". Tool purpose is intrinsically connected to the target user category.

2.4.2 Target users / skills

Learning environments can be grouped by their target audience: *domain experts* – Lecturers or qualified students or *system experts* – people with special training on the tools and domain knowledge in psychology / educational theory like instruction designers, multimedia authors with specialisation on education.

Software addressing systems experts usually offers much functionality / flexibility and is suitable to build highly specialised environments. It usually requires lots of training. Typical software includes multimedia authoring tools (e.g. Macromedia Director) and programming languages used in the process of creating simulations. Software for domain experts ought to be "walk-on-systems", requiring little or no training. These are highly specialised tools which incorporate a certain structure into the resulting learning material. Bad design and usability can push any system in the regime of system experts.

Every learning content has to be produced by somebody. Anybody could produce it, but reality shows, that depending on the university (and field of study) the content is either produced by the lecturers or by the students. Vienna University of Technology and Vienna University of Natural Resources and Applied Life Sciences (BOKU) both have a strong tradition in lecturer produced materials. At the Vienna University materials are mostly produced by students and published by the student union (ÖH). Various departments completely reject the idea of written material and insist on the students attending lectures to write down information .

The main difference between a student and a lecturer author is the fact, that the latter has more experience and quite often a repository of materials to draw from. Nowadays it is quite uncommon to have a secretary to write lecture notes, so this case can be neglected. No Austrian university sports a media or content department that assists in the creation of content. It is therefore safe to assume that all content is written by (aspiring) domain experts. If the author does not happen to be a computer scientist or psychologist or educationalist or have a strong interest and knowledge in these fields he does not qualify as a system expert.

Another possible distinction is the number of authors: *single player* – all content is essentially written by one person – or *teams*. As of now, the vast number of lecture notes are single efforts written by one lecturer/student. This might partly be caused by the fact that todays word processors do not lend themselves to collaborative work [2]. If collaboration takes place it is more likely that chapters are divided among the members of the team leading to even less coherence than usually to be expected.

A more useful collaboration takes place if the domain experts prepare content that is transformed by the system experts. As system experts are usually not found in the university setting, they have to be hired, thus raising the costs of hypermedia development.

2.4.3 Time frame

Time is a frequently unconsidered criterion in the selection of learning environments. One can distinguish *time-consuming*, *time-neutral* and *time-releasing* tools. These are to be put into relation with the time usually spent on lecture preparation. A poll among university teachers showed that lecturers spent 5 to 10 hours per hour lecture for the creation of printed lecture notes and "readers" [3] and 1 to 5 hours when only creating presentation slides.

Time-consuming technologies require more than 10 hours preparation time per hour lecture material and include educational video (other than having a camera running during the lecture), simulations and extensive hypermedia projects. The production of drill & practise exercises and courseware is also very time consuming.

Most hypertext/hypermedia systems (provided they come with a usable authoring tool) fall into the category time-neutral. Instead of using a word processor to generate lecture notes, the lecturer makes use of the authoring tool to provide an electronic version of the notes. As neither word processors nor hypertext/hypermedia authoring

[2] Being able to see who edited what does not make MS Word a collaborative tool. Furthermore these features are rather new and untested and tend to be unstable especially if used on larger documents

[3] A collection of various texts (excerpts) to support the lecture

tools are geared towards academic publishing, there should be hardly any difference in use.

This however does not take learning time for a new tool into account. Depending on the complexity of the tool the learning time can vary from a few hours to several weeks. While a person faced with two or three lectures a year might accept a day learning time for a new tool, several days or even weeks are unacceptable, especially if the usage of the tool is not continuous and most of the functionality has been forgotten at the time of the next deployment. An undue amount of learning time can therefore push a tool from the time-neutral to the time-consuming domain.

A tool specifically geared towards university teaching might in fact hold the status of a time-releasing tool. This can only be reached if the tools can also be used in research and academic publishing thus producing synergies that free up time.

2.4.4 Costs

Wherever computers are involved, cost calculations become difficult. The cost of a learning environment compared to conventional ways of informations transfer as lectures and lecture notes are manifold. The cost can be grouped into three categories: acquisition cost (software + hardware), maintenance cost, time related cost.

First let us determine the cost of the conventional method. The costs consist of the cost for the lecture time and for lecture preparation time plus the cost of technological equipment (mostly a computer and a word processing package). As the technological equipment used is lowtech and mostly covered by the default university gear, it shall not be considered further. Moreover lecture notes can be handwritten. The costs for paper-based information is usually charged to the students and does not enter the equation. The cost for the conventional methods arise mostly from the hours the lecturer spends on the creation of materials.

Considering *e*-nhanced materials, the time spent on the creation of the materials itself (see section above) is not likely to decrease significantly, but rather prone to rise. In addition there is time spent on training for the new tool. The time decrease of actual lecture time is rather insubstantial. Multimedia/Hypermedia production usually requires special software not normally covered in the university pool. Depending on the software chosen (commercial versus open source products) the costs for authoring tools can be substantial ranging into the thousands of Euro. In addition to authoring software most systems require a server to run on and the appropriate server software. *e*-nhanced teaching also procures a lot of maintenance costs. Commercial learning environments frequently offer software support (which usually is a sign that this support is needed) and the server (software) needs to be maintained to assure resistance against attacks.

These cost aspects are not independent from each other. Cheap or free software (e.g. software produced by computer science students as part of their studies) usually has lesser demands on hardware but often results in high maintenance costs, whereas buying an expensive commercial product is no guaranty for little training time or low maintenance costs. Generally it is not possible to minimise all cost aspects individually, but is possible to minimise the TCO, depending on the situation.

A "cheap" solution is certainly preferable, considering all facets, however one can only hope for a costeffective solution. As cost is a major factor in selecting a university learning environment, the TCO is usually the most obscured factor of each product. As the buyers (faculty or university administration) usually are not the users (lecturers), cost-effectiveness calculation omits preparation time and training (aside commercial training out of house) because the lecturer is employed anyway. The time spent on the tool of course subtracts from the time spent on research leading to less publications, which can prove harmful for the lecturer if his contract is up. These non-monetary costs are hard to capture in TCO calculations.

Category		Conventional	*e*-nhanced
Acquisition	Software	– (standard software)	Authoring software
			Server Software
	Hardware	– (desktop computer)	Server (extra machine)
			(desktop computer)
Maintenance	Software	–	Updates
			Software support
	Hardware	–	Server administrator
			Internet traffic
Time related	Preparation	1 - 10 h/ h lecture	Likely to increase
	Lecture	1 h	Can decrease
	Training	–	From several hours onward

Table 2.1: Cost for traditional and *e*-nhanced methods in university teaching. It is assumed that every lecturer has access to a typical office computer (which he has at least basic knowledge of), which frequently is university provided and does not incur further cost for the department/institute.

Table 2.1 contrasts the costs for conventional and *e*-nhanced teaching. Considering the monetary overhead of *e*-nhanced instruction, one is surprised by the frequency of *e*-learning projects. To justify the expenses it is necessary to look for benefits. The next section "Cui bono?" discusses scenarios in which *e*-nhanced learning can prove advantageous.

2.4.5 "Cui bono?" – benefits

Development of any type of *e*-nhanced learning is likely to be more expensive than conventional instruction. Additional expense is only justifiable if additional benefits are apparent. Let us have a look at the key benefits from students and university view:

Quality of education in *e*-nhanced learning The effectiveness of hypermedia in learning is disputed. It is likely that the use of hypermedia does not bring considerable advance to student learning (aside from Hawthorn motivational (Mayo, 1933) or novelty effects) but it is certain, that well developed materials as well as lively discussion are beneficial to student learning at college level and above. By shifting the burden of pure information transfer to lecture notes valuable lecture time is freed up for discussions and explanation of the trickier parts of the curriculum. As we found out (see (Naber, 2002b)) students long for more explanations and discussion of personal view points.

While this type of "blended learning" (a combination of presence and distance learning) does not require the use of the Internet (lecture notes on paper does just as well), well designed hypermedia can result in easier to use materials. This motivational effect can also explain some of the more bizarre findings (e.g. (Assink and van der Linden, 1991)) in the evaluation of drill & practise programs, where the evaluators could not find any difference between paper and computer based exercises, but ended up with the computer users being more satisfied. While drill & practise exercises work on paper, the process of checking the results and computing the ratio is tiresome and not instructive at all. The same is true for lecture notes as well: Everybody will prefer lecture notes containing a table of contents, an index, chapter summaries, side bars, cross references, a glossary, ... over one that only consists of the blank information. The first version will not improve knowledge, it will improve morale.

Main beneficiary: students

Enlarged clientele in *e*-nhanced and distance learning The possibility to study at home has always appealed to certain groups. Numerous distance universities proof that there is a strong market for distance education. Up to now, the student was faced with a choice of picking an isolated distance educational program (which is very hard for people who prefer learning groups or learning by discussion) or a conventional presence university (which they might not be able to attend due to distance, job, kids or a handicap). The new forms of blended and *e*-nhanced learning attracts students who can not attend a conventional university (or don't want to) but do want to marooned completely without the support from colleagues. Universities thus have access to a group of possible students who formerly where beyond reach.

Main beneficiary: students and university

Prestige A university launching an *e*-learning program can be assured of media echo. MITs announcement of putting all their lectures on the web certainly made the news. Unless the quality of the content is really bad a university can demonstrate their concern for students needs by promoting their online content, thus increasing their prestige.

Main beneficiary: university

Cost reduction by cooperation One of the more reasonable benefits is the possibility to cooperate with other universities in the effort to create more demanding projects. Whereas collaboration on conventional teaching methods is impossible or at least tedious, the creation of a shared hypermedia learning system is possible provided the software offers the necessary features. The Internet certainly has helped to form cooperations in university teaching. Before the Internet academic exchange centred around research topics and mostly took place at conferences, whereas on the Internet one can find lecturers who teach the same subjects even if their research is geared towards a different direction.

Main beneficiary: university

Cost reduction in classical distance learning Distance education has always incurred high expenses [4]. The materials have to be produced and shipped in time for the next semester and the high development costs don't allow for frequent updates. Provided that all students can access the Internet storage and shipping costs are reduced, because online material is easily updated, online communication can further cut down on presence phases thus saving in the travel department.

Main beneficiary: university

It is evident that most benefits in the introduction of *e*-nhanced learning systems at universities pertain to (apparent) cost reduction and prestige. The student point of view is rarely taken into account – "The students ought be happy that they can use this *modern* system" is a statement frequently to be heard from university administrators.

The distinctions in the category "Cui bono?" can be marked down as *student beneficial* and *institution beneficial*. Obviously any good *e*-learning environment should be both. If it does not help the students, it is a waste of time, if it is of no use to the institution, it will not be bought. In reality systems tend to be institution beneficial, simply because the institutions pay for them. Student needs are discussed in chapter 5.

2.4.6 Media usage

When it comes to media usage, German literature frequently uses the word "mediengerecht" to denote that the content was prepared in a way suitable for the medium. This is a "Unwort" and speaks clearly of our technic and media centred culture. The content is transformed to fit the medium! This is a clear case of "putting the cart before the horse" as ever seen.

In a perfect education world the medium of course would be chosen for its use to the presentation of content. To make progress in *e*-nhanced teaching we need to switch

[4]Most of todays classical distance learning is not due an impossibility of other communication methods like e.g. in Australia's outback

from media-centred culture to a content-centred culture. As pointed out before (see section 2.3) the term medium is ambiguous, as is the term *e*-learning . As *e*-learning is usually perceived to mean Internet based learning the available so-called *e*-learning tools can be grouped into 4 stages.

Stage 1: Internet based distance learning In this stage *e*-learning mimics classical DL (Distance Learning): Lecture notes (pdf or word processor files) and other materials are offered for download on a web-site or sent by e-mail. The transport medium has changed, but the content media are the same. This stage also encompasses Powerpoint slides casts and video casts. In these systems, the Internet is used to break the synchronous transmission barrier: a slide show (or even a video) of the lecture is put on the web. Again the transport medium has changed: the Internet replaces television.

Stage 2: Static hypertext systems In this stage, the hypertext and multimedia capabilities of the medium are put to use. However, due to the lack of easy to handle authoring tools for educational hypertext, not all capabilities are realized. Most of these systems are generated by a conversion from a word processing or DTP application. Even when created completely from scratch these systems rather resemble books, and make little use of the hypertext features. The online material mimics the hierarchical structure (see figure 4.2) of the lecture note it was converted from. Therefore these systems are termed "bookware" or "electronic page turning" (Jones and Jo, 1998). The systems in this stage use the Internet as a transport medium and to take advantage of the platform independent pre-installed software.

Stage 3: Interactive hypertext systems In this stage, the hypermedia system is augmented by interaction (self-test, group discussions, ...). The integration of interaction usually requires more knowledge of the Internet/WWW environment. In this stage features of the Internet are used (e.g. CGI (Common Gateway Interface) programs) that can not (easily) be emulated by other means.

Stage 4: Knowledge generating systems These systems actively involve the student in the creation process. The student can restructure, comment (annotate), augment and freshly create content. Again this way of involvement could not be achieved by conventional means.

2.5 Tools of the trade

Since *e*-learning has been booming the last years saw an explosion of tools for the creations of *e*-learning environments.

- Multimedia authoring tools

- Hypertext/Hypermedia design tools
- Webdesign tools
- Courseware tools
- ITS (Intelligent Tutoring System) and adaptive systems
- Freestyle systems
- Programming languages and frameworks
- Learning Management systems

Many commercially available *multimedia authoring tools* resemble more a *jack of all trades*, claiming to be suited for the creation of just anything – from kindergarten age edutainment software to realistic simulations of today's world trade. Some of these tools also include specialised programming/script languages. Mostly they lack support for "expert data types" (formulas, construction plans, chemical symbols, ...) and focus too much on the graphical design and too little on the "flow" of the system. As most lecturers will be overwhelmed by the sheer number of possibilities, these tools are only useful if a system expert is at hand.

The classical *hypertext and hypermedia authoring systems* like Hypercard or Hyperwave have been pushed back by the onslaught of *web-design tools.* Whereas the classical hypertext systems where mostly designed in academic environments and thus provided some support for the creation of educational hypertext, the new web-design tools are entirely geared towards business applications and focus on graphical presentation. Moreover support for expert data types is slim as well.

Courseware tools (mostly for the creation of CBT (Computer Based Training)) have been around for some time and still have a high popularity. Easy to use, they are unfortunately too limited for university level teaching, as they revolve around the generation of drill & practice exercises. Thus they can only make nice to have add-ons for other systems.

On the other end of the spectrum there is a fair number of fully fledged *adaptive hypertext educational systems* or ITS (see Brusilovsky's overview article (Brusilovsky, 1996)) which aim at replacing the teacher completely. The creators of these systems usually build them as exercises in computer learning and artificial intelligence. While there are some really interesting (and working!) projects around they always deal with very structured content (programming languages, mathematics) so it is likely that this type of approach is only useful in a very limited field.

Besides being arcane (and therefore solely used by their creators) the amount of time necessary to build an environment is gigantic. Moreover these systems obviously do not accommodate the students' wishes: in a recent study (Naber, 2002b) we found, that despite of the harsh criticism of their lecturers, students are not bent upon replacing them by an ITS, but rather wish for a highly interactive "question and answer" type of "non-lecture" (and better learning materials). More student-lecturer communication was also established a major "pro *e*-learning " factor in a study by Schell (Schell, 2001).

Many forward thinking lecturers battle a conglomerate of unrelated tools like university provided lecture web-sites, third party hosted discussion forums and their choice of word processors or hypermedia authoring tool to support their lecture. In fact, it is possible to build usable educational hypertext systems using this *freestyle approach* by combining any of the above mentioned tools, but you could as well be using assembly language (and this is not generally considered a good idea).

Programming languages offer the most flexibility but require a programmer. Most applications do not need that much flexibility and the use of a programming language quickly becomes a case of breaking flies on wheels.

A specialised university level hypertext system is called for, that provides support for all the tasks specific to creating engaging and effective hypertext study materials. Specialised university level hypermedia systems come under the heading LMS *(Learning Management System)* or LMP *(Learning Management Platform).* In these systems content creation and content distribution is only a small part. Depending on the product nearly every aspect of university life from enrolment, payment, exams ... is covered. The lecturer support among these tools vary. Some of the low-end products only feature a simple lesson planner (lecture time,/place, materials, assignments) with the possibility to upload materials. They usually come with a forum for each course. More advanced tools offer web based hypertext editors or standalone programs for editing materials. Common LMS include Blackboard(Blackboard Inc, 2003), WebCT(Goldberg et al., 1996) , IBT(IBT, 2003) and the German Open Source project ILIAS(Ilias, 2003). Detailed descriptions and evaluations of current LMS are found in (Baumgartner et al., 2002) and (Schulmeister, 2003). As the main focus of LMS is university and student management, they are not suitable tools for lectures only interested in bringing their own courses online.

2.5.1 Why do existing tools fail?

Although there is number of tools available to produce *e*-nhanced lectures, many of them are not suited for the academic environment and complaints are abundant. On a general level, reasons for common failings include:

Wrong focus: Many tools are geared towards the creation of business training, lacking support for data types commonly found in the academic environment (e.g. formulas, qotations, etc.). Even many of the teaching/learning centred ones are aimed at K12 education rather than college or university level.

Skills: Although there are excellent tools about to build learning materials (complete with animations, ...), these require considerable skill on part of the author.

Time: Adding training time, to the time actually spent producing the materials, the use of some tools takes up too much of the lecturer's time.

Money: Purchasing tools and hiring a group of specialists to help lecturers with the tools exceeds the budget.

Benefits: Some of the easier to use tools, do not actually improve the student situation. Downloading the lecture notes instead of buying them at the department office is not much of an improvement.

Approach: Existing tools fail in considering users' needs properly. Usable *e*-learning must lend itself to support different computer literacy levels of lecturers and students. Aside from literacy aspects, user needs and requirements are to be clarified.

Legal considerations: Many of the mostly US based products do not meet European or e.g. Australian standards for privacy or accessibility. Especially Australian universities were faced with the problem last year, when the government passed a new law on accessibility rendering existing installations of LMS (especially WebCT) illegal.

2.6 The case against *e*-learning

One of the leading arguments against *e*-learning lies in the *e* itself: new – electronic – media require new means, and little is still known about structuring hypertext. Even though hypertext has been researched since the 1950s, it became popular only at about 1995, with the rise of the world wide web. As book publishing has gone a long way since Gutenberg, we will require some time to adapt to hypertext publishing. Many of todays hypertext lecture notes cannot deny their print origins - the structure is completely linear/hierarchical, organised in separate subsequent chapters. Only the transport medium has been changed, the power of the new medium has not been released.

Also, *e*-learning involves computers, which adds an additional dimension to the whole problem: to learn with a computer, you'll first have to learn about computers (compare (Watson, 2001)):

> There is a distinct difference between teaching with computers, and teaching about computers. Due to the generally bad usability of "teaching with computers" there arises an artificially created need of having first to teach about computers before the actual goal (teaching something using computers) can be tackled. So the difference between means and subject of teaching became increasingly blurred.

In their article *The Frontier of Web-based Instruction* (Mitchell et al., 2001) Mitchell, Dipetta and Kerr compare the evolution of *e*-learning to the exploration of Americas wild west, complete with *Lone Rangers*, *Greenhorns* and *Band Wagons*. *e*-learning is driven by *Lone Rangers*, interested individuals with the necessary technical and educational knowledge. The occasional *Greenhorn*, an interested person without the necessary knowledge either evolves to a *Lone Ranger* or has to wait for the *Band Wagon*, e.g. the department organising a large scale *e*-learning venture.

> Despite recent advances, moving into technologically mediated instruction and course delivery remains akin to exploring uncharted territory ... Adventure stories are still being written"

Although everbody likes a little adventure now and then, many of the adventurous experiments are at the students' expenses. Even web-based courses designed by experienced *Lone Rangers* can fail, a *Greenhorns* first foray into web-based courses can easily proof disastrous for the students (Montelpare and Williams, 2000).

A move toward *e*-learning should be carefully considered. Will the benefits outweigh the drawbacks? How much time and money will the project cost? Will the students benefit? How will I deal with technological problems? Who will support the students if they run into problem? Do I have the long term commitment to see this project through the years to follow? How will I measure success and what will be considered as failure? Will I be ready to forsake my investment in the case of failure? These are questions a lecturer should pose himself before he plunges headlong into the *e*-learning adventure. Just like everybody else lecturers are susceptible to novelty and Hawthorne effects and thus *e*-learning projects are too frequently started by a short term interest in new media or to console oneself over recently suffered academic defeat.

2.7 The case for *e*-learning systems – or Why we do need yet another *e*-learning system

As of now we have ascertained that *e*-learning projects are time consuming and expensive, the benefits are doubtful and the technology is complicated. So why should anybody embark on another *e*-learning project?

Because

- A well designed system will cut production time
- A well designed system will provide synergies with research
- A well designed system will ease student learning and raise motivation
- Information transfer per Internet will free lecture time for discussions
- Discussions will provide new insights and are less tedious than lectures

- Handicapped students will profit from *e*-nhanced materials
- Working students will gain flexibility

These are all benefits any *e*-learning system claims for itself. Reality however shows that these don't hold. This however does not make the aim invalid, just difficult to reach.

Usable tools for the average lecturer – not computer scientists, not educationalist, but sociologists, architects, theologists ,... – are needed to *e*-nhance lectures.

2.8 *e*-ULE – a jack of one trade

A university level *e*-learning system should reflect the academic nature of the contents to be taught as well as the limited budget most universities are confronted with. It also must take into the account that most university teachers are neither trained information scientist nor educationalists. Thus the system must provide a very easy to use authoring system and a process to guide the lecturer towards the creation of a usable educational hypertext system.

e-ULE is consequently designed from user and task analysis and focuses on usability and end user benefits. Its foremost aim is to reconcile student needs with lecturer requirements.

e-ULE is not a LMS and does not aim to provide functionality to run whole universities, but it can interface to university management systems. *e*-ULE can be used by single work groups or can be interconnected to service whole universities. It is thus suited for grassroot introduction of university *e*-learning .

The *e*-ULE system features

- A cross-platform authoring client that fully supports all stages of academic publishing (see chapter 6 for task analysis and section 7.4 for implementation)
- A cross-platform accessible browser-based student client which supports the student in all learning stages (see chapter 5 and section 7.3 for implementation)
- A server backend to handle adaptive and collaborative features as well the generation of personalised/specialised versions of the material (see section 8.2 for technical details)
- *e*-ULE|doc XML-based language for *e*-ULE content (see section 8.3 for technical details)
- Out of the box server install on a standard computer with the possibility to cluster *e*-ULE server (see 8.2 for technical details)

Chapter 3

Psychology of instruction

You do not really understand something unless you can explain it to your grandmother.

Albert Einstein

Over the years a vast number of learning theories have been established. However these can be reduced to the three *classical theories* behaviourism, cognitivism and constructivism and the younger theories incorporating elements of *personality development* theory and social psychology.

These theories emphasise that teaching and learning can neither be separated, nor severed from constraints of personal traits of outer circumstances.

One promising theory is R.J. Sternberg's theory on *Thinking Styles* (Sternberg, 1999), another on is the *Learning Styles* theory propagated by Silverman and Felder, based on the Myers-Briggs personality type indicators (MBTI) (Myers, 1962).

Both theories take university teaching and learning into account.

3.1 Classical learning theories

Behaviourism: Teaching and learning consists of information transmission, memorisation and reproduction.

Cognitivism: Teaching and learning consists of the tranformation of information into knowledge.

Constructivism: Teaching and learning is concerned with the generation of knew knowledge.

The first two of these evolved in the early years of the 20th century, the constructivism springing from the second half of the 20th century is comparatively young.

Whereas both Behaviourism and Cognitivism stress the teaching (or teacher) aspect, Constructivism favours a student centered approach.

3.1.1 Behaviourism

Behaviourism is a theory of animal and human learning that only focuses on objectively observable behaviours and discounts mental activities. Behaviour theorists define learning as nothing more than the acquisition of new behaviour (Edelmann, 2000), (Krapp and Weidemann, 2001).

Experiments by behaviourisms identify conditioning as a universal learning process. There are two different types of conditioning, each yielding a different behavioural pattern:

1. Classic conditioning occurs when a natural reflex responds to a stimulus. The most popular example is Pavlov's observation that dogs salivate when they eat or even see food. Essentially, animals and people are biologically "wired" so that a certain stimulus will produce a specific response.

2. Behavioral or operant conditioning occurs when a response to a stimulus is reinforced. Basically, operant conditioning is a simple feedback system: If a reward or reinforcement follows the response to a stimulus, then the response becomes more probable in the future. For example, leading behaviorist B.F. Skinner used reinforcement techniques to teach pigeons to dance and bowl a ball in a mini-alley.

There have been many criticisms of behaviourism, including the following:

- Behaviourism does not account for all kinds of learning, since it disregards the activities of the mind.

- Behaviourism does not explain some learning – such as the recognition of new language patterns by young children – for which there is no reinforcement mechanism.

- Research has shown that animals adapt their reinforced patterns to new information. For instance, a rat can shift its behavior to respond to changes in the layout of a maze it had previously mastered through reinforcements.

- Human memory is only regarded as an information repository. Knowledge is stored, not processed (cognitivism) or constructed (constructivism)

This theory is relatively simple to understand because it relies only on observable behaviour and describes several universal laws of behaviour. Its positive and negative reinforcement techniques can be very effective – both in animals, and in treatments for human disorders such as autism and antisocial behaviour. Behaviourism often is used by teachers, who reward or punish student behaviours.

The behaviouristic group also includes Thorndikes *connectionism* (Thorndike, 1932), which is centred around the laws of exercise, effect and readiness, and various mathematical learning theories (e.g. Hull (Hull, 1943)).

Even though behaviourism is rather outdated, many of todays computer mediated teaching aids rely heavily on it. This seems almost natural, if one remembers that behaviourism is only concerned with ensuring that a certain amount of information can be reproduced by the student in the alloted time slot. Presenting a given curriculum in a linear fashion and ensuring that the presented information is correctly repeated is a task ideally suited for computers.

Software based on the behaviouristic aspects includes *drill & practice programms* and *tutorials*. Drill & practise programs like vocabulary or spelling trainers follow the pattern: query - evaluate answer - produce positive/negative reinforcement. Tutorials evolve these systems by adding a presentation phase, which is followed by a drill & practice section.

This shall not lead us to the conclusion that behaviourism has no place in todays university teaching: every subject – no matter how advanced – requires some knowledge that always has to be "pat". Medics have to know the names of the various body parts, it simply would not do to lookup what part the other doctor referred to during an operation. Therefore even a specialised university level hypermedia authoring tool has to provide features for the generation of behaviouristic learning exercises. aspects.(Phillips et al., 1991)

3.1.2 Cognitivism

Cognitivism focuses on the thought processes of the students. The student is regarded as an unique individual. Information is presented and will be processed in the brain and leads to output. Learning is an interaction between external information and internal knowledge. Teaching and learning is not about information transfer, but about information processing, which in turn leads to the generation of knowledge. Cognitivsm aims at improving conceptualised thinking and problem solving abilities (Edelmann, 2000), (Lefrancois, 1994).

Jean Piaget and his works on child development (Piaget called his general theoretical framework "genetic epistemology") are strongly connected with the cognitivism. Another propagator of cognitivism was Lev Vygotsky. His ZPD (zone of proximal development) modell describes an interplay between student teacher and learning material. In the ZPD model student, teacher and problem should optimally contribute to the solution of the problem.

Another propagator of cognitivsm is J. Bruner. Like Piaget and Vygotsky Bruner started with child development stages, but later moved on general learning theories, thus founding the theory of concept learning. Concept learning consists of deveploping new correlations which allow a connection between an unknown object and a linguistic concept. A concept is any rule by which a certain stimulus is connected to a certain reaction. If the rule is represented by a word, this word denotes a concept. A major theme in the theoretical framework of Bruner is that learning is an active process in which learners construct new ideas or concepts based upon their current/past knowledge. The learner selects and transforms information, constructs hypotheses, and makes decisions, relying on a cognitive structure to do so. A cognitive structure (i.e., schema, mental models) provides meaning and organisation to experiences and allows the individual to "go beyond the information given".

As far as instruction is concerned, the instructor should try and encourage students to discover principles by themselves. The instructor and student should engage in an active dialog (i.e., socratic learning). The task of the instructor is to translate information to be learned into a format appropriate to the learner's current state of understanding. Curriculum should be organised in a spiral manner so that the student continually builds upon what he has already learned.

Bruner states that a theory of instruction should address four major aspects:

- predisposition towards learning
- the ways in which a body of knowledge can be structured so that it can be most readily grasped by the learner
- the most effective sequences in which to present material
- the nature and pacing of rewards and punishments

Good methods for structuring knowledge should result in simplifying, generating new propositions, and increasing the manipulation of information.

Later Bruner incorporated social learning into his theory an thus linked cognitivism and constructivism.

A cognitivistic approach is best realised by using an ITS (*intelligent tutoring system*). These systems are marked by continously improved learner profiles, which allow the programm to adapt to the individual learning progress. ITS do not only have to represent the exact knowledge but also knowledge about teaching and didactics. ITS usually rely on a series of tests to analyse the students progress. This is one of the reasons, that ITS usually are found in rather deterministic topics like programming languages (Anderson, 1993) or mathematics (Brusilovsky et al., 1996).

A cognitivistic approach can also be supported by hypermedia and simulation software. When using hypertext/hypermedia the student is forced to select the next stept on his own, which can lead to stronger involvement with the topic or to the "lost in hyperspace" syndrome (Conklin, 1987). Simulation software allows a student to vary various system parameters and to observe the outcome.

All three types of tutoring aids have in common that they are difficult to create. ITS are mostly found in the computer science sector (where they are solely used by their creators and maybe an handful of colleagues. Simulation-software is mostly generated by some expert (costly!). Hypertext/hypermedia systems sometimes are generated by none-experts. However, it has to be noted that a collection of HTML pages does not make a hypertext learning system. Todays HTML editors are far from helping their users in the creation of useful hypertext systems.

3.1.3 Constructivism

In contrast to behaviourism and cognitivism the *constructivism* is less clearly defined. Constructivism originated in the field of epistemology (theory of knowledge). Unlike this radical constructivism, the *new constructivism* in pedagogical psychology centres on thinking and learning processes (Krapp and Weidemann, 2001).

Drawn from the field of business and vocational training, the new constructivism has been around since the 1980'ies and is slowly gaining momentum in central Europe.

Constructivism is a philosophy of learning founded on the premise that, by reflecting on our experiences, we construct our own understanding of the world we live in. Each of us generates our own "rules" and "mental models", which we use to make sense of our experiences. Some constructivists go as far as to deny an absolut objective reality on the premises that everybody constructs his own reality. Learning, therefore, is simply the process of adjusting our mental models to accommodate new experiences.

There are several guiding principles of constructivism:

- Learning is a search for meaning. Therefore, learning must start with the issues around which students are actively trying to construct meaning.

- Meaning requires understanding wholes as well as parts. And parts must be understood in the context of wholes. Therefore, the learning process focuses on primary concepts, not isolated facts.

- In order to teach well, we must understand the mental models that students use to perceive the world and the assumptions they make to support those models.

- The purpose of learning is for an individual to construct his or her own meaning, not just memorise the "right" answers and regurgitate someone else's meaning. Since education is inherently interdisciplinary, the only valuable way to measure learning is to make the assessment part of the learning process, ensuring it provides students with information on the quality of their learning.

Constructivism calls for the elimination of a standardized curriculum. Instead, it promotes using curricula customized to the students' prior knowledge. Also, it emphasizes hands-on problem solving (Brooks and Brooks, 1999).

Under the theory of constructivism, educators focus on making connections between facts and fostering new understanding in students. Instructors tailor their teaching strategies to student responses and encourage students to analyse, interpret, and predict information. Teachers also rely heavily on open-ended questions and promote extensive dialogue among students.

The constructivist approach can be supported by hypertext and hypermedia, simulations (as described in the section 3.1.2 on cognitivism) and micro-worlds. Microworlds extend the concept of simulations in so far as they allow for change of the simulation itself. The student is not confronted with a simulation in which he can vary some parameters, but is forced to generate the simulation "from scratch".

3.2 Thinking Styles

According to Sternberg (Sternberg, 1999) a thinking style is a

> preferred way of thinking. It is not an ability, but rather how we use the abilities we have.

The theory relies on a principle of *mental self-government*, a kind of reverse theory to explain the separation of powers found in many of todays countries. Just as a government needs to decide, allocate resources or react to changes, so does an individual. The mental government can be further divided into

- functions
- forms
- levels
- scopes and
- leanings

3.2.1 Functions of mental self-government

Just as the government serves three functions: *executive*, *legislative* and *judicial*, persons can be classified by their preferred functions:

- Legislative people
- Executive people
- Judicial people

Legislative people like to come up with their own ways of doing things. Rather to follow a given plan, they will search for alternate ways to solve the problem. Prefabricated or pre-structured problems are of no interest to them. Legislative people are usually very creative, but tend to have no mind for details. Many researchers fall into this group.

Executive people prefer to work in pre-structured environments following fixed rules. They like to fill gaps in structures more than creating new ones, and in contrast to the legislative people, they mostly enjoy details. Executive types tend to join administrative ranks, but also make good teachers.

Judicial people prefer to analyse and compare rules, procedures or theories. Rather than creating new theories or evolve existing theories, they will compare competing theories. This style is often combined with both, legislative and the executive style and is quite commonly found in University personnel.

Functions and education

Educational systems have a huge impact on the development of styles. Table 3.1 lists common educational institutions and their favoured styles.
executive styles.

Educational institution	Styles
kindergarten	legislative
primary school	executive
high school	executive
junior college/ university undergraduates	executive, judicial
senior college/ university graduates	judicial, legislative

Table 3.1: Educational systems and their predominant styles

As the majority of people commands all three style functions, promoting a single one, usually is not a problem. However, a longterm selection of one style function, such as is evident in the educational systems persistency on executive style function, can cause severe disadvantages for students ill suited to adopt the favoured style function.

After twelve to fifteen years excelling in a predominantly executive orientated environment, executive students are suddenly expected to fulfil mainly legislative and judicial tasks, they are little prepared for. Legislative minded students, who often are turned off by accumulation of tons of detail information usually fed to students in introductory courses, might have turned to other occupations and career paths by now.

The American educational system suffers from high dropout rates in grad-school. This situation is especially severe as former A-level students suddenly failing their assignments are afflicted by doubts and loose self-esteem.

The situation in Austria is more relaxed, as the break between predominantly executive and predominantly legislative style function occurs with the entry to university. Many of the university drop-outs (previously released to training on the job) find a new home in the more structured "Fachhochschulen" where their executive style function is better supported.

It is evident, that our educational systems selects the wrong people for the job.

It is not only the style function of the system that matters, the inter-person relationships are also governed by style function compatibility. A legislative teacher and an executive student make as bad a pair as executive teacher and a legislative student. In a teaching situation, the lecturers task is to adjust to different students styles.

3.2.2 Forms of mental self-government

The theory of mental self-government defines four forms: monarchic, hierarchic, oligarchic and anarchic. The forms present different approaches to world, and real world problem solving.

Monarchic people are single minded and driven. Once they have set their mind to something they usually succeed. Everything else is of no interest to them. They make dedicated teachers and – provided they are interested in the subject – excellent students.

Hierarchic people are great organisers, able to fulfil various task at the same time. They build hierarchies of tasks and recognise the need to priorise. Their structured approach makes them ideally suited for teaching. As students they stand out by their organised work, but may have difficulty in assimilating unstructured information.

Oligarchic people like hierarchic people are able to fulfil various tasks at the same time, but do not feel the need to structure or priorise in a strict way. Both students an teachers are marked by their flexibility, but left to their own can become unorganised.

Anarchic people are unable to organise or priorise. They are driven by various – often competing – goals and devote their time and energy in a random way. Teachers with an anarchic form will leave their students puzzled. Students will succeed in assimilating information even from the most obscure situations, but will miss every deadline.

Forms and education

The influence of educational systems on forms is much weaker than on functions. Still, there is a slight preference for the hierarchic style, easily explained by the fact, that independent from the personal style form, information presented in an organised hierarchical format is the easiest to understand.

Teachers of the monarchic type might inspire their students, or scare them away. If the monarchic style form manifests itself in a decline to consider synergies or connections, students might be turned away.

3.2.3 Levels, scope and leanings of the mental self-government

In additional to the main styles *function* and *form* there also *level*, *scope* and *leaning* to be considered (see table 3.2).

Levels	Scope	Leanings
global	internal	liberal
local	external	conservative

Table 3.2: Minor thinking styles

Local individuals like concrete problems and details. They are pragmatic, down-to-earth, but may loose the forest for the trees.

Global individuals aim for the big picture. They can plan a mars mission, but will crash the mars probe, because centimetres and inches are all the same to them.

Internal individuals tend to be introverted, task-oriented, aloof, sometimes socially unaware. They prefer to work alone.

External Individuals tend to be extroverted, outgoing and people oriented. They enjoy discussions and groupwork.

Liberal individuals like to go beyond existing rules and procedures. They like change and often are easily bored.

Conservative individuals prefer to work within the system. They like rules and predictable environments.

Minor styles and education

Local and global individuals are equally disadvantaged in educational settings, depending on the teachers style. A local teacher will prefer local students, as global teacher will prefer global students. As the local style is correlated to the executive style (see section 3.2.4), preference for the local style in undergraduate studies can be assumed. Obviously real world problem solving can only succeed if local and global individuals are balanced in the team.

In practice, the question of stylistic scope does hardly arises. In every project there are niches for internal or external individuals. In a purely academic setting style normally enters in assessment situations: external individuals might perform poor on individual exams, just because they lack somebody to voice their thoughts to.

Style leanings, again are mostly dependent on teacher and student having the same style. The most innovative solution is lost on a teacher who will only accept the textbook version.

3.2.4 Correlations between styles

Statistical evaluation shows that there is some correlation between the styles. Although style function, form, level, scope and leaning are basically independent and each individual can have a personal style portfolio, on the average, some of the styles go together.

The two variants of level, scope and leaning seem to be mutually exclusive. People are either, global or local in their style levels, internal or external in their stylistic scope and liberal or conservative in their leanings.

There is a positive correlation between the legislative stylistic function and liberal leanings as well as between executive stylistic function and conservative leanings. Naturally this results in negative correlation for the legislative-conservative and executive-liberal pairings.

The statistics also yield a negative correlation for a pairing between the judicial stylistic function and the oligarchic form. This in itself is not surprising, as one might expect a judicial person to prefer the hierarchical form. The hierarchical scale however remains completely independent from the other styles. This might indicate, that the hierarchical form is not native, but rather learned by socialisation.

3.2.5 Supporting different styles

Teaching and assessing methods compatible with the thinking style of each individual student certainly will improve student learning and performance. At least some of each should match the students' preferences. On the other hand it is unlikely that the world will always provide information in a suitable style – other styles can and should be taught.

Certain instructional methods stress different styles. Table 3.3 contrasts method and preferred style.

For a detailed discussion of styles and teaching methods refer to section 5.3 general question - compare school - spend too much time on things ill suited, ...

Thinking styles also influence assessments. Table 3.4 depicts assessment methods, skills and the most compatible styles.

3.3 Learning styles

Aside the more abstract thinking style there is a vast range of activity and personality centred theories. Among the activity centred theories are Kolb's Learning Style Inven-

method of instruction	most compatible style(s)
lecture	executive, hierarchical
reading	internal, hierarchical
exercises (given problems)	executive
though based questioning	judicial, legislative
cooperative/group learning	external
projects	legislative
discussion	judicial, external
question & answer	external, legislative

Table 3.3: Thinking systems and method of instruction

tory (LSI) (Kolb, 1978) focused on education as well as Holland's occupation centred theory of the Strong Vocational Interest Blank (SVIB) (Holland, 1973).

The distinction between a personality style and a personality trait, albeit is rather weak. The Grasha-Riechmann Student Learning Style Scales and The Myers-Briggs Type Indicator derived from Jung's personality types theory as well as several others belong to that group.

3.3.1 Kolb's learning style model

Kolb's model categorises students in four groups by their preferences for *information intake* (*concrete* or *abstract*) and *internalisation* (*active* or *reflective*)

Although this model is limited in its ability to represent real-life learning situations, its simplicity is quite catching. Julie Sharp reports good results in teaching the Kolb model to undergraduates engineering students as part of a technical communication class (Sharp, 1998), (sharp, 2001).

The four *characteristic questions* (Why?, How?, What?, What if?) make for a good rule of thumb to check newly added content for minimal learning style diversity.

3.3.2 Grasha-Riechmann Student Learning Style Scales

The Grasha-Riechmann Student Learning Style Scales (GRSLSS), an instrument developed in the early 1970s, has been used to identify the preferences learners have for interacting with peers and the instructor in the classroom setting (Grasha and Richlin, 1996). This instrument was specifically designed for college and university settings.

The six social learning styles identified by this model on three axis are *Independent* versus *Dependent*, *Competitive* vs. *Collaborative* and *Avoidant* vs. *Participant*.

The Independent learner prefers independent study, self-paced instruction and would prefer to work alone on course projects than with other students. Dependent learners look to the teacher and to peers as a source of structure and guidance and prefer an authority figure to tell them what to do.

method of instruction	skills tapped	most compatible style(s)
multiple choice, short answer	memory	executive, local
	analysis	judicial, local
	time allocation	hierarchical
	working by self	internal
essay form	memory	executive, local
	Macro analysis	judicial, global
	micro analysis	judicial, local
	creativity	legislative
	organisation	hierarchical
	time allocation	hierarchical
	acceptance of teacher viewpoint	conservative
	working by self	internal
projects and portfolios	analysis	judicial
	creativity	legislative
	teamwork	external
	working by self	internal
	organisation	hierarchical
	high commitment	monarchic
interview	social ease	external

Table 3.4: Thinking systems and method of assessment

Competitive learners learn in order to perform better than their peers and to receive recognition for their academic accomplishments. Collaborative learners learn by sharing and by cooperation with teacher and peers. They prefer lectures with small group discussions and group projects.

Avoidant learners are not enthused about attending class or learning class content. They are typically uninterested and are often overwhelmed by class activities. The Participants enjoy class and make good class citizens. They are interested in class activities and discussion and eager to do class work.

3.3.3 Jung's personality types

In Europe Carl Gustav Jung is best known for his work on the *collective unconscious* and his creation of pan-cultural *archetypes* (e.g. Anima, Animus, Hero, Saviour, Dragon, ...) (Herkner, 1992). In the Anglo-Saxon world his work on personality types has influenced many of todays learning theories, especially those dealing with learning types.

Jung based his theory of personality psychology on three different *dimensions* or

type	characteristic question	description
divergers concrete, reflective	Why?	Interested in applications of material. The teacher should act as a *motivator.*
assimilators abstract,reflective	What?	Information should be presented in an organised, logical way. The teacher should act as *expert.*
convergers abstract, active	How?	Active work on predefined tasks, trial-and-error. The teacher should function as a *coach.*
accommodators concrete, active	What if?	Learners like to apply learned materials to new situations. The teacher should *stay out of the way.*

Table 3.5: The four types of learners in the Kolb learning style model

functions. Each dimension or function is marked by two opposing types.

The *perception* dimension combines *Sensation* and *Intuition.* Sensing people prefer objective measure and facts, whereas intuitive individuals prefer a subjective and more mysterious way of perceiving. These two views are mutually exclusive.

The *judging* dimension encompasses *Thinking* and *Feeling.* Thinking rules between true and false, feeling will judge pleasant or unpleasant. Again, thinking and feeling are mutually exclusive.

The third dimension classifies our dealings with the world and divides between *Introversion* and *Extroversion.*

The theory also states, that each individual unites both ends of the scales and that his subconsciousness is ruled by the complementary styles. This complement is the root of personality problems as well as a source of personal change.

The three Jungian dimension are also used by the Myers-Briggs Type Indicator, Murphy Meisgeir Type Indicator, and the Keirsey-Bates Temperament Sorter.

3.3.4 Myers-Briggs type indicator

The Myers-Briggs Type Indicator™(MBTI) is an instrument for measuring a person's preferences, using four basic scales with opposite poles. The four scales are: *Extraversion* versus *Introversion*, *Sensate* vs. *Intuitive*, *Thinking* vs. *Feeling* and *Judging* vs. *Perceiving.*

The MBTI adds fourth dimension (judging - perceiving) to the three Jungian dimensions. People who prefer *Judging* tend to like a planned and organised approach to life and prefer to have things settled. People who prefer *Perceiving* tend to like a flexible and spontaneous approach to life and prefer to keep their options open.

The various combinations of these preferences result in 16 personality types, denoted by 4 letter combinations (e.g. ESTJ for extrovert-sensate-thinking-judging)

The MBTI enjoys great popularity in the USA and is used by many colleges and career counselling institutions. Tests come in various formats, specifically designed for school, college or vocational uses.

The widespread use in colleges has lead to a number of materials (e.g. (Brightman and Robinson, 1998)) concerning type based teaching strategies.

3.3.5 Felder-Silverman learning style model

Felder and Silverman derived their learning style model (LSLSM) (Felder and Silverman, 1988), (Felder, 1993) from the MBTI / Jungian personality types and the Kolb model. However they integrated elements of about sensory modalities also found in *Neuro-Linguistic Programming* (NLP): visual, auditory and kinaesthetic.

The four dimensions in the Felder-Silverman model therefore read *sensing – intuitive*, *visual/spatial – verbal/auditory*, *active – reflective* and *sequential – global*.

A fifth dimension *inductive – deductive* was later abolished, as it became clear that the students preferences on this axis were altered by external influences (exam cram - "just the facts please") and "sheer laziness" ("before I think about it, I'll listen to the lecture, even if it bores me to death"). Induction refers to "learning by observing/doing", whereas a deductive teaching style would represent mostly facts in an ordered manner.

The sensing – intuition dimension is straight out of Jung and refers to the ways in which people perceive the world: Sensing involves observing, data gathering, while intuition involves direct perception by the way of the unconscious (hunches, speculation, ...).

The visual/spatial – verbal/auditory dimension was bereft of the third kinaesthetic mode and now voices a student preference for visual or verbal input. Textual input remains an unsolved problem, as the information is basically captured by the eyes, but neuro-psychology confirms, that the text is converted to verbal/aural information to be understood.

The active – reflective dimension replaces the introversive – extroversive dimension in the MBTI and also encompasses the kinaesthetic modality. Active learners want to experiment, talk things through and get their hands on something. Reflective learners require time to think things through. This dimension seems to overlap with sensing – intuition, but in fact they are completely separated: the sensing-intuitive dimension refers to the information acquisition process, while the active – reflective dimension is concerned with information processing.

The sequential – global dimension again refers to preferences in information acquisition. Sequential learners learn in little steps, global learners in whole blocks, they require a big picture to get started.

Felder and Silverman also specified teaching-styles, which for the most part are analogous to the learning styles, with one exception: The active– reflective dimension becomes active – passive and not even reflective students will profit from an passive instructional method.

The FSLSM was especially design for college and university education in the engineering field. In their original article (Felder and Silverman, 1988) Felder and Silverman analyse teaching and learning styles in engineering schools, finding a large amount of type mismatch.

3.4 Popular fallacies in type/style theories

As any psychological (or – as matter of fact – scientific) theory released to the public, learning and thinking style theories are prone to be misrepresented, overrated or misunderstood.

A common side effect of type theories is the danger of *pigeon-holing* individuals. A personality type or thinking/learning style is neither absolute, nor unchangeable. Moreover types/styles are statistically derived elements, which need not necessarily fit each and every individual. Each individual is capable of choosing an apt style in its personal inventory.

A theory also should not turn into a religion, as has commonly been reported in the case of the widespread MBTI. The MBTI's somewhat horoscope-like description is susceptible to the Forer effect [1]: people, surprised by the high likeness of their profile, are lured to believe that they can/must not deviate from it and are thus actually hampered in their personal development.

Psychological theories are also susceptible for the *post hoc ergo propter hoc fallacy*, based upon the mistaken notion that simply because one thing happens after another, the first event was a cause of the second event. A student failing an exam was not necessarily tripped by an incompatible assessment style, it could be plain old laziness or simply lack of ability.

Still, thinking and learning styles can lend important insight into the learning process. For a teacher, stepping outside traditional paths can enlarge the personal understanding of the subject matter, a student might find new ways to assemble information. Testing *e*-learning frameworks against various learning theories therefore will certainly do no harm.

3.5 Comparison of learning theories

A comparison of the introduced learning theories can be found in table 3.6. Although Sternberg's thinking Style theory is not dedicated to higher education learning, it

[1]The Forer Effect, also known as the subjective validation effect states that people tend to accept vague and general personality descriptions as uniquely applicable to themselves without realizing that the same description could be applied to just about anyone.

proposes a very comprehensive instrument. However it can be augmented by taking Felber-Silverman Learning Style modalities into account. The Grasha-Riechmann Learning Styles add more detail to the rather bland introvert/extrovert dimension.

The classical Learning theories behaviourism, cognitivism and even constructivism are off the chart, because their concepts are too limited. Each of this theories can only explain a very small segment of learning.

As can easily be seen from table 3.6, Steinberg's thinking styles model is the most detailed theory, with the other theories covering parts the same ground with slight variations. Felder however, introduces the visual/spatial and verbal/aural categories, which stem from NLP (Neuro Linguistic Programming). NLP also includes a third domain called kinaesthetic/haptic, which describes a preference for "hands-on" experiences. Together they form the basis for learning style evaluations used in this thesis. The sections on student needs and lecturer requirements (see chapter 5 and 6) will illustrate specific needs in the light of personal thinking and learning styles and list measure for enhancing effectivity for various groups.

thinking styles		Kolb	Jung/MBTI	Felder	Grasha
functions	legislative executive judicial		sensing, thinking intuitive, feeling	inductive deductive, sensing intuitive	independent dependent
forms	monarchic hierarchic oligarchic anarchic				
level	local global	concrete abstract		sequential global	
scope	internal external	active reflective	internal external	active reflective	avoidant, competitive participant, collaborative
leaning	conservative liberal		judging perceiving		
				visual/spatial verbal/aural	

Table 3.6: Comparison chart of various thinking and learning style theories. Most noteable is the slim support for style form (organisation) and input modality (visual/spatial, verbal/aural). Steinberg's thinking styles model is the most detailed theory. Together with Felder's visual/spatial and verbal/aural identifiers, it form the basis of our learning style model.

Chapter 4

Hypertext and hypertext usability

If I have seen farther than others, it is because I was standing on the shoulders of giants.

Isaac Newton

This section will take a look at existing hypertext systems and the advantages and disadvantages of hypertext in general and in the light as a learning medium.

4.1 Hypertext Systems

To the children of the WWW-era a world without hypertext is hardly fathomable. But indeed there was live before the WWW, in fact there was hypertext before HTML (HyperText Markup Language). Vannevar Bush, Douglas Engelbart, Ted Neslon, the early trailblazers of nonlinear information produced a wealth of concepts, both fresh and inspiring if sometimes impossible to implement.

In designing a hypertext system it is good practice to go back to roots and analyse the strong and the weak points of the various concepts available.

4.1.1 Some very early ideas

The earliest precursor of todays hypertext systems date back medieval times with the development of a rich system of cross references and marginalia. The basic document model for hypertext was set: things the combination of text and graphics, and cross references to other works. These early hypertext links were able to target documents to a fine level thanks to conventions for numbering lines or verses, which in fact represent a somewhat crude content based markup. With the mass print production, references were no longer based on the actual content, but on page numbers, thus causing a separation of content and reference.

4.1.2 Ahead of time: Vannevar Bush and the MEMEX

The history of modern hypertext begins in July of 1945. President Roosevelt's science advisor during World War II, Dr. Vannevar Bush, proposed *Memex* in an article titled "As We May Think" [1] (Bush, 1945a), (Bush, 1945b), published in The Atlantic Monthly.

In the article Bush touches upon various technical methods to improve academic work, business and private communication/information storage but the most impressive of these artefacts is the *Memex*. The *Memex* is a machine fashioned like a writing desk, but augmented by a microfiche storage and two projection screens. A camera can capture handwritten/drawn content and save it to microfiches. A duplication unit can reproduce the microfiche content for distribution.

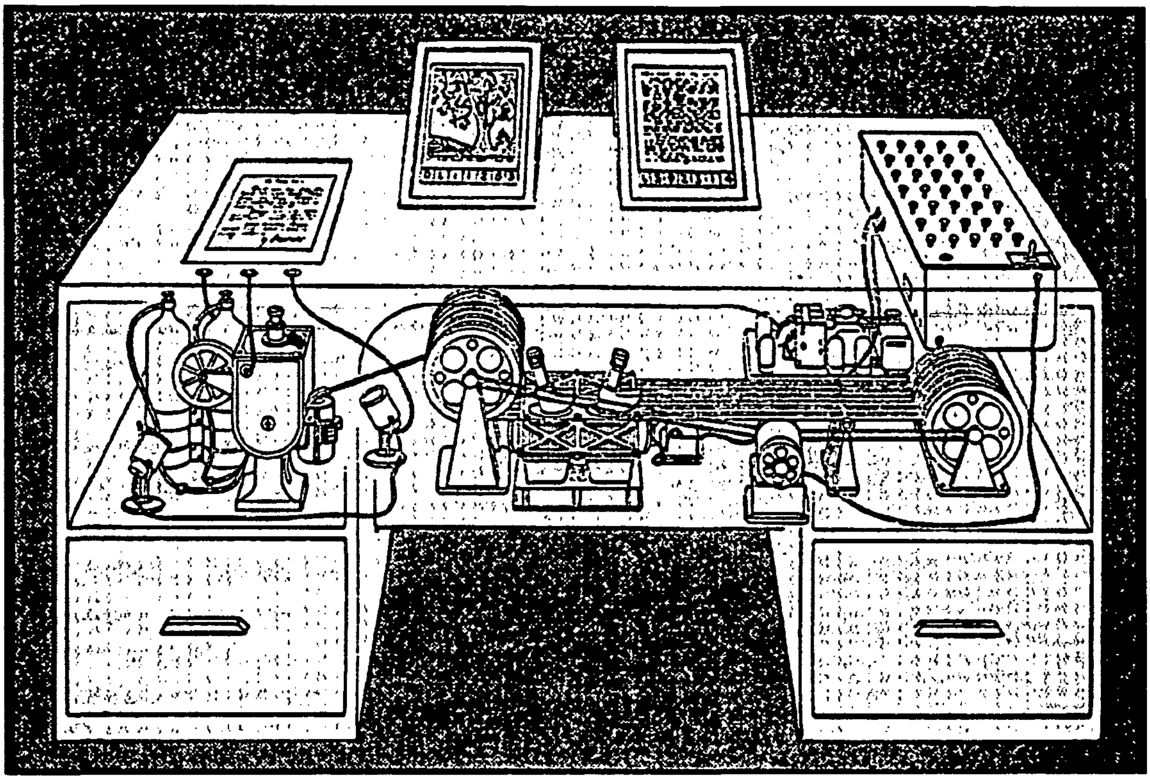


Figure 4.1: Vannevar Bush's vision of Memex

When transforming some of the very outdated technical concepts like microfiches of dry photography, on ends up with a fairly accurate description of a personal computer, complete with scanner/webcam and printer. But that is not the point - the real value lies in the description of content handling a là *Memex*. In order to raise the systems for

[1] http://www.ps.uni-sb.de/ duchier/pub/vbush/vbush.shtml, or
http://www.theatlantic.com/unbound/flashbks/computer/bushf.htm

organising content to the level of development the content they managed had already reached, Bush introduced a new filing concept. Bush characterised the system of filing information on paper alphabetically in cabinets or on shelves as unnatural saying:

> The human mind does not work that way. It operates by association. With one item in its grasp, it snaps instantly to the next that is suggested by the association of thoughts, in accordance with some intricate web of trails carried by the cells of the brain.

Bush's *Memex* would have the capacity to store textual and graphical information in such a way that any piece of information could be arbitrarily linked to any other piece. In his own words:

> He[the user] can add marginal notes and comments, taking advantage of one possible type of dry photography, [...]. It affords an immediate step, however, to associative indexing, the basic idea of which is a provision whereby any item may be caused at will to select immediately and automatically another.

Bush's orientation towards the thought processes of the individual ("as we may think") makes Memex the first radically user centred software design in history.

> Our ineptitude in getting at the record is largely caused by the artificiality of systems of indexing. When data of any sort are placed in storage, they are filed alphabetically or numerically, and information is found (when it is) by tracing it down from subclass to subclass. It can be in only one place, unless duplicates are used; one has to have rules as to which path will locate it, and the rules are cumbersome. Having found one item, moreover, one has to emerge from the system and re-enter on a new path.

Information will no longer be stored according to the rules of taxonomists, but shall reflect the natural though processes. Information units are linked by *trails* and thus blaze a path to knowledge through the vast body of information.

Although most of Bush's ideas about the technical implementation of the *Memex* are somewhat antiquated (dry photography and microfiches - "all this is conventional, except for the projection forward of present-day mechanisms and gadgetry"), the concept of *Memex* still holds. Even todays highly sophisticated information technology leave us wading helplessly in ever growing mires of (non-)information. There still is no efficient way to collect, connect or employ information.

In his article Bush also anticipates "Peer-2-Peer computing" by suggesting that a trail (and all depending information) could be passed on to colleagues and friends for insertion into their own Memex.

4.1.3 Douglas Engelbart and the oNLine System (NLS)

Engelbart started developing the NLS (oNLine System) in early 1960s (Engelbart, 1962), promoting the concept of word processing and real-time interaction with computers. It was first demonstrated at the Fall Joint Computer Conference in 1968 (later dubbed as "the mother of all demos").

The final implementation of NLS included several features that are associated with computing today (such as screen windowing, hypertext links, the mouse), as well as many which have yet to be implemented on the Web: typed links, links to links and automatic multiple views for different levels of users as well as information organized by relevance.

Two visionary fundamental ideas of NLS are: the idea that any document or note or link can be implemented, and thus referenced, as an independent object, and the "document centric" model of documents being accessible by multiple applications rather than belonging to a single one.

Because of its steep learning curve the NLS never spread much outside the SRI (Stanford Research Institute).

4.1.4 The birth of Hypertext

In 1965, Ted Nelson coined the terms "hypertext" and "hypermedia" in a paper to the ACM 20th national conference (Nelson, 1965). In *Literary Machines* (Nelson, 1982), Nelson explained:

> By 'hypertext' I mean nonsequential writing text that branches and allows choice to the reader, best read at an interactive screen.

The first hypertext-based system to see real-world use was developed in 1967 by a team of researchers led by Andries van Dam at Brown University. This first hypertext implementation, HES (Hypertext Editing System), ran on a mainframe computer and was later sold to the Houston Manned Spacecraft Center, which reportedly used it for the Apollo space program documentation.

Andries van Dam and his colleagues at Brown University developed FRESS (File Retrieval and Editing System) after meeting up with Douglas Engelbart and seeing NLS. FRESS made use not only of alphanumeric display terminals but also of a then cutting-edge graphical technology, which allowed the system incorporate windows and vector graphics.

FRESS was the first hypertext system to to be used in teaching: in a poetry course in the mid-70s sponsored by a National Endowment for the Humanities. FRESS was used for over two decades at Brown for personal hypertext libraries and courses and had several commercial spinoffs.

Nelson, who contributed to the development of HES as well as of FRESS, called FRESS it "the first visual word processor" (they are in fact precursors of today WYSIWYG (What You See Is What You Get) word processors) and disapproved of both systems because they centred more on graphical than on structural aspects. Nelson promoted Xanadu (see next section) as the magic-bullet for all document organisation needs.

4.1.5 Far, far away: Ted Nelson and Xanadu

When the first precursor of Xanadu[2] was proposed in 1965 (Nelson, 1965), the system (Nelson, 1982) was impossible to build. Even in 1992 when the WWW just started to gain momentum, Xanadu would have been cutting edge technology. Xanadu was never implemented. In his article in Wired magazine, "The Curse of Xanadu" (Wolf, 2003), Gary Wolf writes

> Xanadu, a global hypertext publishing system, is the longest-running vaporware story in the history of the computer industry. It has been in development for more than 30 years.[...] Xanadu has set a record of futility that will be difficult for other companies to surpass.

Theodor Holm Nelson, a writer, filmmaker, sociologist and software designer is not a technically minded person. This resulted in Xanadu being the first information management tool not geared towards technicians or even software specialists but instead modelled after needs of writers and scientists.

> The original idea was to make a file for writers and scientists, much like the personal side of Bush's Memex, that would do the things such people need with the richness they would want.

In Nelson's own words, "explaining it quickly" (Xanadu, 2003):

- Xanadu is a system for the network sale of documents with automatic royalty on every byte.
- The transclusion feature allows quotation of fragments of any size with royalty to the original publisher.
- This is an implementation of a connected literature.
- It is a system for a point-and-click universe.
- This is a completely interactive docuverse.

The most revolutionary idea of Xanadu is the concept of transclusion. Transclusion provides methods for deep structuring of text and links as objects, allowing version management, re-use and republication, and ownership concepts of these objects.

True to the famous locution

[2]Named after the Xanadu in Samuel Taylor Coleridge's poem, Kubla Khan, to suggest a "magic place of literary memory" where nothing is ever forgotten.

Hardware lives for years, software for decades, but data lives forever!

Nelson's docuverse is completely data/document centric. Nothing will ever be lost, all hyperlinks will survive, all version can be compared. Any change can be reviewed or undone, or redone, or re-undone, or ....

Compared with the WWW which it intends to replace (some day) Xanadu offers a wealth of advanced features:

- Version control and evolutionary development (documents are stored incrementally, nothing ever vanishes)
- Exact linking (every byte is addressable)
- bi-directional links (get to the document that linked to this one)
- Micro-payment
- Transclusion/transcopyright - rights and royalty system included

It is clear, that while Neslon's Xanadu sounds impressive, its implementation is as far away as ever.

4.1.6 Hypertext is here to stay: WWW

Despite the popularity of Apples Hypercard in the private sector, of Asymetrics Toolbook in the business sector, the first hypertext system to receive world wide attention was the WWW (World Wide Web). Within two years after its release in late 1991, it became the world leading online (and offline) hypertext system.

The WWW however, was never destined to be a world comprising information network, it started as an in-house information system for CERN (Centre Europeene a la Reserche Nucleaire) an international high energy physics research centre near Geneva. Tim Berners-Lee and Robert Caillau both worked at CERN. In 1989 they collaborated on ideas for a linked information system that would be accessible across the wide range of different computer systems in use at CERN. At that time many people were using TeX and PostScript for their documents. A few were using SGML (Standard Generalised Markup Language). Tim realized that something simpler was needed that would cope with dumb terminals through high end graphical X Window workstations. HTML was conceived as a very simple solution, and matched with a very simple network protocol HTTP (HyperText Transfer Protocol) and a simple (but graphic) browser called WWW and running on the Next computer system..

CERN launched the Web in 1991 along with a mailing list called www-talk. Other people thinking along the same lines soon joined and helped to grow the web by setting up Web sites and implementing browsers, such as, Cello, Viola, and MidasWWW (all lost in the mists of history). The break through came when the National Center for Supercomputer Applications (NCSA) at Urbana-Champaign encouraged Marc

Andreessen and Eric Bina to develop the X Window Mosaic browser. It was later ported to PCs and Macs and became a run-away success story. It later evolved into the Netscape browser.

The Web grew exponentially, eclipsing other Internet based information systems such as WAIS, Hytelnet, Gopher. Today we see a migration from long established Internet services towards web-applications offering similar functionality: Usenet groups are replaced by web forums, E-mail is accessible via web-frontends and IRC is replaced by web chats.

4.2 Hypertext critique

This section takes a look at possible uses and misuses of hypertext and the problems arising from it.

The early 1990ies saw a flurry of papers (e.g (McKendree et al., 1995), (Thüring et al., 1995), (McKnight et al., 1993)) dealing with the usefulness of hypertext in general and its aptitude for teaching and learning in special. The interest in hyper-learning died down around 1996. Hardly any articles can be found after that time. What had happened? Had people graduated in hypertext? Far from it, but the end of hypertext-critique coincides with advent of the World Wide Web. Household phenomena simply make bad research topics. None of the researchers had banked on curiosity helping readers to overcome the most insidious hypertext structures (after all most early WWW users had already plenty of training on idiosyncratic operating systems and office software). It took another 4 years (roughly up to the year 2000), for hypertext critique to surface again. This time it is major business factor, coming under the heading of "Web-Usability". Any self-respecting computer book publisher now carries at least one book about web usability. Still, it will take some more till the majority of web-sites can be labelled as "user friendly". Heuristics a hypertext (web) usability are discussed in section 4.2.3. For now let us return to question of hypertext mediated learning.

4.2.1 Lost in hyperspace

A lost in hyperspace/hypertext feeling is harmful to comprehension and learning. In his article "The 'Homoeopathic Fallacy' in Learning from Hypertext", McKendree (McKendree et al., 1995) states

> They [hypermedia] certainly offer an exiting and creative way to deliver engaging, effective materials. However, they also offer myriad new ways to deliver confusing, ineffective material.

He goes on to debunk some popular myths on the effectiveness of hypertext such as the much claimed similarity to the brain or the mind. He derives hypertext efficiency from the principle that hypertext visualise dependencies between topics easier than printed, linear text. Another positive contribution is the possibility to rearrange hypertext to ones liking (if such a possibility was included by the author). A hypertext structure that does not resemble the students idea of the matter, will lead to "lost-in-hypertext" symptoms and less effective learning. The use of hypertext to organise information to ones own benefit and ease of use was already proposed by Vannevar Bush in his article "As we may think" (Bush, 1945a).

Thüring et al. (Thüring et al., 1995) analysed "Hypermedia and Cognition" in detail. According to them the two single most reasons for the "lost-in-hypertext" syndrome are *lack of coherence* and *cognitive overhead.* Coherence can be achieved on the local or global scale. Local coherence is achieved by not splitting the "information atom". Information that can not be divided any further, must not be split. This is unfortunately the case by screen orientated system, that require the content to be fit on the screen without scrolling. Global coherence is achieved by aggregating information to units and by providing a document overview (table of contents). Cognitive overhead is generated by orientation and navigational problems. Orientation is improved by marking the current position and the path that lead to it. Navigation is greatly helped by typed links. Types can be logical (more detailed information, example, exercise) or spatial (previous, next, up, down).

Even if these guidelines sound trivial, many *e*-learning environments are far from heeding them. Many of those (e.g. blackboard (Blackboard Inc, 2003)) do little more than connect word-processor documents by hyperlinks. This of course promotes neither deeper inside in the dependencies, nor allows for making new connections.

4.2.2 Hypertext structure and Hypertext design models

Although there is a limitless amount of actual hypertext structures, they all can be derived from four basic types: linear, hierarchical, network and semi-structured. See figure 4.2 for a graphical representation of these types.

linear Linear hypertext is like a children's picture book or a novel: on page after the other. This is certainly, not the format that usually springs to mind when thinking about hypertext. Linear "hypertext" is a change of transport medium. A linear structure is the optimum for learning content presentation (provided the content is presented in the right order), but not always achievable. However care must be taken not to "cage" advanced students in the linear course. All pages should be accessible directly.

hierarchical Hierarchical hypertext is like a college textbook: chapters, sections, subsection, ... and a table of contents. The hierarchical presentation has along tradition in teaching and science. As old habits die hard, many educational hypertexts end up in hierarchical format. Again this is just a change of transport medium, not "real" hypertext.

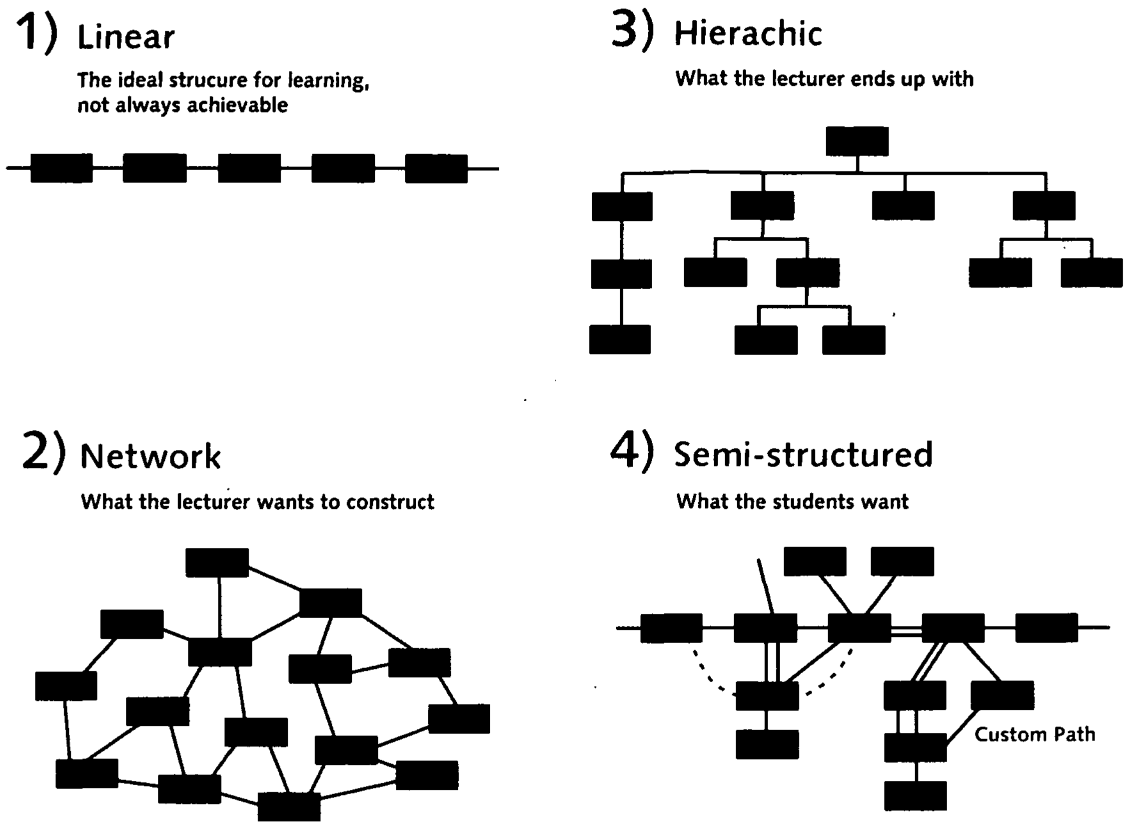


Figure 4.2: Hyperstructure – Hypertext geometry: four basic types of hypertext structure: linear, hierarchical, network and semi-structured.

network The network structure is usually associated with hypertext. Every node is connected to various other nodes, they form an organic whole. There is neither a clearly defined entry nor exit point. Many complex phenomena can only be described in network structures. A network structures offers a maximum of freedom to the student a will provide the maximum of confusion as well. Teachers subscribing to the school of constructivism aim at producing network structured hypertext, but frequently fail to achieve this form and finally end up with another hierarchical presentation.

semi-structured The semi structured approach combines the best of all worlds: linear, where linearity is necessary, hierarchical where dependencies are essential and networked by a plethora of hyperlinks both inside an outside the given material. A semi structured hypertext offers the student guidance, while allowing him to follow links at leisure and make new connections.

As hypertext evolved, several design models were created to describe new and existing hypertext structures and designs. Some of the more well known models include the HDM (Hypertext Design Model) (Garzoztto et al., 1993), the OOHDM (Object Oriented Hypertext Design Model) (Schwabe and Rossi, 1995) and the RMM (Relationship Management Methodology) (Isakowitz et al., 1995). An overview of these and some of the lesser known models is given in (Koch, 1999).

While all these models are valid and represent structured approaches to the design of hypertext applications, they are far to complex to be mastered by laymen. On the other hand, an experienced hypertext designer will feel bogged down by the process. These models also are not suitable for designing frameworks for hypertext design. However it becomes clear, that hypertext design is too complex to be left to laymen. A (*e*-learning) authoring tool should let the author concentrate on what he knows best content. The generation of structural elements such as navigation is best left to framework.

The XML (eXtensible Markup Language) revolution starting at around 1996 put many of the hypertext design models and their CASE (Computer Aided Software Engineering) aspirations out of business. XML allows the definition of custom markup languages. These languages are tailored to the specific needs of the application and leave little leeway to the author. The resulting documents can be tranformed and displayed by standard tools and processed by server application to add navigational support.

4.2.3 Hypertext usability

Hypertext and web usability is finally taking hold, but still there is a large number of hypertext projects showing less than optimal usability.

Usability processes can usually be broke down to three major stages (see figure 4.3): *analysis*, *design* and *implementation.* These stages form a closed, circle as it is evident that usability can always be increased (at least when considering a normal software life time of roughly 5-10 years).

Each stage is associated with various different usability engineering methods. While some methods such as user and task analysis (see (Constantine and Lockwood, 1999) for details) belong firmly to the analysis phase, one method – the *heuristic evaluation* – can be used in every stage. Heuristic evaluation refers to experts judging a system, a design or a prototype according to a pre-agreed set of guidelines (heuristics).

Jakob Nielsen was one of the forerunners of hypertext usability movement and pioneered low cost usability engineering and heuristic evaluation. He shortlisted ten of his heuristic guidelines to form a his well known and much cited list of "Ten Usability Heuristics" (Nielsen, 2003b):

Visibility of system status The system should always keep users informed about what is going on, through appropriate feedback within reasonable time.

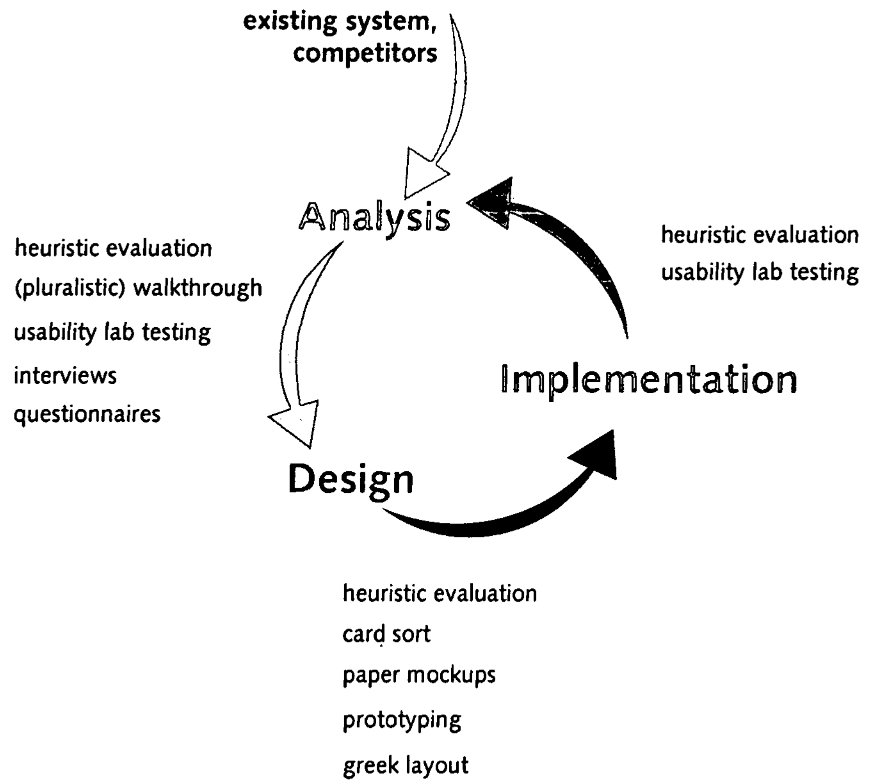


Figure 4.3: Usability engineering phases and methods used therein. Heuristic evaluation is one of the most powerful methods. It can be used in every phase and requires nothing more as some people with moderate usability skills and a proper checklist.

Match between system and the real world The system should speak the users' language, with words, phrases and concepts familiar to the user, rather than system-oriented terms. Follow real-world conventions, making information appear in a natural and logical order.

User control and freedom Users often choose system functions by mistake and will need a clearly marked "emergency exit" to leave the unwanted state without having to go through an extended dialogue. Support undo and redo.

Consistency and standards Users should not have to wonder whether different words, situations, or actions mean the same thing. Follow platform conventions.

Error prevention Even better than good error messages is a careful design which prevents a problem from occurring in the first place.

Recognition rather than recall Make objects, actions, and options visible. The user should not have to remember information from one part of the dialogue to another. Instructions for use of the system should be visible or easily retrievable whenever appropriate.

Flexibility and efficiency of use Accelerators – unseen by the novice user – may often speed up the interaction for the expert user such that the system can cater to both inexperienced and experienced users. Allow users to tailor frequent actions.

Aesthetic and minimalist design Dialogues should not contain information which is irrelevant or rarely needed. Every extra unit of information in a dialogue competes with the relevant units of information and diminishes their relative visibility.

Help users recognize, diagnose, and recover from errors Error messages should be expressed in plain language (no codes), precisely indicate the problem, and constructively suggest a solution.

Help and documentation Even though it is better if the system can be used without documentation, it may be necessary to provide help and documentation. Any such information should be easy to search, focused on the user's task, list concrete steps to be carried out, and not be too large.

This list was originally designed to give guidance on GUI (Graphical User Interface) design but proves equally valuable for web-design or the design of *e*-learning environments.

When dealing with *e*-learning environments it is most important to convey a sense of place ("Visibility of system status "). On every learning topic must therefore express the unit and course it belongs to. It level (essential, important or additional information) and possible connection to other topics. If content is strapped from context, learning will not take place efficiently. "Consistency and standards " are also very important. Therefore all learning topics should follow the same master plan, e.g. by having the same subdivisions like title, abstract, text body, literature, examples and ,exercises.To provide an efficient interface ("Flexibility and efficiency of use "), the system should adapt (automatically) to the student's needs by offering the most wanted links or listing pages recently viewed.

Much off these essential heuristic guidelines can be incorporated in to a framework, thus ensuring that no part of the system can violate them. The *e*-ULE framework proposed in this thesis will ensure a "sense of place", consistency and efficiency of use. Minimalistic and aesthetic design is provided by built in templates. It will also help author to write better content by giving hints readability and scanability and error prevention. Detailed information on (hypertext) usability can derived from (Nielsen, 1993), (Mayhew, 1999) and (Nielsen, 2003a).

4.3 On the shoulders of giants: Towards a new hypertext system

Any attempt at todays hypertext design should start with readily available concepts: to ignore the omnipresent structures provided by HTML and the WWW would be foolish indeed. A new *e*-learning orientated hypertext environment should leverage existing infrastructure and build upon already existing frameworks. See chapter 8 for details about the technical implmentation.

4.3.1 Analysis of the WWW concept

Before progressing to a new hypertext system let us analyse the status quo and discuss the advantages and discadvanteages of existing systems:
Advantages of the WWW/HTML System

+ Widespread
+ Many tools available (browsers, editors, ...)
+ Extensible via plug-ins, mime-types, external programs
+ Fairly standardised, open standard

Disadvantages of the WWW/HTML System

– Essentially a one-to-many communication system
– Lacks version control
– Informations kept in the WWW is too volatile (citation problems)
– Lacks support for identification, authentication and rights management
– Lacks typed hyperlinks
– Default language (HTML) is too limited for academic/scientific use
– Poor handling

The advantages speak for themselves, the disadvantages bear closer analysis.

One-to-many communication system

The WWW is basically a one-to-many communication medium. A person publishes text on a server to be read by others. Collaborative editing which is frequent in the academic setting is not supported. Collaboration requires third party tools like a

CMS (Content Management System). This is in stark contrast to the Wiki (Leuf and Cunningham, 2003) System where any number of people can work on the text.

Version control

Unless the author of the information specifically add some sort of version information, there is hardly any support for versioning in the WWW other than the possibility to query the last change date (and this is not a compulsory feature).

Ted Nelson's Xanadu proposes a very tight versioning, where any state of document can be reproduced. Aside from the technical problems in developing such a system, such a feature might not be required. A practical versioning control should provide the possibility to manually create versions an tag them with comments and change information, much in the way of the CVS (Concurrent Version System) (CVS Organisation, 2003), (Subersion Group, 2003).

Version control can be easily implemented on the server side. Additional information can be coded into the documents itself.

Volatile Information

One of the WWW's taglines might be "Here today, gone tomorrow". Information on the WWW has the tendency to magically disappear. Even if the information stays basically in the same place (e.g on the same web server), a simple renaming of the file or reorganisation of the file structure might make it difficult to find the information again.

Dead links ("linkrot") are a major source of annoyance on the WWW (Nielsen, 1998), so annoying in fact, that a popular search engine (Google Inc., 2003) caches the documents in its database in case they have gone missing in the meantime. Again Neslon's Xanadu (the place were data lives forever) and its bi-directional links are far ahead of time.

Bi-directional links can be implemented by server to server messaging. As a matter of fact, the basis does already exist. HTTP status codes (IETF, 2003) provide various messages for different cases of misplaced documents, but the response is only sent to the client, no to the referrer. By collecting referrer addresses in a database, the referrer could be notified of changes.

Identification, authentication and rights management

Since everybody can publish on the WWW (and its basically good that anybody can do so), the authenticity of a specific item can not always be assured. Identifying the author of a document can be tricky, even though mechanisms (digital signature, ... (OASIS, 2003; Alliance, 2003; W3C, 2003)) for ascertaining the identity of the publisher exist.

Quoting web-sources is difficult because of the volatile character of the medium. To avoid problems with dead links the material is frequently copied and saved locally, a process which sometimes fails to preserve the original link. In Xanadu Ted Nelson envisioned trans-clusion and micro-payment. Bases on todays web-infrastructure neither is easy to implement.

Typed hyperlinks, typed content

HTML Hyperlinks are not typed [3]. Typed links are the basis for the "thinking" or semantic web, as they allow humans and computers alike to grasp the context of the information.

> Most of the structuring forms I'll show you stem from the simple capability of being able to establish arbitrary linkages between different substructures, and of directing the computer subsequently to display a set of linked substructures with any relative positioning we might designate among the different substructures.
>
> You can designate as many different kinds of links as you wish, so that you can specify different display or manipulative treatment for the different types.(Engelbart, 1962)

Some parts of Doug Engelbert's vision have been realised by using external programs and mime-type. The vast amount of text based data is still untyped. Type support in HTML was weak from the beginning and deteriorated with the ascent of WYSIWYG Editors.

HTML

As the rest of WWW technology HTML was originally designed for a in-house information system. Considering that HTML was developed at CERN, its lack of support of markup for scientific purposes has always been puzzling. HTML supports neither mathematical nor chemical formulas or musical notations. Natural scientist took various ways out of the formula trap: some simply wrote their formulas in plain LaTeX, others turned to HyperTeX, quite a lot only used postscript or PDF files or converted their formulas to bitmap graphics (and thereby lost all information inside the formula).

The lack of a separated presentation language or at least tags of content presentations has lead to misuse of the existing tags, like the usage of table tags layout pages.

Since the re-adoption of SGML in guise of XML, specialised markup languages like MathML (Mathematical Markup Language), CML (Chemical Markup Language) or MusicML (Music Markup Language) are advancing, but as of now (early 2003) there is hardly any browser able of natively displaying any of these languages.

[3]Except for the mailto:type, which pops up the email client

Poor handling

Comparing todays "connected workspace" with Vannevar Bush's Memex, Memex comes out on top. Todays web clients are easy enough to handle for beginners, but they lack in advanced features and thereby cripple the usefulness to the advanced or professional user.

Todays browsers are capable of displaying web-sites, but browsing has been degraded to tv-like "zapping". Advanced functions like proper bookmarks or powerful history functions or offline storage, history search or advanced caching are only available as third party tools, difficult to find, install and integrate. To the researcher the WWW is nothing more than 7x24 library, albeit one without a proper catalogue.

Storage classification and reuse are not supported by this generation of browsers. To leverage information science for your research you still have to become a computer science expert.

4.3.2 Educational hypertext concept

An new educational hypertext system should try to combine the best features of all existing concepts. As pointed out in section 4.2.2 a semi structured approach to hypertext will proof best for educational purposes. A *hierarchical skeleton* will provide guidance and a clear overview of the subject matter, while additional hyperlinks cross reference the various topics and thus allow for an explorative approach to the subject matter. On very complex matters, alternate hierarchical skeletons (tables of content) can be provided to provide a totally different access to matter at hand. Typed *bi-directional typed* links will help in pointing out the next step.

On the user side the system should provide detailed *history* and *bookmark* functions and good *search capabilities.* The student should have the possibility to *annotate* and *restructure* the material to his own liking.

On the authors side the system should provide a (XML-based) *language* adapted to the needs of the *educationalist* and a version control system. The system should also help the author in *linking* his texts, by providing possible choices and ensuring the *validity* of all hypertext links even in case of renaming or deleting.

For a more detailed discussion of the use of hypertext in the *e*-ULE system see sections 7 and 8.

Chapter 5

Students' needs

Why, a four-year-old child could understand this. Someone get me a four-year-old child

Groucho Marx

Although university *e*-learning systems are per default geared towards students, surprisingly little thought has been given to students needs. As *e*-learning systems are mostly purchased by university administration, who frequently fail to take lecturer needs into account it is not surprising that the needs of students who are twice removed are not considered. It is commonly considered that providing "modern" materials (e.g. Internet based materials) is enough to satisfy all needs.

5.1 Survey

Following our research into the reasons for skipping lectures, we interviewed students (Naber, 2001),(Naber, 2002b) on their preferred ways of learning an their feelings on *e*-learning (classical distance learning, CAI, WBT, ...) as well as their potential uses of such study aids and the features such a system must/should/could provide.

The first round of the survey was questionary based, but a test survey showed, that all features are desirable. Therefore the format was switched to individual interviews, which highlighted several "must have" features and shed some deeper light on the actual uses of common features, like bookmarks or hypertext.

5.1.1 Questionnaire and interview guideline

Both the questionary and the interviews centered around the following topics:

- Previous experiences: type of *e*-learning , product, topic, rating

- Interest: replace lecture notes, replace lecture(er), augment lecture or lecture notes
- Feature evaluation: liked/disliked
- Feature evaluation: would like to have ...
- Feature evaluation from list

5.1.2 Results

Demography

The interviews comprised roughly a hundred junior and senior students from a wide range of fields. The participants where mostly gathered from university sports and around libraries and the farther circle of friends.

Most of the interviewed students studied in Vienna. This should not pose a restriction, as the results are not likely to vary from University to University, aside from the fact, that students from less frequented studies are overall more content with their study situation. Smaller universities normally have fewer students per lecturer, thus less complaints are voiced.

Experiences

All of students had previous experience with computer mediated learning[1]. In all cases this experience included language learning tools. This is in phase with the assumption, that adult language learning consists mostly of memorising and can be supported by *e*-learning tools following the behaviouristic concept. These systems can be easily computerised and multimedia can in fact provide additional features, such as native speaker sound or computer based pronunciation control.

Several students claimed experience with CD-based reference works (e.g. Dubbel Taschenbuch für Maschinenbau, or Pschyrembel Klinisches Wörterbuch) or educational calculation software (e.g. Teach Me Data Analysis, or specialised MathCAD Workbooks). Literature students frequently use literature CD-ROMs. These products are specifically geared towards the academic environment.

Although basically classed as "useful", the CD-based reference works often fall behind the hopes of their users: unnecessary technical constraints, lack of interactivity ("book-ware") and unstable behaviour are frequently cited.

Technical problems and constraints plague many of the CBT products: the further use of the contained information outside the software is often hindered due to copyright reasons[2].

[1]Note: Previous experience was NOT a prerequisite for the interview!

[2]Or maybe nobody thought, that the users might want to print the information to read in on the train.

About half of students picked up "Edutainment" CD-ROMs vaguely pertaining to their field of study. In most cases, these products are aimed at high-school students or interested amateurs, but offer too little in-depth information and – even more annoying – lack necessary scientific meta information like citations.

Nearly all students in the technical field and several from non technical fields of study had access to lecture web-sites, which offered materials for download. The download of materials is mostly preferred over the option of buying the lecture notes at the department office.

The Vienna University Department of Psychology, used to ran open discussion boards for their students. These discussion boards, were originally born out of the need to administrate 8000 to 10000 students with a staff of only 40 people. None the less, these boards were extremely popular among the students, as well as among the lecturers, as the provided new and creative insight. However the overhead of monitoring and moderating these boards, proofed too high and now the access is restricted to students actually enrolled in the course. The students set up an alternative (completely unmoderated), but this failed due to a too high noise to information ratio.

The experiences vastly differed among the students (the best marks were given to language learning tools), even most of the disappointed students keep an open mind. In fact it's a common believe that the "*e*-Nürnberger Trichter" is just around the corner.

Enthusiasm

Considering the mixed experiences, it is suprising, that most students are enthusiastic about trying new forms of *e*-learning .

This reflects a common central European attitude, that anything can be solved by technical means. Central Europe (and the German speaking countries especially) are technically driven, when compared to the rest of Europe or the rest of the world. Technical solutions will be implemented simply because they can be implemented, not because they are required.

Information transfer by a lecture was generally perceived as anachronistic. The result of the questionary regarding the reasons for skipping the lecture were confirmed: again the students confessed their preference for lecture time to be given up to discussion and *question and answer* sessions.

A few students claimed to have no interested at all in *e*-learning . Interestingly, all of these were computer science students. They argued that after spending much time behind a computer for practical work, they did not want to use a computer to learn the theoretic aspects as well.

Their preferred learning styles included group learning, verbal presentation and highly structured and organised material. There preference of direct oral communication, coupled with a field of study stressing independent work and learning from books, makes them unlikely candidates for *e*-learning . Felder (Felder et al., 2002) investigated the influence of teaching methods on the student performance in dependence of the student MBTI (Myers-Briggs Type Indicator) style. This study found a performance increase for students of the extravert, sensor feeling style, which had in previous studies been found disadvantaged in engineering education.

Still, these students were all in favour for a well organised lecture site as download spot for the lecture notes, as a preview for the lecture's contents and a spot to discuss the topic in question with other students.

Material usage

Several of the students observed that their use of learning material changes according to the "learning phase". Learning about the material or the course requires a different structure, than learning for the exam or accessing the material for reference purposes. The five learning phases and their specific requirements are detailed in section 5.2.

Useless features?

When questioned about their experiences and the features they liked or disliked most, several of the interview partners discarded features as

- full text search
- bookmarks
- hypertext(!)

When asked why they turned down these features, the replied that their experience showed that "these never worked", and thus they were probably a "complete waste of time to implement" and the "poor, overworked lecturer should not waste time in trying to get them working". Of course everybody was in favour of working full text search, bookmarks and hypertext. This highlights a common computer usage problem: as pointed out by Shackelford (Shackelford, 1990) users are so brainwashed by the myth of the "smart computer" that they will blame themselves if they encounter stupidity on the software's part. Instead of questioning the design of the software they assume that nothing can be done about it.

Students seem to care a lot about the efforts their lecturers have to undertake to produce usable *e*-learning . This seems surprising, as most of the reasons for skipping lectures pertained to bad lecturer performance. Obviously, effort and result are not the same. Students value personal engagement of a lecturer higher than skilled presentation. A lecture manuscript (no matter how disorganised or error prone) will probably garner a higher rating than a book by somebody not directly involved in the lecture. This is even more true for social studies, where personal opinions are even more a factor, than in natural science or engineering.

A feature to use the online version (or an offline rip of this version) on a mobile device was mostly discarded. Nobody considered these devices ripe for such a task. About half of the students conceded that this would be an interesting option once the devices reached a certain maturity.

Personal styles certainly have strong influence on the usefulness of certain features. A person who prefers verbal/oral presentation will complain if a literature CD-ROM does not feature a literature reading, whereas somebody preferring visual input will scorn the very same feature and demand still photographs of the theatre production or film sequences.

5.1.3 Most wanted features

The interview also addresses the question, which features were the most desirable. This question was divided in free form answer part, in which the students had to name features of hand an structured part in which suggested features could be rated.

Interestingly enough, some features that had been marked as "useless" in the previous question about not needed features, reappeared among the "most-wanted" features.

The most wanted features according to their ranking are listed below.

Online discussion

Nearly all students were in favour of discussion groups. Obviously one of the reasons to attend lectures is to stay in touch with colleagues. The virtual meeting room is even more popular with commuting students.

In addition nearly all of the students use IMS (Instant Messaging Systems) tools (ICQ (I Seek You), MS Netmeeting, Yahoo Messenger, AIM (AOL Instant Messenger), and others) as means of communication with friends.

Online discussion can be separated into *asynchronous discussion* that takes place via mailing lists, news groups and web-based discussion groups (forums) and *synchronous discussion* that take place via IMS (including video conferencing), IRC (Internet Relay Chat), or web-based chat rooms.

In the interviews students showed equal interest in participating in forums and in chat rooms, but a newer study regarding the usability of online discussion groups (forums) and chat rooms (Holmgren and Tomitsch, 2003) found that chat room would not appear to students as a means for problem solving, but rather as a means for socialising.

Students prefer moderated open discussions groups, university departments, however do not: the moderation of an open discussion group is too time consuming (Werneck, 2003). Completely unmoderated groups tend to be flooded by spam and will die from neglect. The easiest way to ensure that discussion groups stay mainly on topic seems to be to restrict access and authenticate users.

Print version

Another "must have" is a printing feature. The inability to print or otherwise reuse material was a frequently cited annoyance among CBT users. Even in the digital age, information on paper is still useful. Printed information can be read on the train, in the park or even in bed. Learning habits vary, so should the printing options.

The printing should be flexible and allow for various printing methods: full text, partial text: with-/without annotations, discussion group items, examples, exercises, self-tests, or other content.

Annotations

Annotations play an important role in student learning. Conventional lecture notes are normally augmented by scribbling on the borders or between the lines and by inserting pages. This approach is naturally limited.

Electronic media allow for the seamless integration of annotation in the text body. The annotations should support the same markup possibilities as the original text. To maximise their usefulness annotations should be included in features such as printing or searching.

An interesting option is the possibility to make private or public annotations. The students favoured a possibility to make private annotations as well as public ones.

Intelligent search function

Search is one of the features students had bad experiences with. Good searching is difficult to implement and requires interaction on part of the author. A simple full text search will rarely do good, especially if there is no relevance ranking included. Even a search engine with relevance ranking, might not be sufficient, if the search item appears too frequently, or is a synonym of the word.

A search engine for a comparatively self-contained information block (like lecture notes), can include references to the history ("I know, I read it last week ...", "I came across it when I was reading about ..."). Search engines should include annotations and other augmented content in their search.

Excerpt tool

A frequently mentioned feature was the possibility to rearrange the material to one's own liking and the generation of "sub notes" which might serve as an additional, direct path to a special topic (compare Bush (Bush, 1945a)). Rearranging or excerpting text is a frequent occupation in the third learning phase (see 5.2.3).

Based on a semi-automated keyword and linking system, the excerpt tool might also support just-in-time learning by automatically resolving dependencies and thus support phase 5 learning (see 5.2.5).

5.1.4 Feature list

Aside from the above mentioned most-wanted features, the following features were put on trial:

Hypertext

Hypertext offers the possibility to lookup definitions or to check out necessary data without leaving the current working environment. Hypertext can connect topics in new, refreshing ways and provides different views on the same subject matter. According to cognitivist and constructivist learning theories, learning takes place by interconnecting. Hypertext applications therefore should be the ideal representation of facts and information.

That fact that many students claimed hypertext to be useless, can be explained by the big number of "previous – next" pseudo-hypertext applications that can be found in CBT/WBT.

Continous updates

Digitally available material can be easily modified, thus errors can be quickly corrected. Results of discussions during the lecture could be incorporated into the lecture notes. The notion of "self-correcting" lecture notes enticed most students, the prospect of a continously growing body of in formation did not raise as many supporters. Students prefer stable situations and like to know the amount of information to be processed in advance.

A minimum requirement for the usage of growing lecture notes, is a flagging system (or some other notification system) that alerts the students to changes.

Bookmarks

Good bookmark implementations a rarely seen. Once the number of bookmarks reaches a couple of dozens, electronic bookmarks (as well as real live bookmarks) become unwieldy. Todays browsers offer only sorry excuses for book-marking, they mostly lack integration with the history function and search/indexing capabilities.

Thus, it is not surprising that bookmarks made it onto the list of "useless" features, the existing implementations are unsatisfying.

History

The history function was perceived as more useful compared to the bookmark function. A history functions lists the places visited today, yesterday, this week, ... A history can be very helpful in navigating large information spaces.

Frequently used pages

User tracking enables the server application to asses the frequency of page calls. Pages that are frequently called up (most likely reference pages like a periodic table) are grouped in special menu for quick and easy access.

This feature was very well received, and as it is quite easy to implement, it is surprising that it is not in widespread use.

Examples

Examples illustrate the practical use of the learning content. Examples should start with a problem and provide a detailed, annotated solution. Examples are necessary for students with a executive, local, dependent or sensing style and those with a haptic preference. All others will also profit from detailed examples.

Although the students were eager to have examples included in their lecture notes, only a few mentioned examples spontaneously in the interview. Examples are seen as integral part of lecture notes, not as a separate "goodie".

Exercises

In contrast to examples, which are merely illustrative, exercises should engage the student in further thinking. As *doing* is always more strenuous, than *following*, exercises, naturally are not too popular with the students. Still, the students realise the need for exercises, but prefer to have some pointers (hints) and a final solution to be included. Again exercises were not seen as a separate entity, rather as an extension of learning content.

Self-tests (with solutions)

Self-tests (multiple choice or free form, with solutions) were demanded by various students. Although these tests seem somewhat childish (as they resemble some of the more primitive CBTs, were you only are allowed to see the next chapter after you have successfully mastered the test), passing the test seems to reassure some of the students.

This feature was commonly regarded as an "extra goodie", although the difference between an exercise with solution (see previous section) and self test is very slim.

Frequently asked questions (FAQ)

FAQ were warmly received, but mostly regarded as pertaining to the usage of the *e*-learning system. FAQ are ambivalent feature: neither well designed *e*-learning systems nor well designed learning content should require FAQ, but foreseeing every possible type of non-understanding is nearly impossible. The greatest disadvantage of FAQ is their need for maintenance. Even tools like faq-o-matic (Howell, 2003) cannot lift the burden of identifying the problems-pots and generating the answers. More promising candidates might be Wikis (Leuf and Cunningham, 2003) or annotations in the content itself.

Links to related sites

Hardly mentioned, but much acclaimed were links to related topics. The lecturer (or the students themselves) should be able to add links to other interesting pages on the web.

The main problem with links to external content is the "shortvity" of hyperlinks. Technical support is needed to avoid cemeteries of dead links. The most promising method is caching (keeping a copy of the original file), but it might not work with interactive sites and there are unresolved copyright questions. Am minium requirement is the automatic surveillance of the links. In case of a dead link, the system should notify the administrator and mark the link as inactive.

Predefined paths

Most university level information is highly complex, a single linear path through the information might not be possible or the best solution. Paths connecting individual topics in unique ways, can highlight the information available and put it into context.

Let us take this thesis as examples: its network structure makes it unsuited for strictly linear presentation. The information herein could be presented from the students point of view (student requirements → *e*-ULE support for learning styles), or from the lecturers point of view (lecturer needs → *e*-ULE scientific workflow support), or from a technical point of view (how do *e*-ULE features facilitate university learning). Whereas the first two views can be merged into one linear view on *e*-ULE , it is difficult to incorporate the third view into a linear structure without repeating much of the information presented in previous chapters.

Fancy printing

Aside from the basic requirement of printing, students were in favour of "fancy", specialised printing styles. Especially students from the fields which require extensive definition/fact learning (like medicine, pharmacy, chemistry) wanted a possibility to create flash cards from the lecture notes. Other print formats include: summaries, keyword listings and sub-notes (compare 5.2.3).

Other features

Some of the features mentioned by the students are not desirable from the lecturers view. An exam questions exchange was frequently cited, as was the possibility to contact the lecturer via *e*-mail. Whereas direct student-lecturer communication via chat or email is probably out of the question, a help-line system could be installed, provided the department has enough staff to run it. The help-line could also be realised as a moderated discussion group, were senior student tutors answer the questions, that could not be answered by peers.

5.2 Support of different learning phases

In individual interviews we discovered 5 separate learning phases, which each possess unique requirements. To provide added value to the students a learning environment should take the different learning phases into account. Table 7.1 in section 7.3 lists the stages and the students' tasks involved plus the support offered by lectures, printed lecture notes and *e*-ULE notes. Figure 5.1 depicts the learning phases in context of lecture attendance patterns.

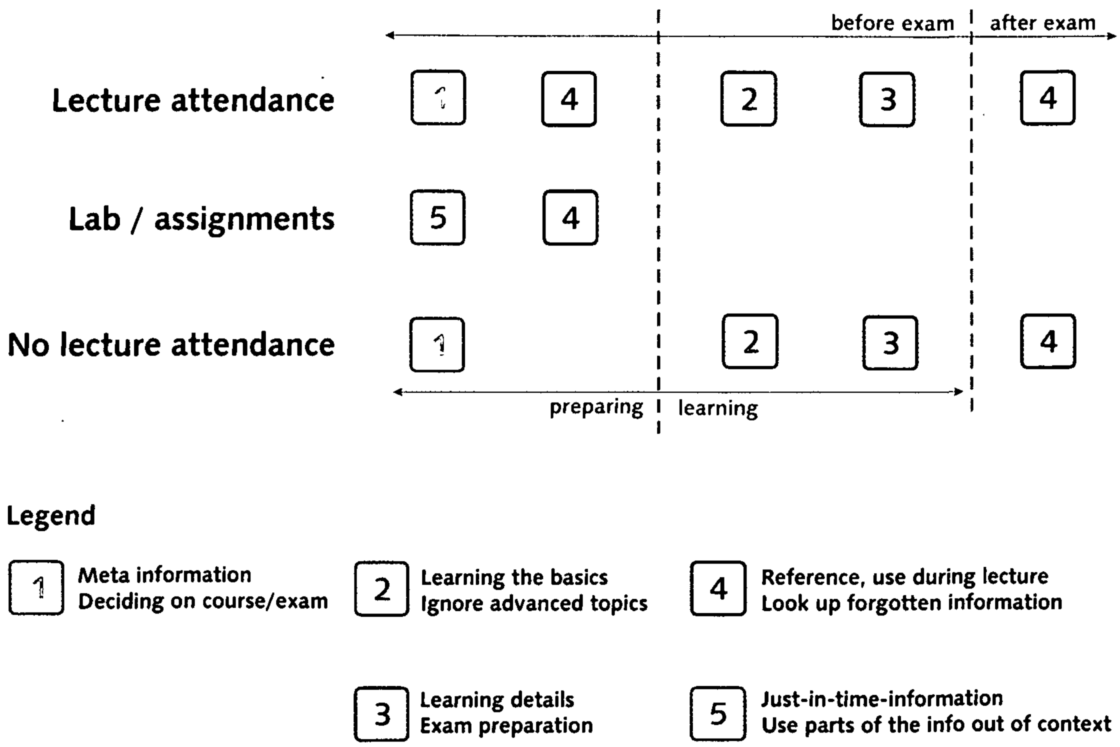


Figure 5.1: Learning phases in context: typical sequences of learning phases for the cases: lecture attendance, no lecture attendance and attendance of labs or solving assignments.

5.2.1 Phase 1 – reconnoitring

Before deciding on taking a course or an exam the students want a detailed overview of the topics covered. Some students referred to is as the "90 minutes notes" or "executive summary".

Information for this phase should include a list of necessary prerequisites, a list of goals for the course, a padded table of contents, summarising the information of each chapter and stating an approximate time frame.

Many courses lack this type of information. Sometimes this information is transported via the departments web-site of an university provided course directory. See section 7.3.3 for *e*-ULE's support for phase 1.

5.2.2 Phase 2 – learning the basics

After having decided to take the exam most students read through the material available (and try to solve the exercises). In this stage they try to focus on the main points. Many lecture notes do not differentiate between *essential*, *important*, and *additional* information.

Support for this phase should include: identifiers for the relevance of the topic, examples, exercises and solutions, test to ensure that the material has been understood.

Though commonly found in newer textbooks, relevance/level identifiers are rarely found in lecture notes. More likely the notes will lack all advanced information, than provide information on the level of the included topics. Support could be improved from the student's point of view. See section 7.3.4 for *e*-ULE's support for phase 2.

5.2.3 Phase 3 – exam preparation

Detail-learning for exam preparation differs vastly from the learning strategies in phase 2. Incomprehensible sections can not longer be skipped, missing prerequisites need to be filled in, complicated passages need to be annotated and internalised information can be removed from the learning cycle.

To facilitate all this, most students make custom notes of the topics they need to study further. Classic lecture notes provide no support for this stage, all excerption is done by hand. Although these excerpts can be of great pedagogical value, they cannot efficiently be integrated with original material and they are likely to contain errors.

Technology can help a lot in this phase: discussion boards and online chats can help in clarifying concepts. Electronic material can be seamlessly annotated and annotations can be shared with other students. Materials could be rearranged, or shortened to fit the student's learning needs. See section 7.3.5 for *e*-ULE's support for phase 3.

5.2.4 Phase 4 – reference

Lecture notes should not be worthless after the exam. Human memory being as it is, it is often necessary to return to previous "haunts", to look up what has been forgotten. While most books come with a somewhat decent index, most lecture notes do without, so information retrieval can be quite tedious.

"Too difficult" is a commonly cited argument against including an index, as is "Buy a reference work". True enough, index generation remains one of the trickiest parts of document engineering. While modern word processors are equipped with such a function, the results often disappoint. As regards the suggestion, that the students should buy a reference book, if they want to look something up, it does not hold, as the students spent quite some time with the notes and feel familiar with the structure, terms, notations, ... used therein. A switch to another source of information, just for reference purposes, will not lighten the information blight.

Phase 4 is not limited to after exam stages. Using the notes parallel to the lecture or using it to prepare for labs or to solve assignments follows the same overall usage pattern. See section 7.3.6 for *e*-ULE's support for phase 4.

5.2.5 Phase 5 – selective retrieval – just-in-time information

This phase is set apart from the other four phases: without detailed knowledge on the provided information the student wishes to retrieve selective parts thereof which might be required for a different subject matter. Chances for solely extracting *relevant* information are slim. Prerequisites to later chapters tend to be scattered over various chapters, forcing the student to completely canvas the material.

Although this scenario sounds farfetched, it is far more common. In fact, a large proportion of student learning takes place in this way. University curricula are weakly linked at best, so flexible structures and no real sequence of lectures soon lead to students sitting in lectures without the necessary prerequisites (which most of the time were not specified – see phase 1). The student, forced to either drop out of the course (and maybe loose a year) or somehow scamper along, will usually decide to patch together the necessary knowledge. Just-in-time-information need also arises in the process of preparing for lab exercises or in solving assignments.

Including support for this learning scenario in a conventional lecture note setup is difficult. It requires a strict notation convention (preferable with a list of conventions used), cross-references and the possibility to place lecture content into small, independent boxes. See section for *e*-ULE's support for phase 5.

5.3 Style and method

The effectiveness of course elements is influenced by the students style portfolio. Different types respond to different elements. An overview of various thinking and learning styles and their correlation is given in table 5.1. Thinking and learning styles are discussed in chapter 3.

External students will respond better to discussion and group activity than *internal* students. However, the use of a web-based discussion board might entice internal students, who otherwise would be to shy to act. Anonymity in the discussion board should therefore be ensured.

Students of a *legislative* or *liberal* disposition prefer free or creative exercises, whereas those of *conservative* or *executive* inclination rather stick to preprogrammed exercises. Students with a *judical* inclination like to compare and rate concepts and are likely to

	legislative	executive	judicial	monarchic	hierarchic	oligarchic	anarchic	local	global	internal	external	conservative	liberal	visual	verbal	haptic
	function			form				scope		level		leaning		NLP		
Lecture	–	+	O	+	O	+	+	+	–	+	–	+	–	–	+	O
Books/Notes	O	–	O	+	+	O	O	–	+	+	–	+	+	+	–	–
Q&A session	O	O	O	+	+	+	O	+	O	O	O	O	O	O	+	O
Discussion	+	–	+	+	–	+	O	–	O	–	+	O	+	–	+	+
Online discussion	+	–	+	+	O	O	O	O	O	+	+	O	O	O	+	O
Teamwork	–	O	+	+	O	O	–	O	O	–	+	O	+	O	+	+
Self study	+	–	O	+	+	O	–	–	+	+	–	O	O	+	–	O
Examples	O	+	O	+	O	O	O	+	–	O	O	O	O	+	O	+
Drill & Practise	–	+	O	+	O	O	O	+	–	O	O	+	O	O	O	+
Analysis Exercise	O	–	+	+	O	O	O	–	+	O	O	–	O	O	O	O
Creative Excercise	+	–	O	+	O	O	O	O	O	O	O	–	+	O	O	O

Table 5.1: Students styles and course elements: + denotes a positive corelation of type course element, whereas – denotes a negative one. O means, that the style has little or no influence of the effectiveness of the course element.

profit from discussions. Content should therefore be accompanied by different types of examples and exercises.

Visual, *verbal* and *haptic* inclinations should also be taken into account. These three basic inclinations are frequently ignored. While visual or verbal input can easily be provided, the generation of haptic exercises is sometimes difficult, if not impossible. People with a strong haptic component also prefer to learn from examples, therefore it is important to provide examples wherever possible. The generation of some of these types may be especially difficult if the author has a strong preference for one of those styles. Where possible content should be presented in a visual and a verbal format. Examples should be given where ever possible.

While a student with *monarchic* form will profit from any type of material or interaction (provided he is interested in the subject matter), students with other form preferences need special attention. *Hierarchically* minded students learn best in structured, hierarchical environment like an online course or textbook. *Oligarchical* orientated students are better equipped to assimilating information from various sources, including lectures and discussions or scientific papers. *Anarchically* inclined students can basically assimilate information quickly, however they tend to get distracted. They profit from a tight lecture format that will present the basics in a compact way. To ensure maximum efficiency all essential information should be presented in a terse lecture and in an online course or textbook. Additional information can be provided in various ways.

Finally, students can prefer a *global* or *local* scope. Students with a global disposition are only interested in the grand scheme. Some of them even prefer to find things out for themselves and are miffed if information is presented in a very structured way. Students with a local scope preference on the other hand, love to dwell on details. Mastering the details, will make them confident to tackle the whole. To ensure that local and global thinkers are motivated, material should be hierarchically structured and of increasing detail. This is easily achieved by separating content into essential, important an additional information. The small number of students bent on a DIY (do it yourself) access to information, is easily pacified by providing all the original sources.

When taking all variants in style into account the "ideal" course should consist of

- A short, terse lecture laying out the basics
- Question and answer session
- Discussions, both in real life and virtual
- An hierarchical presentation of all basic information as online course or lecture notes
- Level indicators on all materials (essential – important – additional)
- Content in visual and verbal form where appropriate
- Additional resources to make up one's own mind
- Detailed examples
- Different types of exercises

Additionally it any topic should answer the question "Why?", "What?", "How?" and "What if?" (compare Kolb (Kolb, 1978)).

Chapter 6

Lecturers' needs

One mark of a great educator is the ability to lead students out to new places where even the educator has never been.

Thomas Groome

As Shackelford pointed out in 1990 (Shackelford, 1990):

> [...]the great majority of effort has been devoted to bypassing the teacher. We estimate that in excess in 99% of educational software development has focused on products for use by students.

Most education related software is indeed geared towards students and to replace the teacher. Even the software for the creation of new educational content does not answer the needs of most university lecturers. The software commonly used to create materials like word processors, presentations programs or web-design programs is not primarily designed for educational but for business purposes. Necessary features like support for large documents, formulas or literature citations only came in as an afterthought. Even software targeted for university use like LMS does not support the academic workflow in a sufficient way.

As usability is the backbone of the *e*-ULE system, a careful user and task analysis was carried out to establish lecturer needs. Most of the analysis stems from personal experiences as a ghostwriter/co-author of physics lecture notes and the software support for other lecturers. Another part was gained by watching colleagues at the department for software engineering.

To avoid the fickle ways of physicists and computer scientist to exert undue influence on the analysis, scientist and lecturers of diverse fields as psychology, agraric marketing, botany, climatology, geology and others were interviewed about their habits and practices concerning the production of lecture notes or other lecture supporting material. Participants were encouraged to share their views on the tools used and voice their requirements and wishes for the "ultimate" authoring tool(Naber, 2002a).

A prepared online survey was dropped after test run with colleagues (whose methods are well known) showed that they blatantly lied. A similar effect had already been seen in the student's need survey, so again the format was switched to individual interviews.

6.1 Materials already in use

This section discusses materials commonly used to support student learning. The task of providing supporting material for students was frequently characterised in terms of "necessary evil" or an unbeloved duty.

Frequently mentioned learning materials were:

- No materials ("Visit the lecture and write down what I say, or else learn to use a library.")
- Copies of presentation slides
- Reader (Collections of papers and book excerpts)
- Lecture notes
- Books
- Course web-site

The most frequently used materials were *slide copies* followed by *reader* and then lecture notes. Very few lecturers use a book of their own, but some make references to the work of colleagues. The percentage of lecturers giving out *no material* at all, varies from university to university and often from field to field. While Vienna University of Technology (TU) students as well as students at the University of Natural Resources and Applied Life Sciences (BOKU) are well provided with materials and even lecture notes, University of Vienna students frequently have to make do with readers of slide copies or rely on materials provided by colleagues (via the student union). The least materials are provided by the department for interpreting, which completely discourages any materials at all requires students to attend all lectures.

The course *web-sites* were mostly used to provide a download version of the lecture notes and to spread news (and other meta information) concerning the course. A few lecturers made use of discussion groups (mostly third party provided, e.g. MSN Communities). Discussion groups are one of the students' most wanted features, but moderating these groups proves to be very time consuming, thus they do not rank highly with lecturers.

The *none* option certainly is not a favourite with the students. The *reader* versions are acceptable for students in higher terms, but spook the freshmen. Everybody finds it difficult to get up to speed when required to read through dozens of papers or book excerpts trying to assimilate just the basic facts. Full fledged *lecture notes*, or even *books* are highly favoured by the students, but still, some information needs can better be catered to by online versions.

6.2 Tools

This section discusses tools involved (or not involved) the creation of learning materials as well as their usefulness to the process.

6.2.1 Tools in use

The production of *lecture notes* is nearly exclusively done using a *word-processor*, or the TEX/LATEX typesetting system (mainly in natural sciences, technology). One interviewed lecturer mentioned the use of a DTP system and one was using an *e*-learning authoring environment. The slides, again, were done in an *presentation program*, or using the TEX/LATEX typesetting system.

Additional materials (mostly illustrations) are created by using drawing (e.g. vector based drawing programs like Coreldraw) and graphing programs (spreadsheets) or by the use of domain specific software like statistic programms (e.g. SPSS, SAS), plotting programs (e.g. Sigmaplot, Origin), CAD Systems (e.g. AutoCAD).

Several were using various WYSIWYG *web-authoring tools* (mostly Macromedia Dreamweaver, Microsoft Frontpage, Adobe Golive) to produce lecture web-sites (course information and download of materials). Nobody tried to produce entire *e*-learning systems – complete with lecture notes, forums and course information – using a web-authoring tool. Some where exporting their word processor or LATEX file to HTML and put them on the WWW.

Some lectures used discussion forums provided either by the department or hosted by a commercial/free forum hoster. Although they valued the online discussion as a medium for gaining deeper insights, they found it very time consuming and several considered outsourcing the moderation task to senior students, or else stop using it.

One university provided tool (Sides4mi, Vienna University of Technology (unikat, 2002)) to disseminate information about a lecture on the web received very bad marks. Mostly it was considered too much effort for too little gain. While the student interface is straight forward and easy to use, the administration interlace is buggy, not cross browser compatible and requires a complicated authentification process. This system features a "push client" to keep students informed about their "subscribed" lectures and was rather popular among the students until the first hype died down, and the information therein became increasingly unreliable.

6.2.2 Tools not in use

The usage of *document management systems* and even *citation software* is not very widespread. Nobody used a document management software aside the file system and citation software is mostly used by TEX/LATEX users in the form of the BiBTEX bibliographic reference program.

This is surprising considering the amount of literature involved in creating lecture notes. As most of the information is still paper-based the absence of document management software is not too surprising, however with the rise of the WWW scientific publishers are switching to web-based delivery. While computer science, natural science and technical journals are completely digitised nowadays (even their archives are accessible via the WWW), social science journals have just started the process. Still, coming years will see a complete move towards digital academic publishing. But still "document management" stays paper based: freshly down-loaded paper is printed and filed into ring binders.

This is partly caused by the fact that even the completely digitised journals keep their print layout intact and thus are not suitable for reading on screen. Because they can not be read online, they are printed and because they are printed they cease to exists as digital material, forsaking possibilities like full-text search.

The absence of *citation software* is even more puzzling but can be explained by the imperfect interplay of journals, citation software and word processing unit. Moreover citation software suffers from the "yet another software package" effect.

LMS are another group absent from the survey. No Viennese University offers a large scale learning environment, but several a toying with the idea. As LMS are usually either expensive or maintenance intensive (or both), they not used by single lecturers or even departments.

6.2.3 Tool critique – "Word processors considered harmful"

Several years of co-writing lecture notes clearly showed that most lecturers are overwhelmed by the sheer number of authoring tool features to choose from without being provided with a link to the task they are trying to accomplish. When watching the same lecturers using domain specific software (like a scientific graphing program or a measurement software), one is often amazed at the efficiency those tools are used with, however idiosyncratic and intricate they may be. Obviously a software that has a strong relation to the task at hand, will be easier to use.

The interviews showed that nearly everybody (even experts) had come to grief while using their favourite *word processor* package. No matter what program (Microsoft Word being the most common), these programs were not designed to write lecture notes, lest encyclopaedias. Word processors are designed for writing one page business letters, the features necessary for lengthy documents like table of contents generation, outlining, indeces ..., were added as an afterthought to increase the clientele, but never tested thoroughly. Using them sometimes is akin to charting "terra incognita". Creative page numbering ranks among the more harmless annoyances data corruption or loss is more serious. Using word processors for large documents requires detailed knowledge about the inner workings of the software – the "walk-on" approach will not do. As word processors are a necessary evil they still managed to receive middle to high marks for their overall usefulness. Some users mentioned that were not content with the software at hand, but could not think of a better software because they had become too accustomed to the existing concepts ("brainwashed").

Not surprisingly the users of TEX/LATEX systems are highly content with their tool as it was especially designed with the academic environment in mind.

Presentation software generally received higher marks than the word processors, simply because they are easier to learn and proved more stable. However, later generations sporting advanced "intelligent" features like auto-replacement and correction can drive their users nearly round the bend by correcting "errors" where there are not any.

The opinions on *web-development* packages differed vastly. Depending on the program used and the background knowledge available the tools where rated from very good to very bad. Easy to use tools, that require little or no background at all were rated highly by their users until complaints about browser incompatibilities started to arrive.

6.3 Most wanted features

In the interviews the lecturers were asked which features they would like to see in an authoring tool. The spontaneous answers receive a higher priority than the answers to a set of proposed features. This section only deals with the spontaneous answers. Of course oral testimony of things "users might want to do" in the future are to be taken with a grain of salt (Nielsen, 2001).

6.3.1 Basic needs

The ideal authoring tool should be

- easy to use[1]
- stable
- able to provide collaborative editing

That is the bottom line that all participants agreed upon. It is notable that everybody found it necessary to point out that the software ought to usable and stable, two characteristics any software should offer. Obviously all participants had had too many bad experiences.

The frequent cry for collaborative editing came as a surprise, as existing features of word processors are hardly used [2]. But as holding lectures (or generating materials for one) is not esteemed at our universities it is likely that lecturers seek to save time by collaborating on lecture notes. As the same lecture is not likely to be offered twice at the same department, it is important that the collaborative features work well even without the possibility of direct communication.

[1] devoid of function with no relevance to the task at hand

[2] On the other hand the combination of lengthy documents and collaborative editing is an invitation too trouble.

6.3.2 Media inclusion

The inclusion of other material (graphics, literature, multimedia, etc.) must be easy and reliable. "Cut and paste" inclusion of various graphical material is often still accomplished with scissors and glue rather than by keyboard and mouse commands. Inclusion of various media always raises the question of data formats. This question obviously overwhelms a significant number of users, so they rather forsake the use of media. This problem is easily overcome by providing a library for format conversion on the fly.

Mendes et al. investigated the use of hypertext authoring tools of 13 lecturers (Mendes and Hall, 1999). Among those 13, only 9 made use of images and less than 6 used other media. As the authors point out this reflects the typical contents to be found in an academic environment: text, text and more text. We do not concede to this view, as it seams unlikely that only one of the projects had need of bibliographic information and only 5 needed links to other resources. We gather that the lack of media is mostly due to difficulties in creating and integrating these very media.

6.3.3 Meta-information

Courses do not only require study materials: courses tend to come with an administrative overhead like student enrolment, times and places of the lectures/labs, ... Some universities offer support for the distribution of this kind of information, but again, this is done via additional tools and not integrated in the workflow. The authoring tool should provide facilities to enter information for real life events.

6.3.4 Interaction and collaboration

The easy (non-technical) integration of features as discussion forums and online chats was much welcomed and many lecturers reacted positively to the idea of involving the students in the creation of the materials. Some lecturers try to augment their lecture web-sites by adding third-party hosted discussion groups. Although in principle easy to use these boards are not as tightly integrated as would be necessary for optimal benefit. It is difficult to integrate content generated there, and the user is naturally dependent on the selected provider .

6.3.5 Support for link generation

Students frequently lamentated the lack of hyperlinks in so-called hypermedia applications. Lecturers on the other hand, tend to make hyperlinks only when explicitly referring to a section or resource, thus leaving the power of hyperlinks in educational materials unharnessed. As most of the HTML materials generated stems from conversion of word processor files, the materials reflected the hierarchical structure of the original document rather than the network type structure normally associated with hypertext (compare figure 4.2). As the network characteristic of hypertext was the original motivator for switching to hypertext documents, the lectures were surprised to find that link-generation is a rather tedious task and software support for this task is slim. More linking support was a frequent demand of lectures who already had used online materials.

6.4 Understanding the Authoring Process

The academic workflow and especially the authoring process are not at the centre of scientific interest. While there exist some papers discussing the possibilities of using hypertext or electronic media in academic publishing (Kolb, 1997), little is to be found on the overall workflow. (Molitor-Lübbert, 1997) and (Knorr, 1997) discuss details of the literature research and information retrieval but the data is based around case studies focusing on tool use and centres around the technological aspects.

6.4.1 Style and process

A broader insight can be gained by applying Sternberg's *thinking styles* (see section 3.2) theory. Scientists tend to favour the *legislative* and to a minor part the *judicial* functions of mental self-government. As the legislative style is correlated with the *liberal* style, it is not surprising to find that many lecturers like to present materials in their own way instead of relying on ready made materials (mostly textbooks written by somebody else). As regards the forms of mental self-government, researchers tend to *oligarchic* to *anarchic* in their preferences. Aside a certain amount of lecturers favouring the *monarchic* style, very few show a strong *hierarchic* tendency. It has already been proposed that the hierarchic style is not a natural style itself, but a learned one. Several years spent in executive-hierarchical environment such as the school system, imprints that a hierarchic style is a desirable asset. More than half of the participants in the survey expressed that they would rather be organised and structured in their ways. While a hierarchic is favourable for preparing lectures and lecture notes, it is less of an asset in the academic environment per se, because the information accumulated by research hardly tends to be structured.

Depending on the individual style therefore the creation of academic study material can be a rather chaotic process. Although the picture of the "scatterbrained professor" is a little bit too farfetched, it should be taken with a grain of salt. Complex systems – as lecture notes – involve complex, even chaotic processes – processes that are often out of the reach of common authoring tools like word processors. Many of the interviewed lecturers confessed to a somewhat chaotic creation process, which often involves writing many chapters concurrently, while trying to figure out the most intuitive order. All this is done while juggling a plethora of scientific papers which might "come in handy".

6.4.2 Building blocks

Any, even the most complex, workflow can be broken down to some simple building blocks:

1. Decide on a course
2. Brainstorm
3. Literature research
4. Write content
5. Define/find keywords
6. Add additional content
7. Organise/structure material
8. Provide course information
9. Organise students

The steps and the tools involved in the design of new courses are detailed in table 6.1. Sections 6.4.3 to 6.4.11 detail the individual steps. Steps 4 to 6 can overlap (adding media, however, is frequently the last step before publishing). Steps 2 to 7 can be repeated as necessary.

6.4.3 Step 1: Project start – decide on a course

The reasons to give a lecture can vary. Among the most frequently cited are

- "Someone has to do it"
- "This my research topic and I need some more students"
- "I like teaching and I think the students will benefit from this lecture"

Obviously the motivations vary widely. In the first case the junior members of the faculty/department are doomed to give all the boring introductory lectures. Giving this lecture usually is not their own decision, therefore no steps in decision taking are required.

In the second case the lecturer hopes to attract students to help him with research. As the students need to fulfil specific requirements, the content of the course is easily selected and the materials are already available. If all the necessary knowledge can be crammed into one course, the decision step is finished at this point. If the content/material severely exceeds the amount suitable for a single course, it will have to split over two or more courses.

In the third case the content of the lecture may be completely unknown at the start. Some university curricula offer extreme flexibility by introducing generic lectures like the AKIK ("Ausgewählter Kapitel der Informations- und Kommunikations Systeme) at the Vienna University of Technology. A scientist inclined to teach can choose not

No.	Stage	Tools
1	Decide on a course	unknown
2	Brainstorm	pen & paper
		brainstorm software
3	Literature research	printed journals: filed in binders
		e-journals: stored in digital document management software or printed and filed in binders
		WWW: bookmarked, saved
4	Write/reuse content	word-processor
		DTP software
5	Define keywords	indices are rarely used
6	Add additional content	specialised software (e.g. CAD program, Music editor, formula editor)
		Literature references: specialised software
		interaction: web-based discussion groups → server software
7	Organise/structure material	word-processor ?
8	Provide course information	Lecture web-site: web-design program
		ftp program
		University provided tool for lecture web-sites
9	Organise students	*e*-mail program
		University provided tool for student data manipulation

Table 6.1: Creating a course: stages and tools used

only from his own current research field but from other fields as well and in fact might further his own knowledge in the process. This approach to teaching is certainly the most rewarding, but requires the more preparation in turn. The decision process therfore involves a survey of upcoming scientific topics of interest to lecturer and student alike and a canvassing of related lecture activity at other departments. The most useful tool for these information search is the WWW. Nowadays any university has a web-based university calendar and most institutes sport web-sites to inform students of the contents of this years generic lectures. The 'up-and-coming' topics are easily gleaned from conference calendars and agendas. Many of those have the advantage of linking to last years conferences, complete with at least abstracts and even papers. The information gathered on these expeditions is for the larger part kept in the head, or anchored in the form of browser bookmarks. Only when a decision about the topic is reached, serious material gathering will take place. The information reviewed in the decision process is frequently ignored. See section 7.4.1 on *e*-ULE's support for step 1.

6.4.4 Step 2: Brainstorm

As we have already established the major part of lecturers favours an somewhat unstructured approach to the creation of lecture notes (and other writings), it is not surprising that the most useful modus operandi is brainstorming or free association. There are several very good brainstorming and mind-mapping programs available ((Mindjet GmbH, 2003), (TheBrain Technologies Corporation, 2003), (Inspiration Software, Inc., 2003)) alas they where again designed for business purposes and do neither support formulas or other specific requirements, nor are they geared towards the generation of lengthy documents. The information gathered and structured within has to be exported and recycled in a conventional word processor.

Another problem in the brainstorming process is the fact, that oligarchic or anarchic inclined people hardly stay on topic but switch forward and backward between two or more major topics (e.g. courses or research interests). See section 7.4.2 on *e*-ULE's support for step 2.

6.4.5 Step 3: Literature research

As pointed out by Nkambou et al. (Nkambou et al., 1998) curriculum development can be *content* (content to be provided is known), *course* (concept of a course already exists) or *material* (available material is sorted and connected to teaching goals) driven. Normally curriculum development is an iterative succession of the three types.

As Austrian university curriculums are not "air-tight" and allow for some leeway, one might expect that course design is not *course-driven.* In reality most new courses are in fact replacements for existing courses, thus a certain backbone is available. Still the larger part of course design is either content or material driven.

Undergraduate or introductory courses tend do be *content driven*, as the desired outcome is well known, but the contents itself are not the focus of recent research and thus materials are not immediately available. The preparation of a content driven course requires extensive literature research that will to a large part encompass textbooks.

Graduate courses on the other hand mostly centre on current research topics. Therefore enough material is at hand and these courses are mostly *material driven.* Little additional literature research is needed, but the existing material needs to be sifted, structured and prepared for non-experts.

Content driven and material driven design require two totally different ways of interaction with literature. Given the current state of technology even the means of interacting with the literature is different. Table 6.2 shows common academic knowledge media and the means of manipulation. Figure 6.1 depicts the media and information workflow.

Literature research can be subdivided into 4 tasks

- Finding literature

Medium	Find	Use	Process	Store	Search
textbook	library bookstore publisher	paper-based	–	–	index TOC
journal article (print)	library literature-provider	paper-based	scan	file system	–
journal article (web)	publisher literature-provider	electronically paper-based	–	file-system	system-search
web-site	WWW	electronically (www) electronically (local) paper-based	 save copy	www file-system	 system-search

Table 6.2: Knowledge media involved in the lecture creation process: how to find, use,store and search them.

- Obtaining bibliographic information
- Storing & retrieving information
- Referencing information

Subtasks may vary depending on the source and format of the information in question.

Finding literature

Until recent years the way to literature started at the library using their book and article databases. As working inside ones own office is preferable to working in the library, the spread of personal computers opened a market for personal databases such as Ovid (Ovid Technologies Inc, 2003). The rise of the WWW prompted many publishers to offer online access to their journals (and archives) and almost all technical and natural science journals are available online today with rising numbers in other scientific fields. Smaller publishers can out-source their online archive to literature/information providers such as (Ingenta INC, 2003).

As nearly all of the literature search is done on the web the potential for automation is high. Unfortunately the interfaces differ and no standard query language for literature search is in sight.

Obtaining bibliographic information

One a resource has been found, the extraction of the bibliographic information is the next logical step. Modern web-based literature database interfaces offer a possibility to

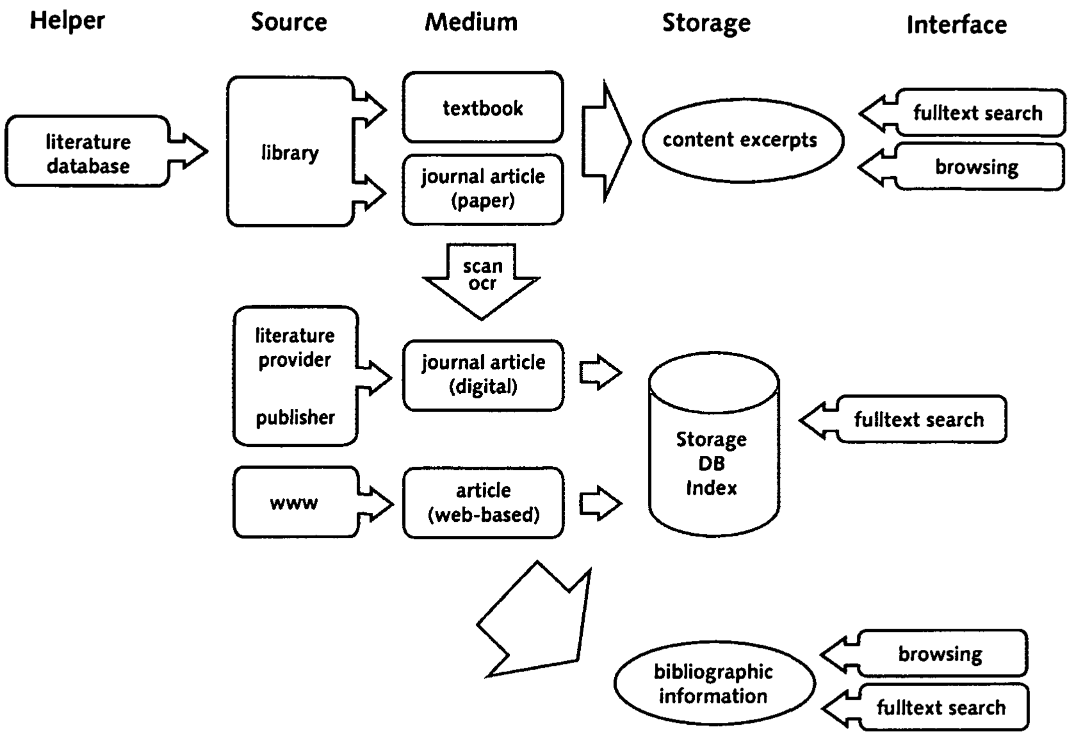


Figure 6.1: Media and information flow in academic publishing

directly download the bibliographic information in a format suitable for direct inclusion in a citation software. As we found in our interviews, the use of citation software outside the technical and natural sciences is not very widespread. One of the causes is the fact that citation and literature management requires another separate program, which in turn requires learning and can produce function as yet another trouble spot (especially regarding the integration with other programs).

Storing & retrieving information

Digital articles are usually stored in the file system. Todays hierarchic file systems derived from the conventional paper and ring binder world, are not an ideal place for scientific information, which naturally is not so hierarchically that one might not want to access the information via various paths.

Textbooks and paper based articles can not be efficiently used by a computer program. While an article might be digitised by scanning and OCR (Optical Character recognition), it would be too time consuming to digitise a whole textbook. However the abstract and annotations can be stored digitally. This allows for a limited search on the literature base.

The documents retrieval is enhanced when using a document management system that allows keyword association and even full-text search across the data stock. Again document management systems are absent from the academic world. Again the argument being "too much trouble for too little gain".

Referencing information

Provided the bibliographic information was separated in the second step the creation of a bibliography is not very difficult. The main problem is the word processor support, as todays word-processors still lack this functionality. Therfore many bibliographies are written manually.

See section 7.4.3 on *e*-ULE's support for step 3.

6.4.6 Step 4: Write content

Depending on the individual style preferences the content creation is either straight forward or very complex. About two thirds of the lecturers confessed that they had difficulties in remaining on track or in building a structure in the first step. Many lecturers start with a structure, than randomly add content, restructure, add more content, restructure again, and so on. The generation of content without structure leads to the creation of parallelism, as sections or parts thereof are written more than once.

This situation becomes even more confusing as more files enter the scene. Hypertext systems are made up of many small files, whereas lecture notes usually only consist of one big file. If the contents of the files change frequently, hyperlinks between the files become cumbersome and will be replaced by table of contents thus resulting in a hierarchical structure instead of a networked or semi-structured one (compare section 4.2.2).

An authoring tool therefore should provide features for structuring and restructuring information, a repository for text that has no place in the hierarchy and features for bidirectional linking that keep intact when the context changes.

Academic writing can contain a number of features not normally found in texts. These include formulas, chemical symbols, foreign languages and special alphabets. While a formula editor comes with every common word processor, chemical symbols require additional software as do special alphabets. See section 7.4.4 *e*-ULE's support for step 4.

6.4.7 Step 5: Define/find keywords

Classical lecture notes only require keywords for index generation. As index generation is more difficult and boring than the creation of bibliographies and not compulsory, lecture notes normally do without. This of courses changes when switching to hypertext. As manual hyperlink generation is out question (more difficult than index generation), semi automatic hyperlink generation is called for. Semi-automatic hyperlink generation relies on keywords that are defined by the lecturer. Every occurance of keyword will create hyperlink. See section 7.4.5 on *e*-ULE's support for step 5.

6.4.8 Step 6: Add additional content

Additional content comprises figures, sound, video, animation, ... for short everything that does not come under the heading "text". Illustrations are often derived from domain specific software such as plotting or statistical software, CAD programs and simulations. While media inclusion in a word processor can in most cases be accomplished by the clipboard, inclusion of foreign formats in hypermedia systems requires in-depth knowledge of data formats so it is not surprising that so-called hypermedia applications lack media usage (Mendes and Hall, 1999).

The production of illustrations is generally unpopular and some lecturers shy away from it to the extent that they rather would write several pages of text describing a effect or process instead of including an illustration. The use of illustration is also influenced by the personal thinking styles of the lecturer. While verbal/aural oriented lecturer would not see the need to include a graphic and thus frustrates his more visually inclined student, a visual/spatial oriented lecturer would include the graph, but might forget a description thus annoying his verbal/aural inclined students. See section 7.4.6 on *e*-ULE's support for step 6.

6.4.9 Step 7: Organise/structure material

Academic materials tends to be complex and the hierarchical structure offered by paper-based media is not always the ideal solution. While it is true that material in question is basically structured linearly or hierarchically there is often more than one possible structure. I call this the multipath model (in reference to the multipath propagation in wireless communication). The ideal authoring therefore allows fore easy regrouping while leaving all links intact and offers support for alternate (multipath) structures. See section 7.4.7 on *e*-ULE's support for step 7.

6.4.10 Step 8: Provide course information

Every course comes with some administrative overhead. Nowadays many universities offer an electronic university calendar that can be used by the lecturers to provide additional information to the students. Again this a separate tool unconnected to the workflow. The functionality vary from tool to tool. More enhanced versions will include the possibility to book a room and a possibility to inform students about changes whereas the more basic variant will only provide slots of date, time, location, name of the lecture and a short abstract. See section 7.4.8 on *e*-ULE's support for step 8.

6.4.11 Step 9: Organise students

The more overcrowded studies limit the number of students. This of course requires a interface to the student database and a mechanism by which to select the students that will be able to attend the course. *e*-nhanced learning as envisioned in the *e*-ULE project will allow for a larger number (or even an unlimited number) of students, as many students will learn at home and only attend sessions if they have questions that they could not answer using the materials or the discussions boards, or if they are really interested in the subject.

Other administrative task involving students include assignment handling, exam sign-up, and assessment. All these tasks are usually covered by individual tools and may involve other persons (secretary, dean, ...). See section 7.4.9 on *e*-ULE's support for step 9.

Chapter 7

e-ULE concepts

If a professor can be replaced by a CD-ROM, he/she should be.
Jack Wilson

7.1 *e*-ULE a jack of one trade

e-ULE centres around the idea of *e*-nhanced learning. *e*-nhanced learning means that lecture time is freed up by moving the process of pure information transfer to electronic media that can be consumed at the students' leisure. The freed time can then be used for question & answer session, the explanation of the more difficult parts of the material, for exploring uses or alternatives, for connecting the information to other fields, in short for anything that benefits from direct student-lecturer communication.

e-ULE thus does not aim at replacing the lecture or the lecturer, but at harnessing the full power of the lecturer. No lecturer is so bad, that he or she can in fact be replaced by a CD-ROM. It just happens that some of them have little rhetoric skill or are unorganised in their approach to information structuring. This however does not invalidate the insights and experience they can share.

e-ULE is especially designed to facilitate the creation of learning materials for *e*-nhanced university lectures. *e*-ULE allows the student to rearrange and augment the content in any way he sees fit. This offers the possibility of the deep understanding necessary to succeed in science, but it is time consuming. It also requires a certain amount of independent thinking to efficiently work with *e*-ULE . It not is suitable for the creation of either kindergarten or primary school applications. It might be possible to create business skills applications, it is however not geared toward that direction. This in sharp contrast to the "jack of all trades" standard software solutions like office suites and hypermedia design tools which provide functionality for nearly everything, but fail to get users actually using these functions. *e*-ULE aims to be independent of any scientific field. By integrating as much support for expert data formats such as formulas, ... it should be usable for every type of course.

7.2 Core concepts

Aside from the support for student and lecturer needs *e*-ULE is built around the following core concepts.

7.2.1 Free software

"Free software" is a matter of liberty, not price. To understand the concept, you should think of "free" as in "free speech," not as in "free beer." (GNU project definition of free software). In the last years the term "free software" has nearly been supplanted by less aggressive term "opensource software".

e-ULE will come free of cost and the user is free to change and enhance it in every conceivable way. As it is impossible to foresee all the needs a certain scientific field may have *e*-ULE has a modular interface and can be extended via plug-ins. The *e*-ULE project will provide a repository to collect additions and additions will also influence further releases. More information on *e*-ULE internal structure and technical details are found in chapter 8.

7.2.2 Zero-administration

A *e*-ULE user should not need a system-administrator. *e*-ULE's aim is to provide a "zero-administration" system, that once setup will only require interaction in the case of hardware failure. Setup of a new system will not require any undue computer skills, but simply a computer dedicated as *e*-ULE server and the installation medium (CD-ROM) that will take care of the rest. As the serving capacity of a single computer is limited, *e*-ULE server can be connected to groups to enhance capacity. More information on *e*-ULE internal structure and technical details are found in section 8

Most of the actual work is delegated to server, thus editing documents only requires a small authoring tool that can be downloaded and installed when needed. All documents are stored on the server and only downloaded for editing. The *e*-ULE system does not require a dedicated student client, a normal web-browser is sufficient. This dramatically reduces student support requirements. To ensure an optimum of accessibility the web interface comes in a version for modern browsers and a version for older browsers and PDA (Personal Ditgital Assistant)s that is also suitable for braille conversion or screen readers.

7.2.3 Collaborative editing

Collaborative editing is a major concept in *e*-ULE . Compared to the generation of presentation slide to as a lecture backbone, the creation of an *e*-nhanced lecture is rather time consuming. As *e*-ULE is server-centred the collaboration with lecturers from other universities, cities or even continents does not pose a problem at all. The creation of an *e*-nhanced lecture can even become a joint effort of the lecturer and his students.

The owner of a project can nominate as many authors as he wishes and can also confine these authors to work on certain parts of the *e*-ULE document. Tasks can be assigned to individual authors or are "left up to grab" on a "first come - first write" basis.

7.2.4 Topic-centred approach

e-ULE centres around *topics.* Topics are information "atoms" which can not be subdivided meaningfully (you can think about them as subsections, or subchapters). A strict process ensures that all meta information necessary to automatically build structures is available. Every topic comes with:

- a title, an optional subtitle, keywords, an abstract
- prequel and sequel information
- level information
- optional extends / requires informations
- optional objects, examples, exercises
- optional communication modules
- optional bibliography and web-links

Figure 7.1 shows the internal components of a topic. Figure 7.3 shows the student view of a topic.

Topics are contained in blocks and are linked by prequel and sequel identifiers, thus resembling book-like structures. In addition topics come in three flavours: *essential, important* and *additional.* This *level information* allows the student a faster access to the core elements of a lecture. Relations between the levels of the topics are marked by *extends* and *requires* identifiers. Figure 7.2 depicts the interplay between related topics.

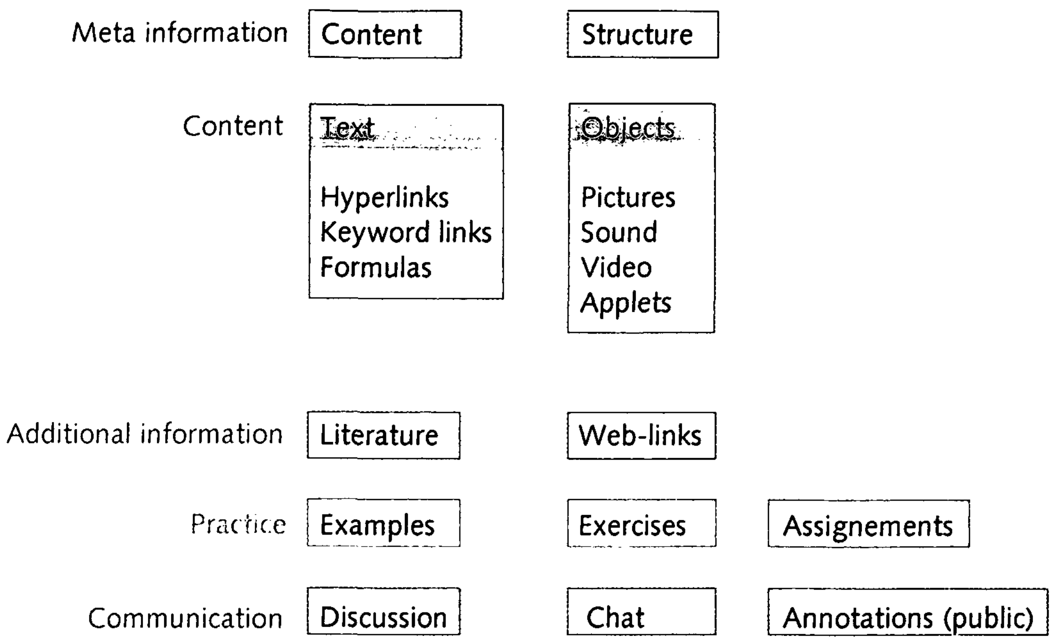


Figure 7.1: The diagram shows the the internal structure of a single topic.

7.2.5 Semi-automatic hypertext generation

Building useful hypertext/hypermedia systems is difficult indeed. It can easily be compared to the creation of book indices, and index creation certainly is very low on the list of popular pastimes.

Most hypertext systems only offer uni-directional linking (like HTML, the language of the WWW). Links therefore break easily if the material is restructured after the hyperlinks have been written. *e*-ULE offers *bi-directional linking* to ensure that hyperlinks stay in place even if the structure of the project is changed beyond all recognition. When creating an internal hyperlink the lecturer is presented with a list of all available topics. hyperlinks can also be created for non-existing topics. These topics will then be automatically generated.

e-ULE also supports semi-automatic hyperlink generation via keywords. The Link-builder takes on the tedious task of generating the necessary hyperlinks. After all, a HTML document without hyperlinks is no improvement as compared to a conventional printout. The link-builder relies on the keywords provided by the lecturer and builds hyperlinks to the main topic connected to the keyword, the glossary entry and the other occurrences of the word.

As pointing out key-words as he writes is not foremost of the mind of the lecturer,

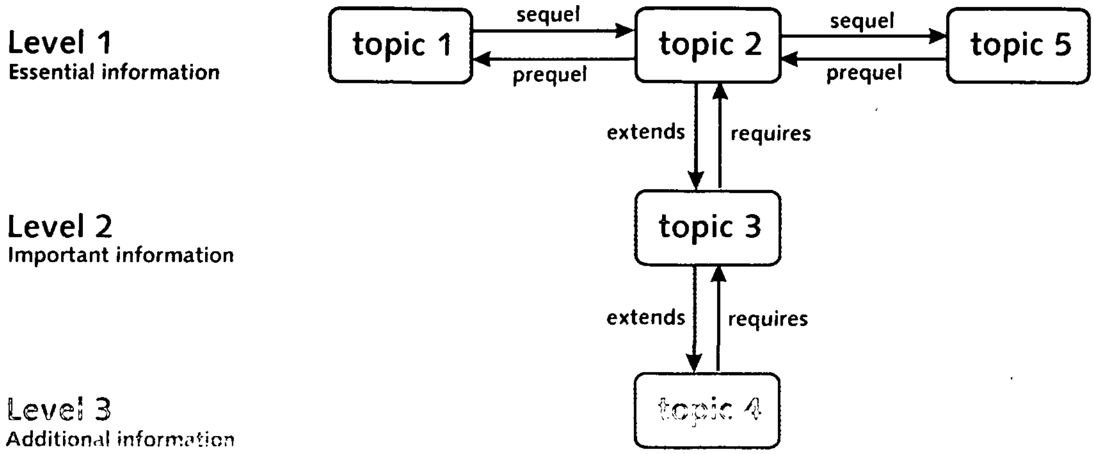


Figure 7.2: The diagram shows the interplay between related topics.

the *e*-ULE will help in identifying keywords. As we are dealing with scientific topics any newly introduced concepts have a high likelihood of absence from dictionaries. Using a modified spell checker that marks the unknown words the lecturer can therefore correct the spelling and define keywords as he goes along. This feature goes by the name "Common-sense-checker". The actions on the unknown word include: correct spelling mistake (choose from a list of recommendations), mark as new keyword, mark as a deviation from a known keyword (choose from list of similar keywords). New keywords come in two flavours: prerequisites, which are used to generate a special list of "should know" items and real keywords. The real keywords are immediately associated with a new topic and a glossary entry. Every topic can be manually associated with several keywords. These entries can be selected from each occurrence inside the text. See figure 7.4 for an example.

7.2.6 Integrated communication

Online communication is not an afterthought in the *e*-ULE system. *e*-ULE provides facilities for synchronous (chat and whiteboard) and asynchronous (public annotations, discussion groups). Communications modules can be assigned on a topic basis (every single topic can be have a discussion board and chat of its own) or one module can serve various topics.

Asynchronous communication is completely integrated in the systems. Searches will include discussions and annotations and reference results from these quarters separately.

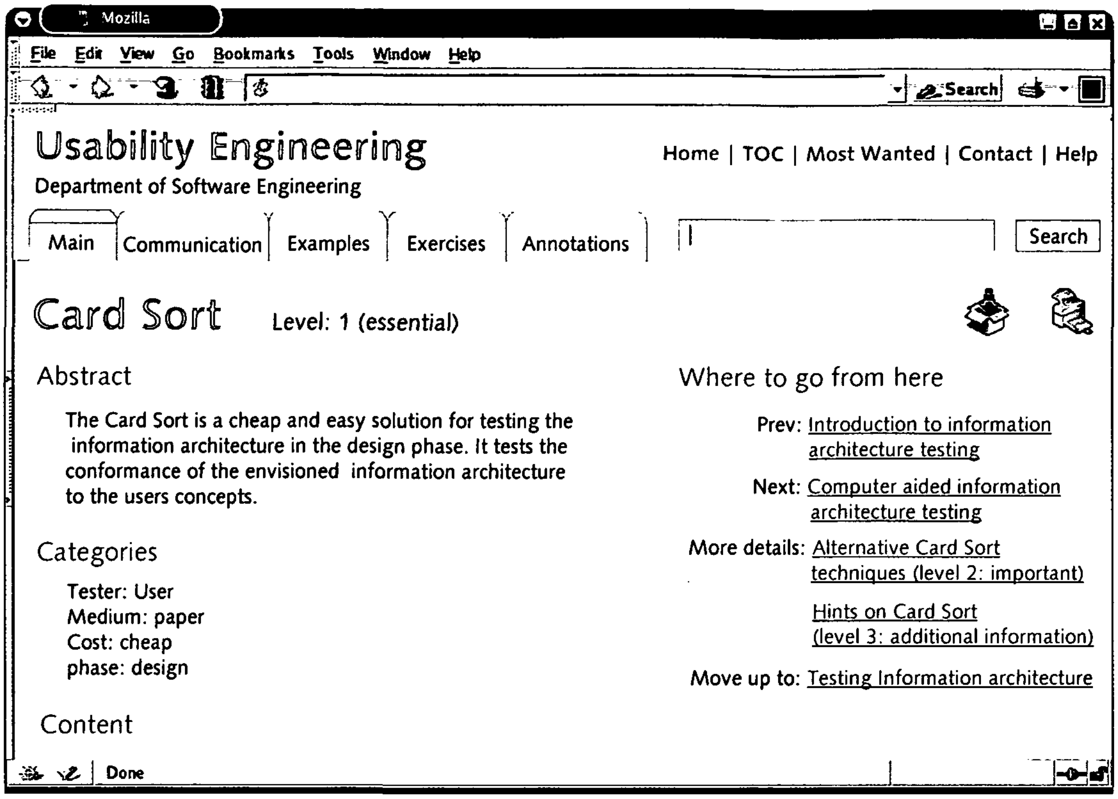


Figure 7.3: This screen shots show the student view of topic. The "main" page contains the abstract, meta information and navigation as well as the bulk of the content (hidden beneath the "fold"). Communication, annotation, examples and exercises get their page, each.

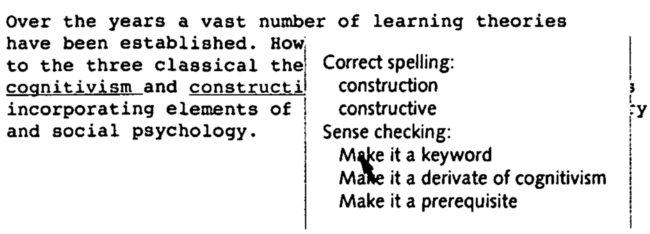


Figure 7.4: Common sense checker in action: cognitivism and constructivism are marked as questionable. Turning a word into a keyword will result in its own topic and glossary entry.

7.2.7 Visual formating

Being web-based, WYSIWYG is not a key concept in *e*-ULE . The author can choose among various *styles*, which can be adapted by a few simple choices (like university or department logo, department name, colour schemes, ...). On screen display varies naturally with the platform and the choice of browser, printable pdf-output is rendered via LaTeX. As the system is opensourced and transformation is achieved by XSLT, it is possible to create *custom styles*.

7.3 Support for student needs

In section 5.2 five distinguished learning/material usage phases have bee established. Combining this with individual learning and thinking styles one can see that properly hypermedia can do a lot for student learning. *e*-ULE aims at offering support for all five learning stages as well as various thinking style related preferences. Table 7.1 compares learning phases support for lectures, printed lecture notes and the *e*-ULE system.

7.3.1 Portal features

The server provides user-centred features. These can be accessed after registering with system. On start the system features a table of contents, a list of user generated views and a list of the pages last accessed. Pages that are frequently used (e.g. reference material like a periodic table or a list of physical constants) are detected and collected in a quick access menu. The portal also stores personal annotations, so they are accessible from any computer. See figure 7.5 for details.

7.3.2 Views

The server acts as portal to the student: content, information, discussions and annotations can be accessed in the usual fashion. Furthermore the server can generate any number of customised views for the student.

These customised views include:

- print version of selected parts for reading while commuting ...
- packaged versions for download and offline reading
- restructured notes for exam preparation (see section 7.3.5)
- goal oriented mini notes that highlight a special subject (see section 7.3.7 - phase 5)

phase	tasks	lecture	print notes	*e*-ULE
stage 1	getting an overview	preliminary discussion	TOC & introduction	mini notes automatically generated from topic summaries, list of prerequisites
stage 2	learning the basics	use exercises to ensure basic concepts to be understood before proceeding to more difficult ones	tag basic content	tag basic content
stage 3	learning	—	provide structured information	custom views
	deepen understanding	answer questions	references	online discussions, FAQ, integration of new material, annotations, involvement in creation process
	repeating	—	compile personal versions of the material on paper	use digital system to annotate and modify material, create condensed versions, discuss problems with other students
	self exam	—	problems & solutions	problems & solutions
stage 4	reference	—	TOC, index	TOC, index, updates, full text search
stage 5	just-in-time information	—	possible, but often cumbersome	automatic retrieval of additional information necessary for understanding the selected topic

Table 7.1: Learning stages and supporting mechanism: lectures, classical lecture notes and *e*-ULE lecture notes

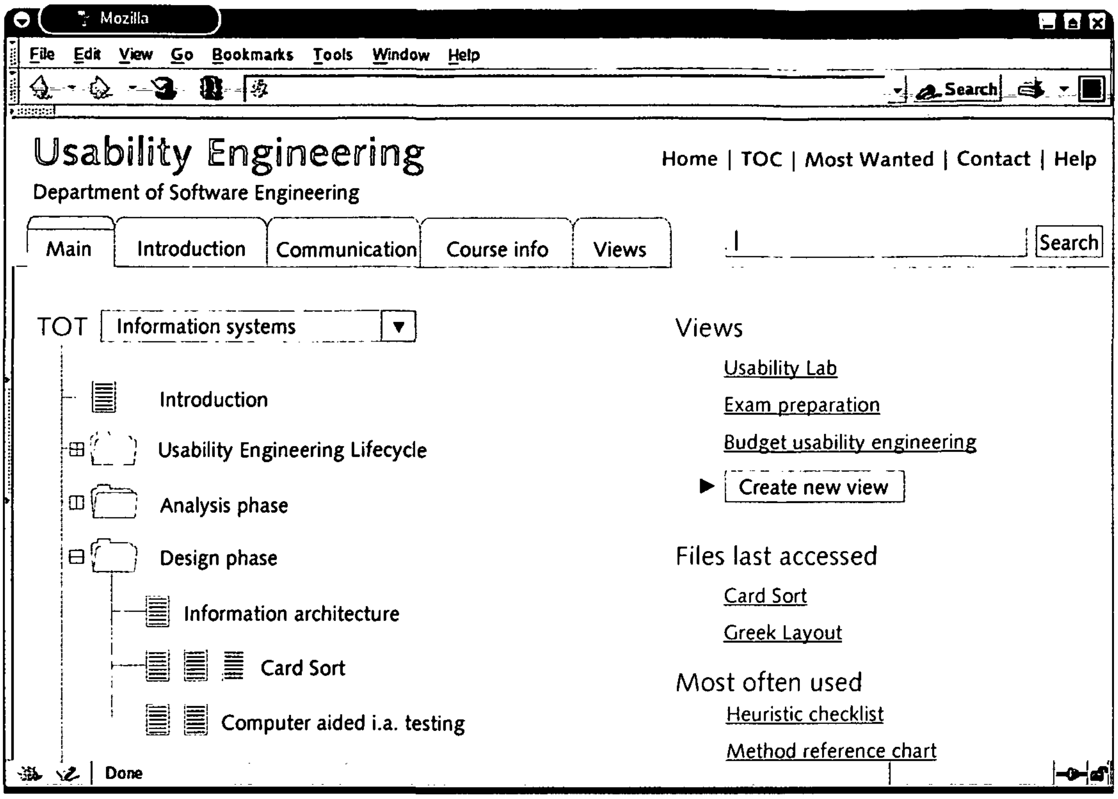


Figure 7.5: Browser view of *e*-ULE project. the "main" page is the central starting point giving access to various TOT, views and the most important files. The "introduction" page contains an overview of the course contents that is automatically compiled from the individual topics abstracts. The "communication" page lists communication resources pertaining to the whole course (not to individual topics) and the "course info" page deals with times and places.

The views (see figure 7.3.2) can be composed of selective parts of the content, information, annotation and discussion sections to the liking of the student. Glossary entries, and corresponding topics are automatically selected by the Link-builder. The student can control the creation of these auxiliary informations by de-/selecting the associated keywords individually (see figure 7.3.2. Views can be also be derived from existing TOT (see figure 7.3.2).

7.3.3 Support for phase 1 – reconnoitring

Aside from the necessary meta course information like times& places *e*-ULE will also provide a list of necessary prerequisite knowledge and "executive summary" of the

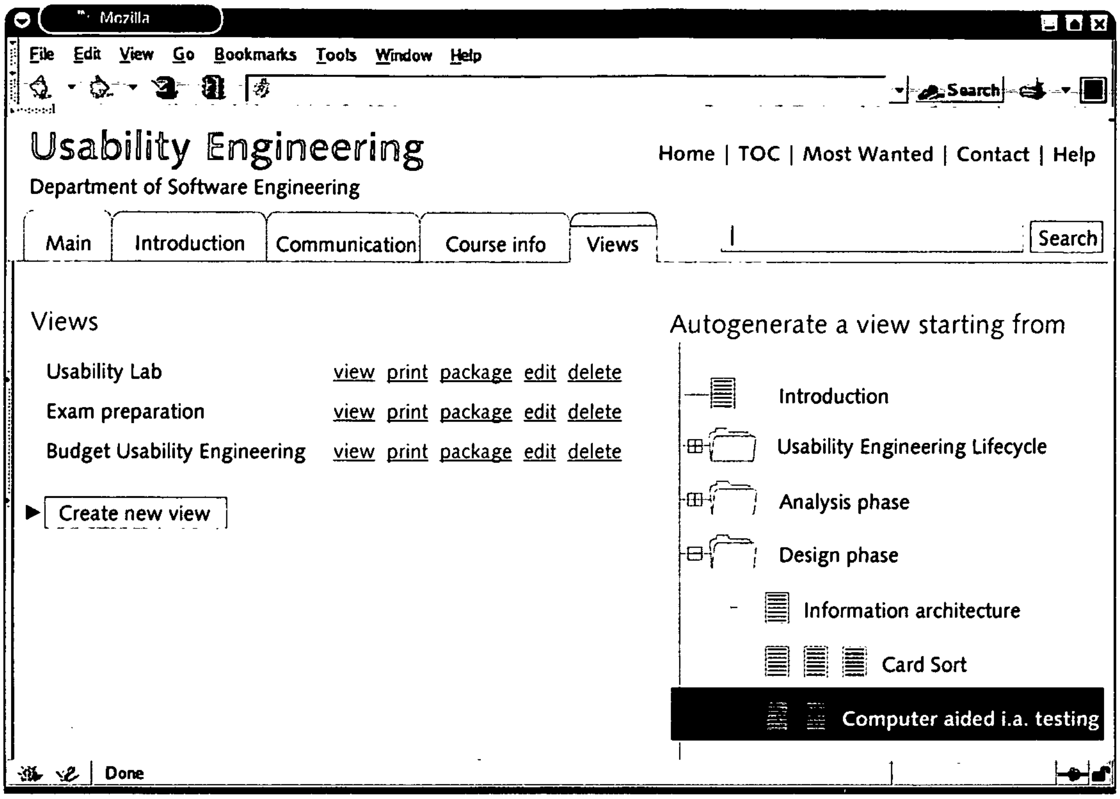


Figure 7.6: View section of a *e*-ULE project: new views can be automatically created to explain a special topic.

materials. All this is generated automatically by the system without the lecturers interference. If the students chooses to "enrol" in the course they can also receive updates and notifications via *e*-mail. See section 5.2.1 for a discussion of this stage.

7.3.4 Support for phase 2 – learning the basics

As *e*-ULE proposes a topic centred model (see section 7.2.4) learning the basics is a straight forward process as a path that only selects essentials topics can be automatically generated. Topics can be equipped with examples (showing practical application of the material covered in the topic) exercisess (to be solved by the student, an optional solution can be provided). These make ideal starting points into the field.

As communication features are topic based as well, as student can discuss the topic at hand. Communication modules include synchronous communication via chat and whiteboard (see figure 7.3.4, which will be most useful if many students will access the topic at the same time (most likely when really attending the lecture or a lab or

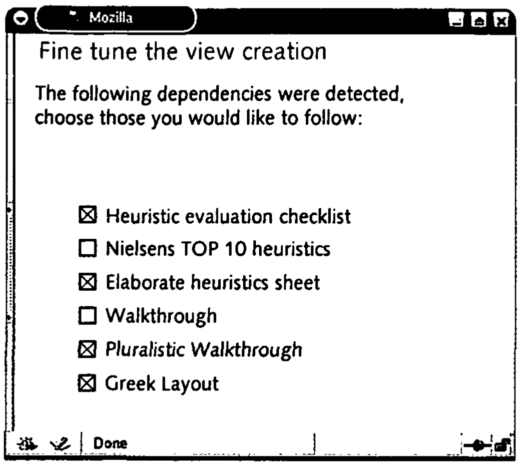


Figure 7.7: Choices at the automatic creation of a new view: dependencies between the topic are automatically resolved by the link-builder. The student can choose which of the detected prerequisites he wants to incorporate in the new view.

while solving (group) assignments) and asynchronous communication via annotations and discussion boards. Annotations can be both public and private. Students can also start peer-to-peer communication using any of the available IMS.

Since not every student is willing to learn while sitting at the computer *e*-ULE comes with extensive packaging option that also doubles as printing wizard. See figure 7.3.4 for details.

- Whole script or single topics
- By level (essential - important – additional information)
- Include annotations
- Include exercises
- Include examples

See section 5.2.2 for a discussion of this stage.

7.3.5 Support for phase 3 – exam preparation

Exam preparation requires detailed study of all the material. To keep track of the progress the student can mark the topics according to learning status as *read*, *completed* (solved all exercises), *learned*, *reviewed* and *done*. As an additional motivational factor, the material can be narrowed down to the topics that require further attention. See section 5.2.3 for a discussion of this stage.

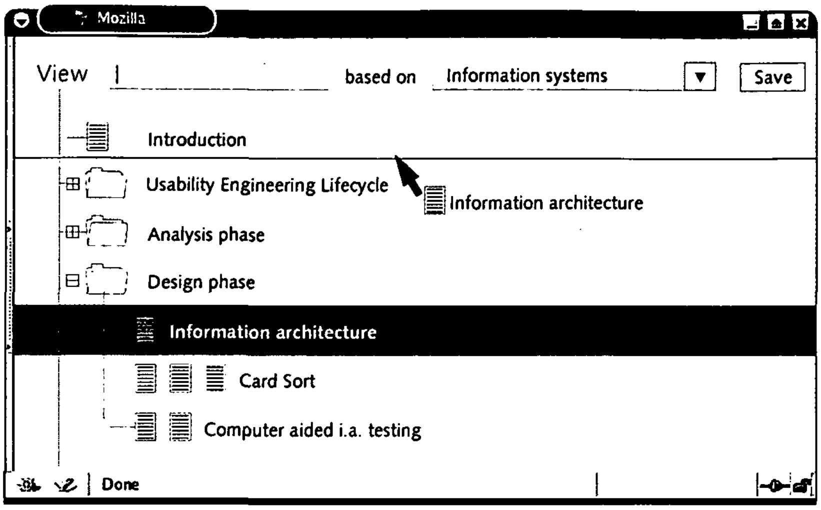


Figure 7.8: New views can be generated from existing ones by using drag & drop and delete.

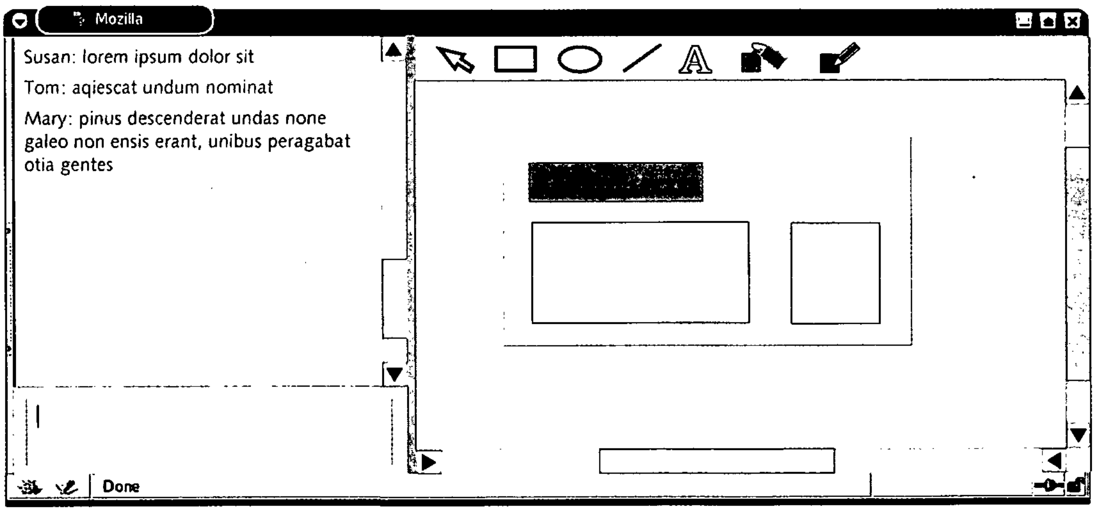


Figure 7.9: *e*-ULE features a platform independent chat and whiteboard. Chat/whiteboard session can be attached to individual topics and the students can setup their own public or private rooms.

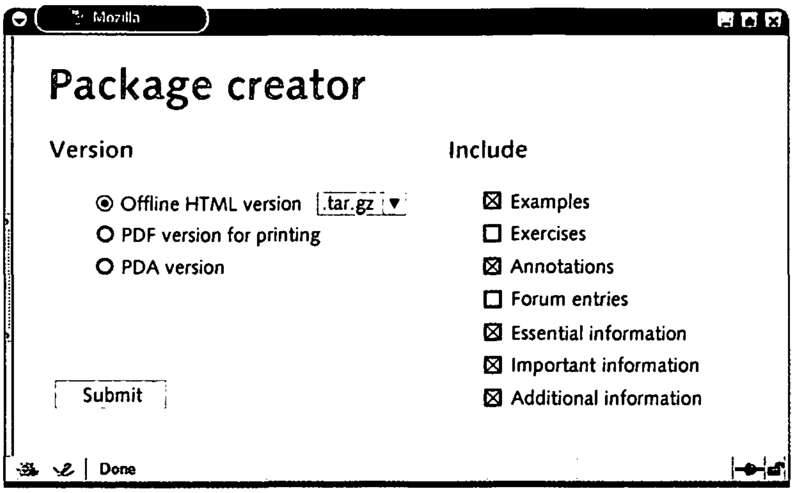


Figure 7.10: Single *e*-ULE topics or whole views can be packaged for offline reading or printing.

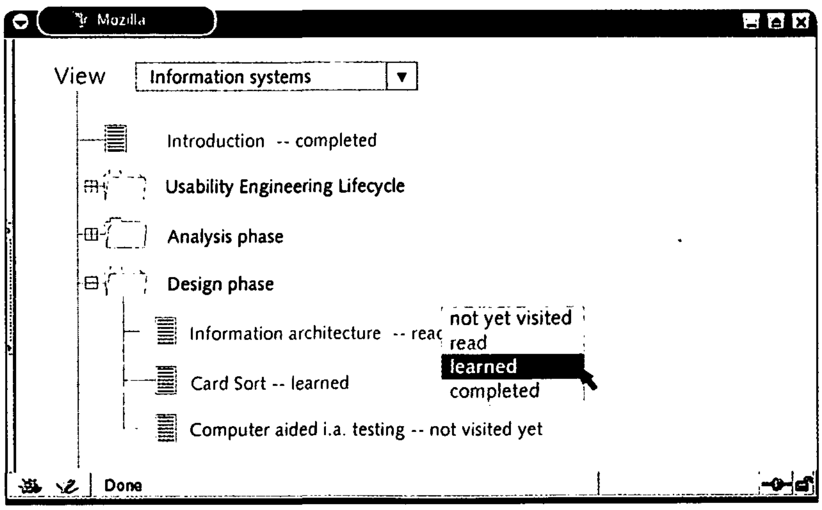


Figure 7.11: This student custom view shows the students progress on the contained material.

7.3.6 Support for phase 4 – reference

In phases two and three the student becomes very familiar with the layout and language/syntax of material in use. Unluckily enough normal lecture notes do not make good reference materials as they lack an index. *e*-ULE features a full-text search that encompasses all permanent information including annotations (private and public) and discussions. Due to the structured data-format the more relevant topics are easily identified. See section 5.2.5 for a discussion of this stage.

7.3.7 Support for phase 5 – just-in-time information

When needing only a small section of the overall material, the student can easily become frustrated because the of the interconnection to other (preceeding) parts of the material. *e*-ULE's Link-builder can solve this conflict: the embedded keywords lead to their associated topics and their embedded keywords again lead to further required topics. When generating a new view ("sub lecture note") the programs lists all dependencies. These can individually selected or deselected according to the persons previous knowledge (see figure 7.3.2). The result is slimed down, personalised lecture note. See section 5.2.5 for a discussion of this stage.

7.4 Support for lecturer needs – *e*-ULE Authoring Tool

The authoring system is a crucial point in the development of hypermedia learning environments. Any learning environment aiming at usable, interactive materials for students' use must provide an equally usable tool for authoring these materials. As pointed out by Shackleford (Shackelford, 1990)

> Typically, such software does little or nothing to support teachers[...]. The teacher still has to do all the various preparation, teaching, grading, and documentation tasks, and in addition must also devote time and energy to "dealing" with the educational software.

Looking at various existing authoring systems, one can find many different approaches to the subject: WEAR(Moundridou and Virvou, 1992) and AHA(Bra and Calvi, 1998) use HTML files which are augmented with meta information on the server side, while Interbook(Brusilovsky, 1998) relies on specially structured MS-Word files plus concept-based annotations. WebCT (WebCT, Inc, 2003) and many other of the last generation of commercial *e*-learning platforms provide a server based HTML forms/javascript interface or WYSIWYG interfaces.

Web-based services offer the advantage of requiring no installation. They are accessible from anywhere. On the downside an efficient web-based editor requires much client-side scripting and thus is frequently limited to one browser and operating-system,

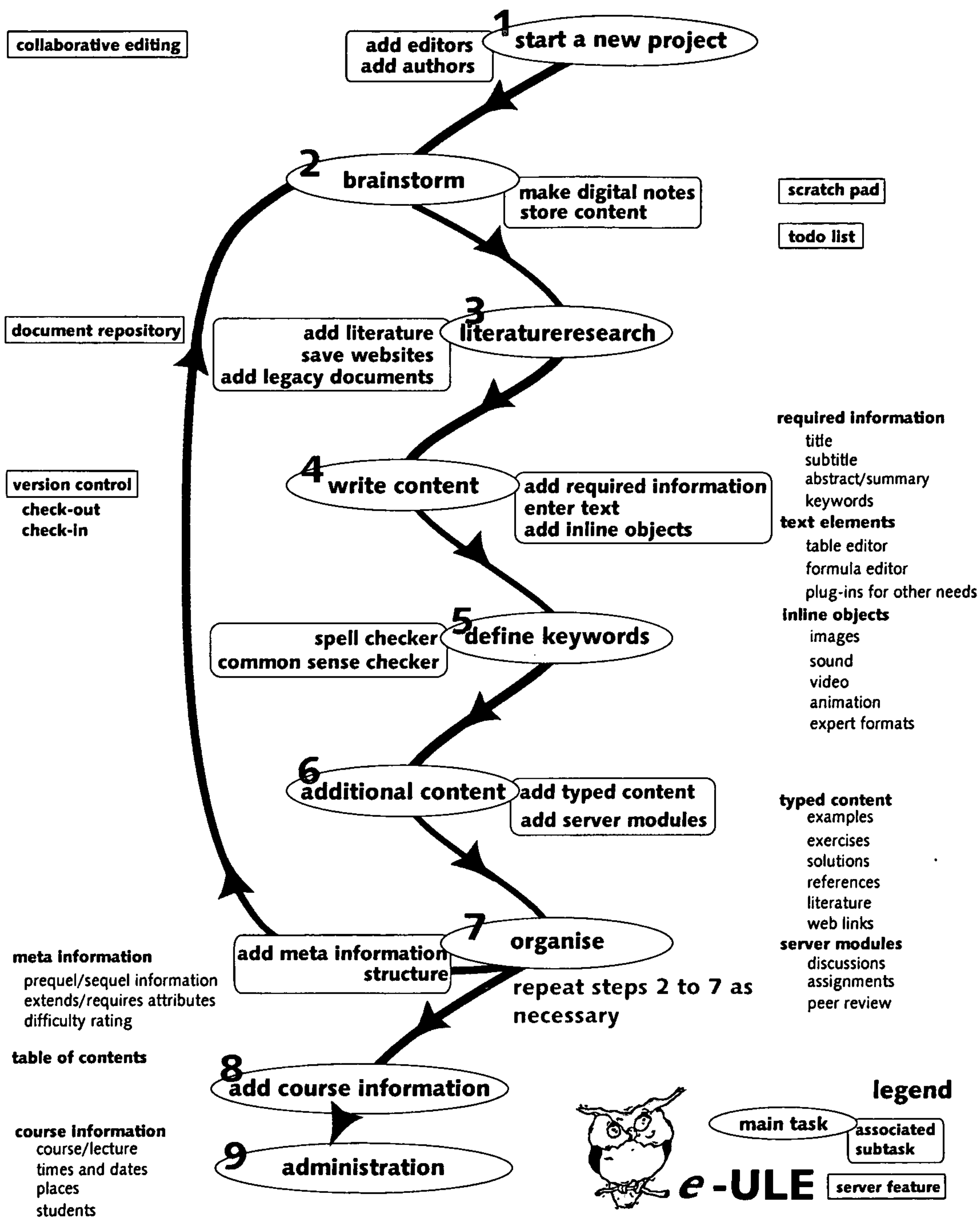


Figure 7.12: e-ULE 's support for the different stages in the authoring process. The steps 2 to 7 mark the core process, which will be repeated. Not all steps need to be accomplished in each iteration.

thus forsaking the biggest advantage of the web-based solution, that being the multi-platform capability.

Client-side authoring tools are often bloated and require installation and maintenance. Thus we settled on a slim multi-platform authoring client. While editing is done on the client, the bulk of the features like file management, version control, collaborative editing, is delegated to the server.

The field of tools for client side development is slim, a Java client (Sun Mcrosystems, 2003) and the Mozilla toolkit (Mozilla group, 2003b), where evaluated. As the Mozilla browser provides much of the functionality needed to deal with the documents – such as a editor, a spell-checker and formula and vector graphics display – we decided on a client based on the Mozilla toolkit.

e-ULE supports the lecturer in all parts of the workflow (see section 6.4). Figure 7.12 details the steps in the course preparation workflow and the *e*-ULE tasks associated with them. The tight integration of information search, information storage and information retrieval is also useful for other academic publishing purposes aside from the creation of lecture notes. *e*-ULE can be used for research and writing papers. The common database and environment will speed up the development of notes and papers alike.

7.4.1 Step 1: Project start

e-ULE can be used in the startup phase of a new lecture, e.g. while still hunting for the right subject. Any course is associated to a *e*-ULE project. A project collects all information including notes, literature and the materials developed therein as well as course meta information like times and places. A new project comes with one automatically generated author / editor (the person who started the *e*-ULE project). The main editor can appoint any number of additional editors (full control over the project) and authors (rights can be assigned individually).

Projects can be built from scratch or can be derived from other *e*-ULE projects. This is especially useful when catering to distinct, but similar student groups (e.g. bioinformatics for biologists and computer scientists). The start of a projects allocates server space and creates several new databases for the collection of *ideas* and *documents*. Existing idea and document repositories can be attached to the new project.

e-ULE includes an integrated web-browser. Thus all preliminary explorations of the subject at hand can be carried out from inside *e*-ULE . Web-pages or other files can – completely or in parts – be committed to memory for further reference. Details are described in section 7.4.3.

e-ULE projects can be password protected or open to the public. Guest accounts ensure that copyrighted material is hidden from unauthorised view (e.g. papers via site license should only be accessible to enrolled students).

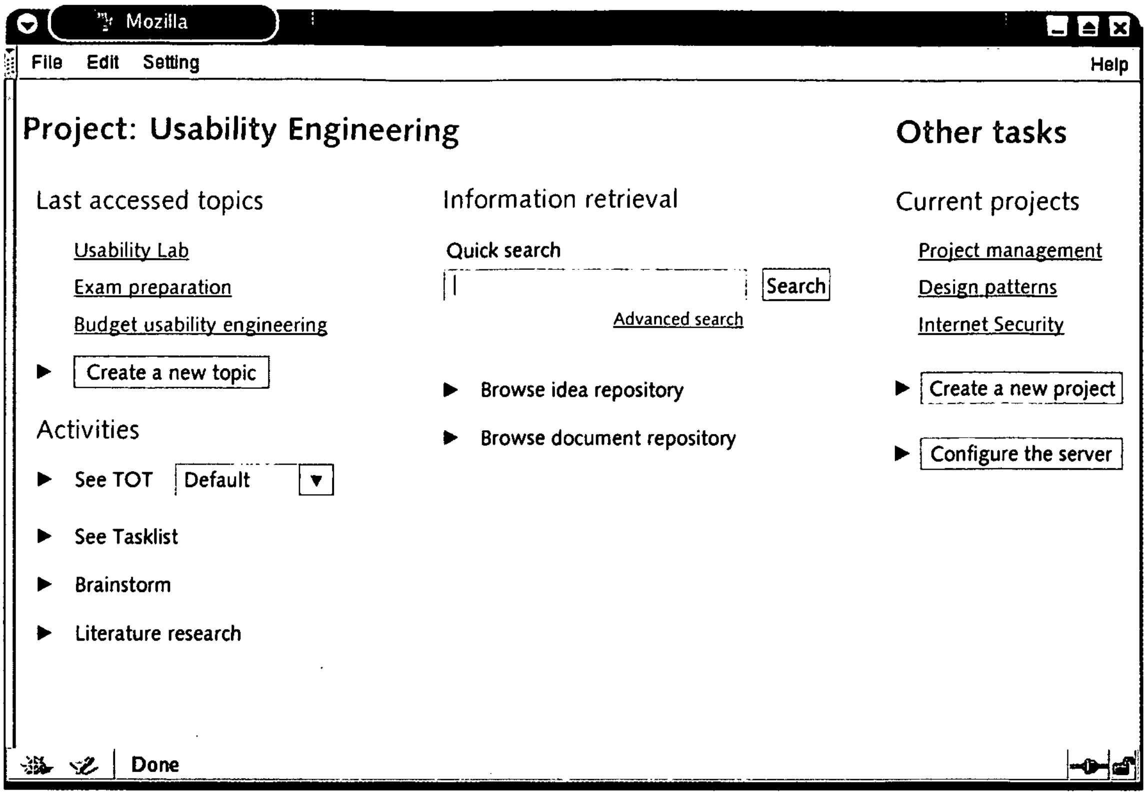


Figure 7.13: At the start of each authoring session, the lecturer portal organises the most relevant files and task pertaining to the project in an handy format. The user can decide if he wants continue working on the last project, switch to another project, or create a new one. the server administration console is also accessible from here.

7.4.2 Step 2: Brainstorm mode

Modern word processors frequently provide an outline mode to facilitate the generation of structured, treelike documents, thus forcing the author to think in chapters, sections, subsections, and so on. *e*-ULE goes one step further and allows for a "semi-chaotic" authoring style by providing a *brain storm mode* and an *idea repository.*

At any given time the author can jot down a new idea or start a brainstorm session (where new ideas are automatically entered into the system at the press of the enter key). The ideas will be kept in the *idea repository.* Ideas similar to the topic at hand, will be presented to the author for easy inclusion in the current topic. The contents of the idea repository can be browsed and search at leisure. Moreover user input is monitored and automatically compared to repository contents, resulting in links to existing information. See section 8.4.3 for details.

7.4.3 Step 3: Literature research

Writing lecture notes usually involves a lot of research on the topic in question. So, support for the literature research phase is a crucial part of the authoring tool.

Even though the web is now a major source of information, there are hardly any tools available to harness the power of the web. Many scientific journals can be read online, and many more are to follow, but handling of this wealth of information proofs rather tricky. Every publisher has his own system - incompatible with anybody else's system – and once the articles are safely downloaded, the real hassle begins. The articles cannot efficiently be organised, categorised and processed. In the end many users capitulate – articles are printed and filed in binders.

e-ULE 's research support will provide extensible support for literature search and management. Articles can be searched, downloaded, indexed, annotated, categorised and quoted without ever leaving the authoring tool. The same applies to any type of web-based information. *e*-ULE aims at providing a common interface to all scientific journals by the means of plug-in system.

Web-sites, and parts thereof can be book-marked and stored offline for further reference. Even if the site vanishes from the web, the information will still be accessible.

Information which was digitally received while working inside the *e*-ULE environment will be stored in the *document repository* on the server. The materials will be indexed an can be searched via an interface (see figure 7.14). As one of the main problems in literature research is remembering what already has been found, *e*-ULE monitors user input and offers related materials from the repositories (see section 8.2.2 for technical details). Additionally all necessary bibliographic information and other meta information such as the original path will be gathered and stored for further reference. Web-resources can be automatically monitored for changes.

Legacy documents (word processor file, pdf, postscript, ...) digitally available can be added to the *document repository* at any time. If the format can not be processed by the indexer, the lecturer can specify a description and optional keywords. Level information and annotations can be attached to any document inside the repository.

7.4.4 Step 4: Adding content: text, inline objects

Text can be added like in any modern word processor (see figure 7.15). *e*-ULE supports all the commonly used content types like lists, figures or tables. We intend to provide support for formulas and chemical symbols as well, as soon modern browsers are able to interpret MathML and CML correctly. At the moment support for formulas is limited to graphics or LaTeX (converted on the server).

Content includes also inline objects, most commonly graphics, sound, animations and video but also expert data formats like CAD files, destined to flow with the text. Data types common to the web are automatically detected and integrated, special types require either user interaction (specifying a link for the appropriate browser plug-in) or are handled as references on external objects.

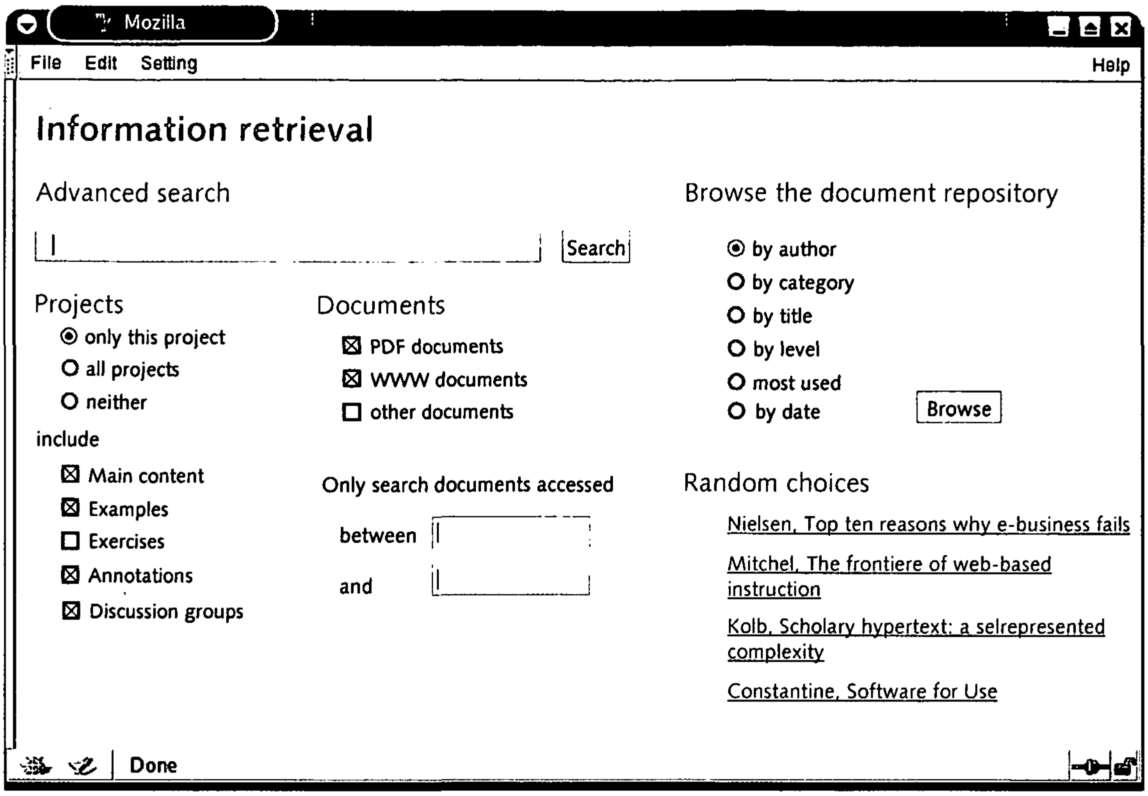


Figure 7.14: *e*-ULE offers a range of tools for information retrieval. Aside from a simple fulltext search, there are an advanced search and various browsing options.

Visual Markup is discouraged, in fact it is limited to italic font to emphasise certain parts. Allowing free markup would interfere with visual cues for keywords, ... provided by the system and furthermore distract the author from the contents. However a range of logical Markup like *quote*, *keyboard input*, *heading*, ... is available.

7.4.5 Step 5: Defining keywords – Common Sense Checker

The Common-sense-checker lays the foundation for the automated link-builder. In a process similar to spell-checking, the authoring tool will mark all unknown words. The actions on the unknown word include: correct spelling mistake (choose from a list of recommendations), mark as new keyword, mark as a deviation from a known keyword (choose from list of similar keywords).

New keywords come in two flavours: prerequisites, which are used to generate a special list of "should know" items and real keywords. The real keywords are immedi-

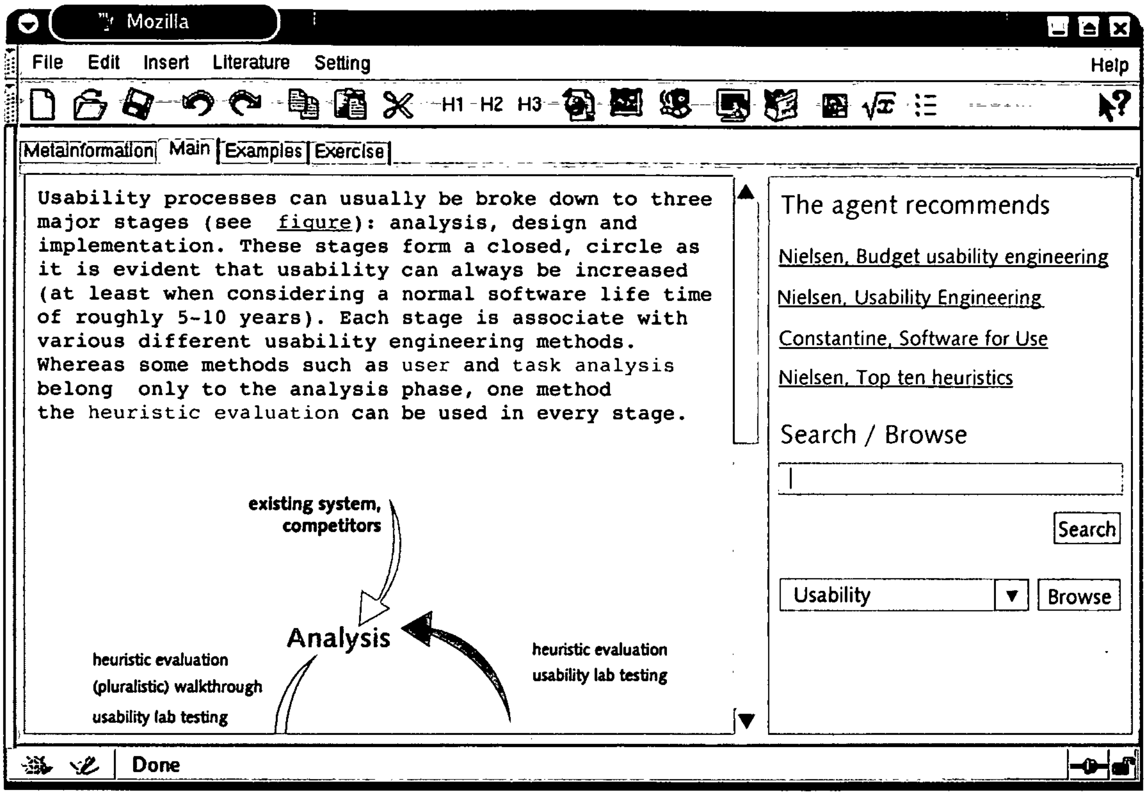


Figure 7.15: This screenshot shows a topic in the process of being edited. The editing process is divided into four major parts: meta data, main, examples and exercises. If the topic has been publicly annotated by students, a fifth tab " annotation" allows the editing of those annotations.

ately associated with a new topic and a glossary entry. Every topic can be manually associated with several keywords. These entries can be selected from each occurrence inside the text. See figure 7.4 for an example.

The common sense checker, of course, relies on user input to point out useful keywords. This cannot be avoided, as the very same word could be a keyword or just ordinary word, depending on the audience. An *integral* in a freshmen's course on mathematics of economists has a meaning different from an *integral* in postgraduate course of particle physics. In addition to the automatically detected words, the user can convert a word into a keyword (and back again) at any time.

7.4.6 Step 6: Additional content

e-ULE features typed links (compare (Wu et al., 1999), teaching operations (Brusilovsky, 2000)) to include additional contents like examples, exercises and solutions, assignments, bibliographic information, hyperlinks and external material.

Server modules add interactivity to the lecture notes. Currently available modules are *discussion groups*, *chats (with whiteboards)*, *annotations* and *online assignments* with automated hand-in and peer review option.

Discussion groups & chats

Communication is a vital aspect of learning. Since the aim of the *e*-ULE system is to deepen student - lecturer communication by reducing the pure information transfer aspect of the lectures, there is room for communication in the lecture hall. Still, there will be the need to communicate electronically: in the forums students can talk to each other, to the lecturer or to tutors. Different communication needs are served by different types of communication channels: asynchronous, newsgroup like systems (forums) and synchronous chat systems.

Server provided discussion groups (figure 7.16) and online chats (figure 7.3.4) can easily be integrated and may belong to one single topic or to a group of topics. The contents of the asynchronous discussion groups are completely integrated with the system and feature in full text-searches of the system. The project owner can conveniently refer to discussion entries and integrate them into the main body of the topic.

Chat sessions can added to a page in general and students can setup their own chat sessions by inviting other students. See figure 7.3.4 for an example of a chat and whiteboard session.

Annotations

Each topic can also be annotated by the students. These annotations can be either public or private. Annotations are another means of communications. In a notice board manner they can serve as platform for collecting additional information about each topic, like errata, implementation examples, notes about practical applications or additional web-links. Again annotations are completely integrated with the system an feature in full-text searches. See figure 7.17 for details.

Assignments

It is also possible to add assignments and have the results submitted electronically. Submissions could then be distributed for peer review: students would review the work of another student. The resulting reviewed assignments can be attached to the *e*-ULE projects.

7.4.7 Step 7: Organisation & structure

As pointed out in sections 5.3 and 6.4.9 the lecture notes creation process can be somewhat chaotic. While lecture content tends to be sequential (compare section 4.2.2),

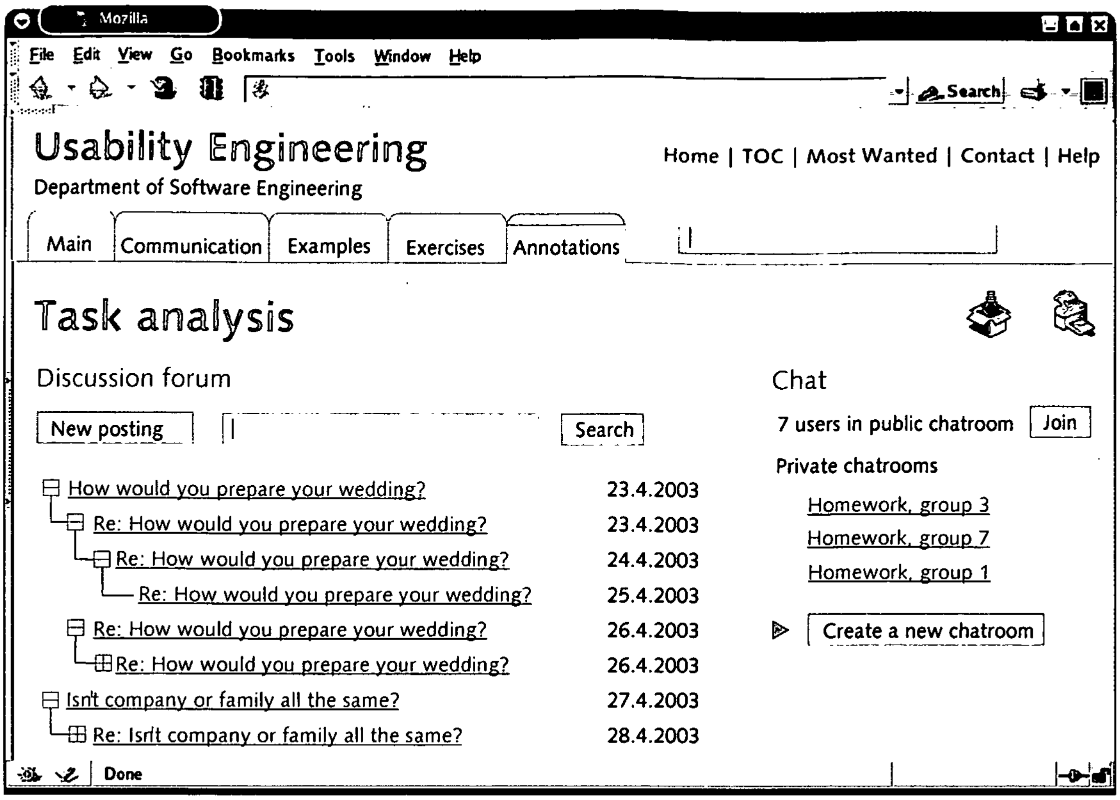


Figure 7.16: This screenshot shows topic specific communication. Left part of the screen contains the discussion forum and the right parts provides access to public and private chatrooms and whiteboards (see figure 7.3.4)

neither the usage of the lectures notes nor their creation need to be linear. *e*-ULE therefore provides a central TOT to group and (re)arrange topics. A *e*-ULE project is not limited to one TOT, different TOTs can provide alternate paths to knowledge. TOTs can also be semi-automatically on the basis of topic-meta information. See section 8.2.2 for details.

Topics can be "stacked", providing information in three (*essential*, *important*, *additional*) difficulty levels (see figure 7.1). Additional flags mark topics as *in progress*, *ready for review*, *reviewed* or *finished* (ready to be published). The all important (and universally disliked) *to-do lists* will be taken care of by *e*-ULE : if a topic misses compulsory parts (like the abstract or a level mark), a need for completion will be listed and it can not change from the *in progress status.* Whole topics, or reviews on topics can be assigned to a specific co-author. The topics of the test site can be published either all at once or individually to the public site.

e-ULE will also help in planning a new course. Every topic can be equipped with

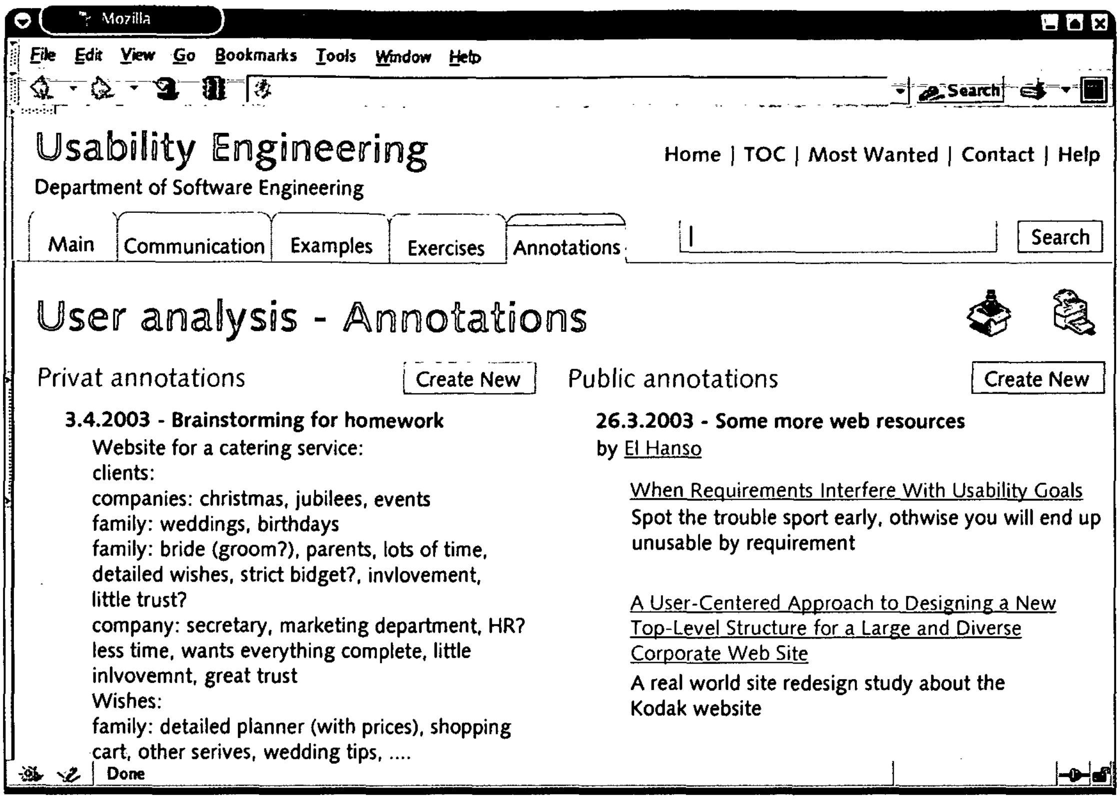


Figure 7.17: Annotations are another means of asynchronous communication. Topics can be publicly and privately annotated. The *e*-ULE project manager can override public annotations.

time frames, both for student self learning and for discussion in lectures. The total time requirement is thus visible and overcrowded curriculums can be avoided.

7.4.8 Step 8: Course information & administration

The information section comprises information about the lecture and the *e*-ULE site. This may include: lecture times / dates and places, an introduction to the *e*-ULE client, contact information, technical support, and other relevant information.

e-ULE is essentially designed to promote grass roots switches to *e*-nhanced learning. Therefore it does not depend on university administration software like room booking systems. A tighter integration with university administration tools can always be implemented as a plug-in.

Registered students can opt to be notified in cases of changes. This can be accomplished via *e*-mail or an optional SMS (Short Message Service) module.

7.4.9 Step 9: Organise students

An *e*-ULE project can be public or restricted to a certain group of students. Student authentification can be accomplished by

- LDAP (Lightweight Directory Access Protocol)
- X500 Whitepages directory services
- Optional interfaces to university data storage
- Automatically generated username/password pairs for distribution during the first lecture
- open access: anybody can pick a user name for personalisation

Additionally an optional *guest account* ensures that a projects content is accessible to the public, whereas the document database that is likely to contain rights managed materials stays inaccessible.

7.5 Usage scenarios

This section highlights some of the uses of the *e*-ULE systems. As the collection of new materials from scratch still is time consuming, the use cases centre around the easy migration to *e*-nhanced learning materials.

7.5.1 Reader based courses – media library

Courses geared towards senior students often are based on "readers". A reader is a collection of materials of different sources, commonly compiled from journal articles, book excerpts and online sources. If most of the material is available digitally it can be easily strung together by a *e*-ULE project. The available material is fed into the document repository and augmented by meta information, abstracts and pointers to the most important parts of the material. Some *e*-ULE topics can provide introductional summaries and connections. This way is probably time neutral especially if the materials is annotated and connected at discovery time (see section 7.5.4), because the usual process of finding the materials, reading it and then forgetting all about till a frantic search for the article in question at the start the new semester is not very time effective either.

7.5.2 New ways to information

e-ULE's TOT concept can provide new insights into available materials. A consequent use of meta data can lead to specialised paths through the content. A *e*-ULE project dealing with art history could assign meta tags for artists, region and years. The materials can thus be accessed by artist, a geographical map or by a time-line. A project on fundamental physics also could include meta tags for persons and times and thus be accessible by time-line and scientist. See section 15 for details about the meta tag system.

7.5.3 Collaborative media generation

Thoroughly to teach another is the best way to learn for yourself.

Tyron Edwards

Another way it reduce building time is to actively involve students in the creation process. This works best with senior students. Explaining and teaching is beneficial to one's own understanding. A whole course and a *e*-ULE project can start by handing out materials to students. It is the students responsibility to digitise and summarise the material. The resulting topics can then be structured and augmented by the lecturer. This project can then serve as a basis for the next course.

7.5.4 Academic publishing system

If the course centres around the one of the lecturer's major research topics a *e*-ULE project can become a time saver. The document and idea repositories are not limited to the creation of lecture notes, they can help in structuring research and in writing research papers and articles. Instead of using a conventional browser for researching information the lecturer uses *e*-ULE's own browser that allows for excerpts and whole articles to be saved and indexed. The just-in-time information agent helps in accessing the stored information and the structure of scientific papers does not differ too much from that of lecture notes. In fact articles for all journals accepting LaTeX format input can be very easily generated from an *e*-ULE project. All others might require some post-production process inside a word processor. With the advent of XML-based journal formats (e.g. (NCBI, 2003)), the generation of articles will be even simpler.

Chapter 8

e-ULE technical considerations

The only thing God didn't do to Job was give him a computer.

I.F. Stone

e-learning naturally involves a certain amount of technology. Technology has a tendency to get into the way (Mitchell et al., 2001). *e*-ULE's aim is to push technology into the background to let lecturers and student alike get on with the more important tasks like information dissemination and digestion. This is accomplished by providing a potent – but low maintenance – server that controls slim clients. This in turn reduces temporal and monetary overhead. The student client in fact does not exist at all as it consists solely of a webbrowser. Figure 8.1 shows all major parts of an *e*-ULE system.

8.1 Free software – opensource software

As university budget usually are slim, *e*-ULE is released as free software ("free speech, not free beer"). For a detailed discussion on free and opensource software and the associated licenses see (Foundation, 1999), (Foundation, 1996a) and (Foundation, 1996b).

Using free or open source software as the basis for an university level *e*-learning environment offers several advantages:

- Operating system can be integrated into the project
- Rely on already existing projects
- Localisation is easily accomplished
- System can be adapted to local environment

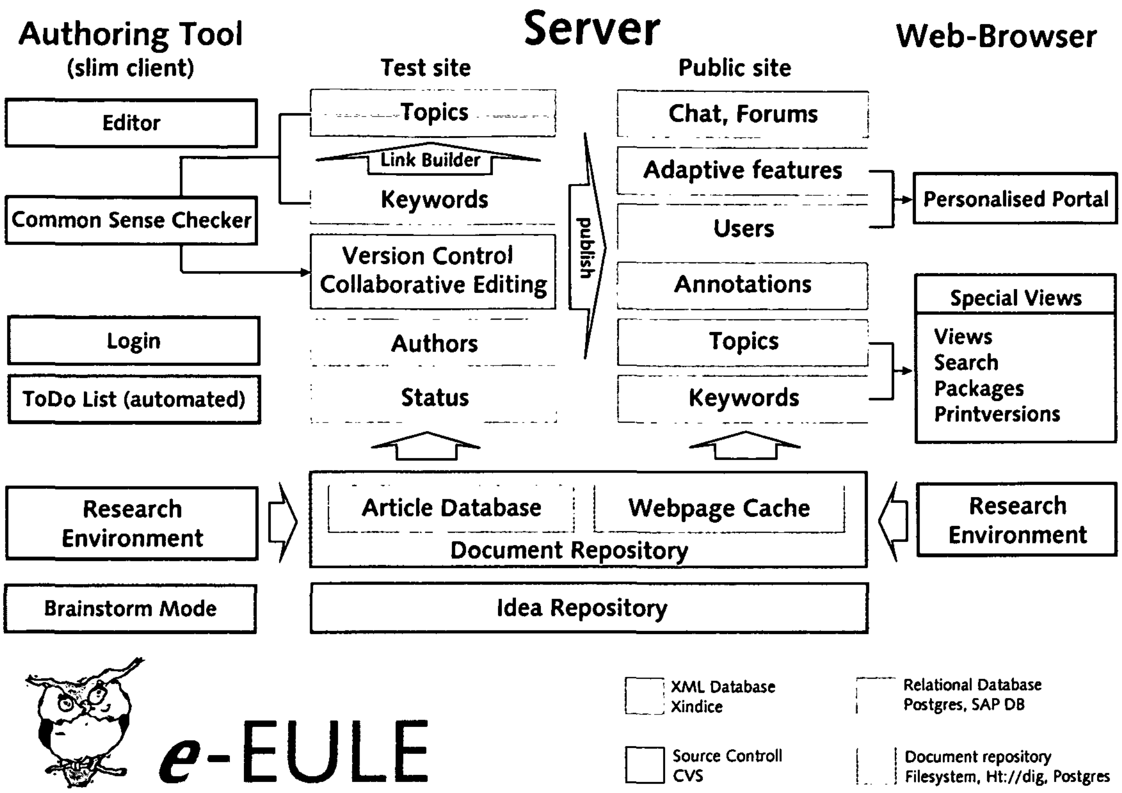


Figure 8.1: Overview of the *e*-ULE system. The server bears most of the features, the slim authoring clients only check out the files and keyword lists to manipulate them offline. Students need only a webbrowser, all "intelligence" is handled by the server.

e-ULE is much more than yet another free software system – it's more like a meta free software project, because it relies heavily on other free software projects available:

- Linux operating system (default server operating system)
- Apache Group Jakarta Java Server
- Apache Group Cocoon XML publishing framework
- PostgreSQL or SAP relational database management systems
- Apache Group Xindice XML database
- Mozilla web-browser and development framework
- Various Mozilla sub-projects including SVG (Scalble Verctor Graphics), MATHML and spell-checking support

In a way *e*-ULE will be a test if large systems can be easily built from smaller free software components.

8.2 Server

The server is the heart of the *e*-ULE system. For the ease of maintenance many of *e*-ULE's features are implemented on the server side to ensure speedy updates. The server is catering to slim authoring clients and the browser-based student clients. The number of *e*-ULE projects is only limited by the servers resources. A single server can serve anything from a single lecture, or a department upto a whole university.

8.2.1 Server tasks

The server's tasks include:

- serving the content
- providing interactivity (forums, chat/whiteboard, annotations)
- providing adaptive features (most frequently used pages, last accessed pages, ...)
- providing custom views
- enabling collaborative editing
- ensuring version control
- student and course management

8.2.2 Content providing

Learning content is stored on the server in form of native *e*-ULE|doc (see section 8.3) XML documents inside a native XML Database. On request the server will deliver HTML files generated from the actual XML code. The server is capable of delivering packaged versions for offline reading or PDF output for printing (see section 8.5.3). A table of contents (and alternate tables-of-content) for each project are stored in relational databases. These are offered as means of navigation. The student is free to add additional views (see section 8.2.2) to the existing tables-of-content.

Interactivity

e-ULE includes a forum/discussion group software and a chat/whiteboard software. These two components can be used in conjunction with every single topic. By attaching a forum or chat to a topic, a new forum group or chat session will be created, dedicated to discussion of the very topic. All discussion groups and chat session can also be accessed via a list of interactive elements.

Annotations are another interactive element. Both private and public annotations are stored on the server in a relational database system.

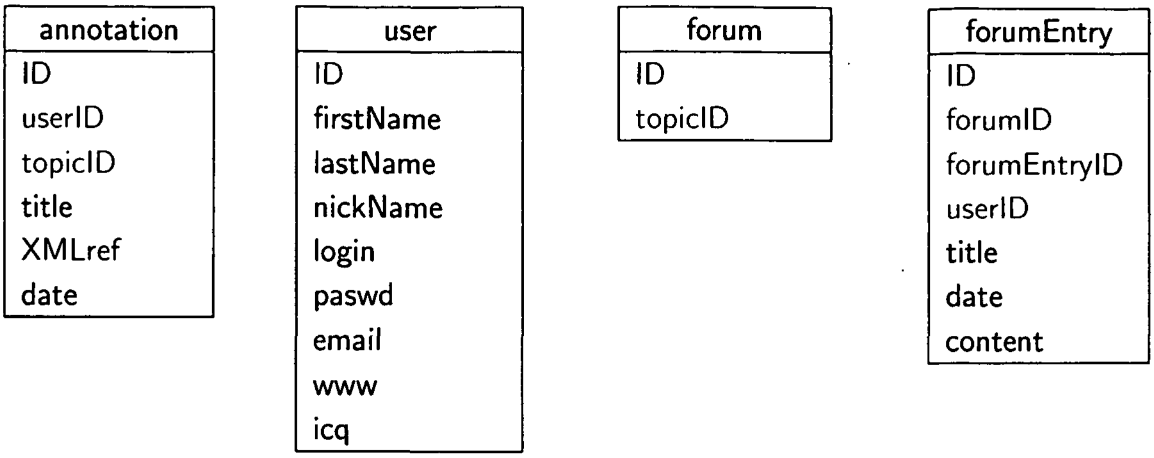


Figure 8.2: Diagram of communication features: discussion groups and annotations.

Adaptive features and custom views

The server also provides adaptive features for the students comfort. This is not to be confused with real adaptive *e*-learning environments like (Brusilovsky et al., 1996). *e*-ULE is able to analyse the students behaviour and build up indeces of the last pages perused or most frequently visited. This again is stored in the relational database, so the feature is bound to the user, not to the client computer.

The lecturer can provide various TOT as starting points into the material. Additionally each student can setup personal TOT, called views e.g. for exam preparation or make excerpts of the material. All views are stored in the relational database and thus are client independent. Figure 8.3 shows an UML (Unified Modelling Language) diagram of the adaptive features.

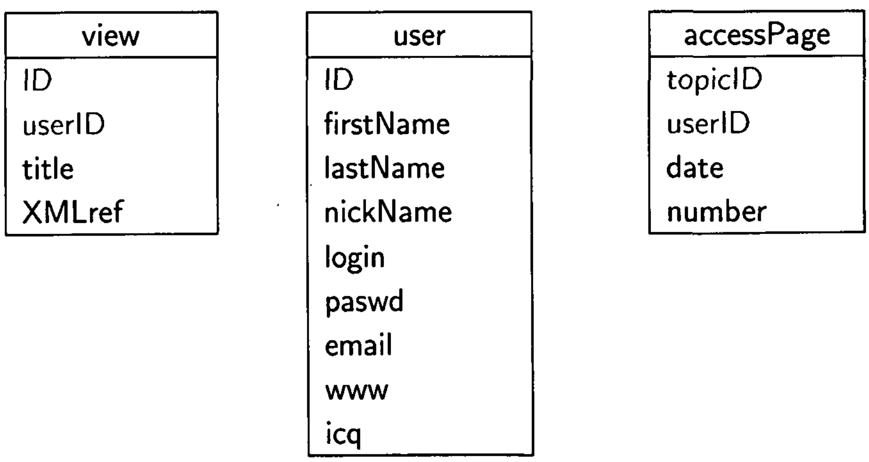


Figure 8.3: Adaptive user features: views, page history and access statistics.

Version control

Any topic is automatically versioned. In addition to the versioning, the topic can be marked up to be in a certain stage: in progress, ready for review, reviewed and public. The versioning is handled via the relational database, whereas the topics themselves are stored inside a native XML database. See figure 8.4 for an overview.

project
ID
name
authorID

tot
ID
title
status
authorID
projectID

block
ID
totID
blockID
title

author
ID
firstName
lastName
login
passwd

topic
ID
blockID

topicVer
ID
topicID
topicVersion
topicVersionName
topicVersionStatus
topicVersionXMLref

keyword
ID
word

keywordTopicVer
keywordID
topicVerID

Figure 8.4: Diagram of the versioning functionality

Collaborative editing

As many lectures are held by a group of lecturers, collaborative editing is another crucial feature. In line with the Constructivistic learning theory, the lecture notes could even be written in cooperation with the students. See section 8.2.2 for details. The server provides collaborative editing. The *editor* can name various *authors* and grant them access to certain parts of his *e*-ULE project. He can assign work to the authors who transparently check it out of the test site along with the latest keyword list. After editing (or inactivity timeout) the respective topic is checked back in, together with the (changed) keyword list. Topics not marked for release are invisible to the public and can only be accessed by authors or editors. User authentification is handled by the relational database.

The *e*-ULE Server includes a *talk-back* function which will report back to the main *e*-ULE site, thus building a network of *e*-ULE sites. *e*-ULE documents from other sites can be integrated seamlessly, but will bear a copyright notice. Of course the editor can shield his own *e*-ULE project or the whole server from unauthorised access or back-reporting.

Student and course management

Student and course management is a central feature in the *e*-ULE system. Any *e*-ULE project is automatically generated with topics pertaining to course management, which just have to be filled in by the author. Today many universities keep online calendars in a format proprietary to the institution. *e*-ULE can be extended by plug-ins to interface which such environments, albeit it might be too difficult a task for just a handful of *e*-ULE projects.

Student management is easier to interface to, as these systems are mostly standard based, relying on LDAP (University of Michigan, 2003) or X500 whitepages (ITU, 1995). Any other solution can be integrated via plug-ins, although again this will only be time effective in the case of a larger number of *e*-ULE projects.

If no connection to the university provided student management is needed (or is not possible), *e*-ULE can take on the task of automatically creating a sufficient number of user accounts in a ready to use format for easy distribution during lectures. Alternatively *e*-ULE can automatically login students on basis of an *e*-mail address belonging to the university range.

Student logins are either handled by *e*-ULE or are referred to central authentification servers (LDAP, X500). In any case *e*-ULE will store a student ID to enable adaptive features and personal views.

Repositories

The server stores documents and ideas inside so called repositories. Documents include scientific papers in PDF or PS format, offline version of HTML pages/sites and legacy documents in various formats such as Microsoft Word or Powerpoint. The documents are stored in the file system and indexed by the fulltext search engine ht://dig (The ht://Dig Group, 2003). Additionally bibliographic information and essential meta data are stored inside the relational database. The bibliographic information can be exported as BibTEX and other popular format to be used for other publications. Depending on the type of publication (book, article, webpage, ...) the bibliographic information consists of author/editor, title, publisher, means of publication, year, volume, number, pages, isbn, issn, edition and series to name the most important. Additional meta data includes comments, a rating (very good – very bad), a level information (beginner, advanced, expert) and categories (review, introduction, ...).

8.2.3 Server structure

The server is logically divided into two parts: the test system and the production system. All new topics start in the test system and are "moved" to the production system on the authors say-so. The production status is achieved via a flag and the author can optionally forego the test system by forcing all topics to the production system by default. Figure 8.5 shows a more detailed view of the server structure and the authoring tool interaction.

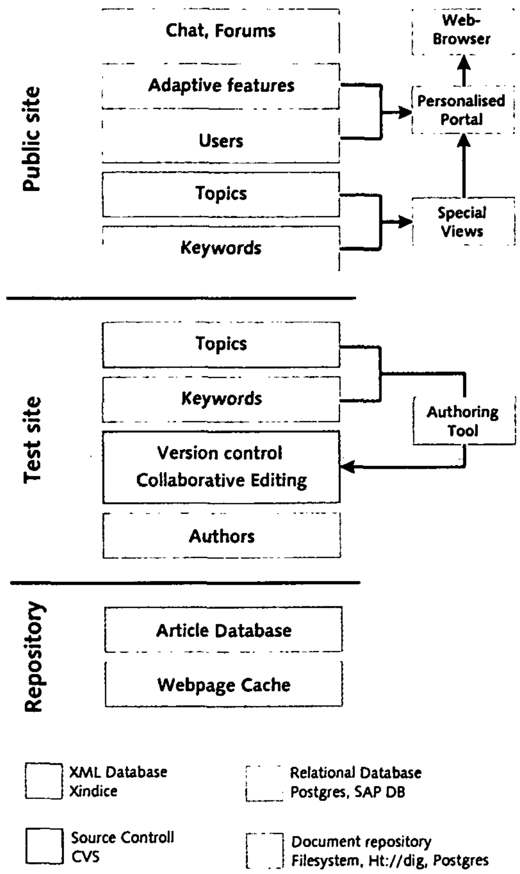


Figure 8.5: Detail of server structure

8.2.4 Programming framework

There are various Web-Application Programming Frameworks available (see (Eden, 2003) for a comprehensive reference). As the system should be free software, and stability and ease of maintenance were major requirements, the field narrowed down quickly to two project groups centreing around the ASF (Apache Software Foundation) (Apache Software Foundation, 2003c): the so called LAMP (Linux – Apache

– MySQL – PHP) solution based on the programming language PHP (Personal Hypertext Processor) and the java based Apache Tomcat Server. As the *e*-ULE projects make much use of XML, LAMP was ruled out as the XML support was not sufficiently developed at the start of the project.

JAVA Servlets – Apache Tomcat Server

Apache Tomcat (Apache Software Foundation, 2003b) is part of the Apache Jakarta Project (Apache Software Foundation, 2003a) whose aim is to build open source solutions on the Java platform. Tomcat is one of the core technologies of the Jakarta project. Tomcat is the servlet container and is known for its standard compliance and stability. In fact it is used in the official Reference Implementation for the Java Servlet and JavaServer Pages technologies.

e-ULE uses Tomcat for the JSP (Java Server Pages) and as a basis for the Cocoon Publishing Framework (Apache Software Foundation, 2003d).

Cocoon XML Framework

Apache Cocoon(Apache Software Foundation, 2003d) is an XML publishing framework based on XML and XSLT (eXtensible Stylesheet Language Transformation) technologies. Designed for performance and scalability around pipelined SAX (Simple API for XML) processing, Cocoon offers a flexible environment based on a separation of concerns between content, logic, and style. Cocoon interacts with most data sources, including filesystems, RDBMS, LDAP, native XML databases, and network-based data sources. It adapts content delivery to the capabilities of different devices by rendering in various formats such as HTML, WML (Wireless Markup Language), PDF (Portable Docuent Format), SVG, and RTF (Rich Text Format).

Cocoon adheres to the MVC modell. It implements a three tier architecture that strictly separates presentation, business logic and data access. Figure 8.6 shows the MVC model.

Cocoon relies on a pipeline model (see figure 8.7): an XML document is pushed through a pipeline, that consists of several transformation steps. Every pipeline begins with a *generator*, continues with zero or more *transformers*, and ends with a *serialiser*. The generator is the starting point for the pipeline. It is responsible for delivering SAX events down the pipeline. Generators usually read from the file system or databases or from other XML processing applications. A transformer can converts an XML document to another XML document (or SAX events). The serialiser is responsible for transforming SAX events to a presentation format. Predefined serialisers exist for generating HTML, XML, PDF, VRML, WML. Custom serialisers can be easily implemented.

e-ULE uses Cocoon to render *e*-ULE|doc documents for online viewing and printing. An *e*-ULE project site is realised in XSP (eXtensible Server Pages) technology, the Cocoon specific alternative to JSP. Both XSP and JSP rely on mixing XML/HTML

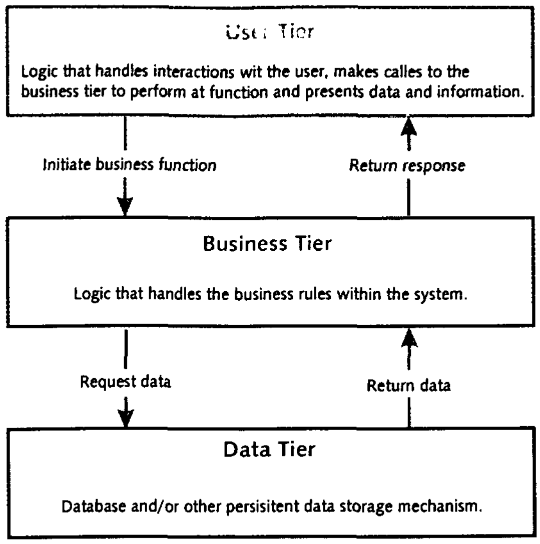


Figure 8.6: The MVC model: separation of user, business and data layer.

code with JAVA code. Cocoon/XSP however go one step further by introducing logic-sheets and taglib that further separate logic from presentation and reduce the amount of JAVA code inside the XML code to a necessary minium. All other programming logic is capsuled into new tags (aggregated into libraries, hence taglibs) and moved to a separate file (see figure 8.8).

8.2.5 Databases

e-ULE makes extensive use of databases. To accommodate the different data types *e*-ULE uses a relational database and a native XML database. As *e*-ULE means to be flexible, the relational database can be substituted against another relational database provided there exists an JDBC (Java DataBase Connectivity) driver and the database is capable of *subselects* and transactions.

Officially supported databases include PostgreSQL and SAP DB but not MySQL (subselects, transactions). Existing installations of DB2 or Oracle can be incorporated. The XML database is also exchangeable as long the replacement conforms to the XML:DB API (XML:DB Initiative, 2003). Currently the choice of open source native XML databases is limited.

PostgreSQL

PostgreSQL (PostgreSQL Group, 2003) is a major open source relational database management system suited for larger projects. It is available for almost every plat-

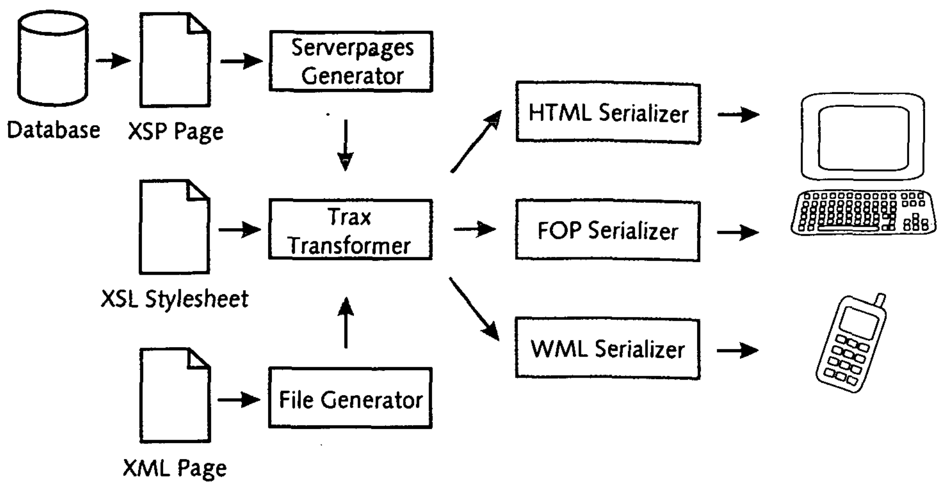


Figure 8.7: Cocoon pipeline: Various input formats and data sources are converted by the appropriate stylesheets and result in HTML, FOP/PDF or WML output.

form, but practical evaluation showed that installation on anything aside a Unix based operating system is difficult. PostgreSQL is the standard database for the *e*-ULE distribution (see section 8.2.6).

SAP DB

As PostgreSQL installation proved difficult (mainly) on MS Windows systems, we searched for an alternative relational database for these platforms. SAP DB (SAP, 2003) fulfils all the requirements. Incidently SAP DB is available for a lot of platforms as well, but is known to show instability on some Linux installations. SAP is therefore as of now only recommended for MS windows based *e*-ULE servers. The additional "goodies" like a graphical administration client ... are of little importance to *e*-ULE .

Xindice XML Database

Apache Xindice (Apache Software Foundation,) formerly known as dbXML is a database specifically designed to store XML data or what is more commonly referred to as a native XML database. The benefit of a native solution is that XML data does not need to be mapped to some other data structure. Data is inserted and extracted as XML. This is most useful when dealing with semi-structured problems (like hypertext files), which can hardly be mapped to a relational database, other than storing it as a BLOB (Binary Large OBject). At the present time Xindice uses XPath for its query language and XML:DB XUpdate for its update language. Xindice provides an implementation of the XML:DB API (XML:DB Initiative, 2003) for Java development.

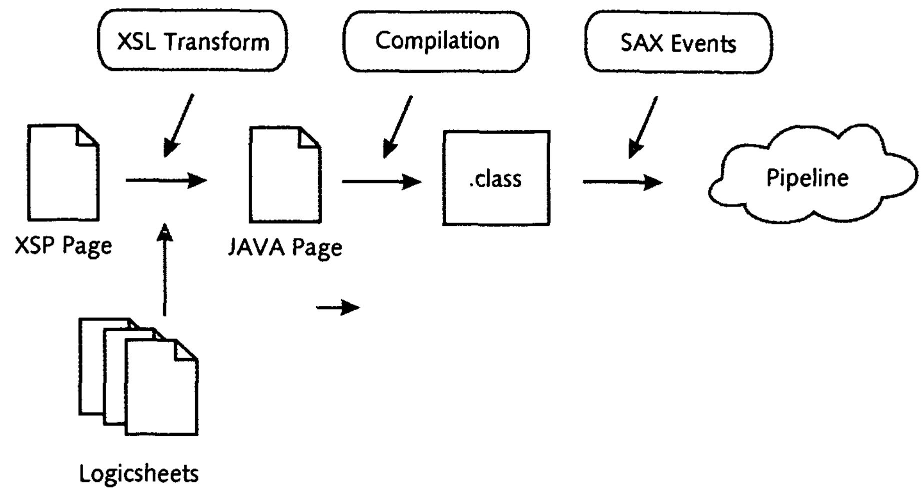


Figure 8.8: XSP pages containing taglib tags are converted by logicsheets (XSL stylesheets) to form JAVA code, which in turn produces SAX events fed into the pipeline.

The newer eXist (Meier, 2003) native XML Database could replace Xindice, as it also supports XML:DB API. As the product is quite new, it is impossible to say which solution will be better in the long run. Currently Xindice has problems with handling large XML files (more than a few MB), a constraint which does not seem to exist in eXist, but is not relevant to *e*-ULE , as *e*-ULE|doc documents are rather small.

8.2.6 Operating system

The client, the authoring tool and the server of *e*-ULE are platform independent. Any common operating system can be used as the basis for the *e*-ULE server provided a JRE (JAVA Runtime Environment) exists for that platform. That includes virtually any operating system in use today. Installing all the components needed to run *e*-ULE however is quite complicated and out of the scope of the average lecturer. As it is not recommended to use a *e*-ULE server for anything else as serving *e*-ULE project pages (security, stability, performance), the Linux operating system is included in the distribution.

The purpose of this distribution is threefold:

- Bootable life system for tryout
- Bootable life system to test hardware compliance
- Easy to use “three click install” to setup a new server

The distribution consists of a downloadable ISO Image which can be burned to a CD-ROM. The CD-ROM is bootable on the PC platform and offers a trial of the *e*-ULE system complete with a sample project (any changes to the material can not be saved permanently though). Besides of offering a "hands-on" glance of the system the CD-ROM checks if the computers hardware is suitable for a *e*-ULE server. If the hardware is suitable the user can install *e*-ULE permanently on the system.

Currently there are several Linux live systems available [1] but one of the most stable and popular is the Knoppix distribution (Knopper, 2003) which is based on the Debian distribution (Debian Project, 2003). Knoppix is marked by a very good hardware detection, that makes sure that it can be booted on nearly any "run-of-the-mill" PC. Knoppix itself is too large and contains many features unnecessary for using *e*-ULE , but several other live-systems have been derived from it, many of which require a much smaller foot-print.

Modern PC hardware is strong enough to run a *e*-ULE system for a whole department, maybe even a small university. If this is not sufficient the platform independence of *e*-ULE insures that the system can be transfered to a stronger computer, e.g. a mainframe system or a cluster. This, albeit, will require at least above average computer skills or an administrator. Future releases of *e*-ULE may incorporate a clustering feature in which several *e*-ULE servers can be joined together.

8.2.7 Up-to-date – zero maintenance

The *e*-ULE distribution will also feature an automatic update. New software releases and security fixes will be downloaded and deployed at the users say-so. As *e*-ULE systems can quickly amass several GB of information, backup is a crucial part. *e*-ULE supports RAID, tape drives and CD writers as backup media and can be integrated into a larger backup system by sharing the *e*-ULE data directory via the SMB (Server Message Block) or NFS (Network File System) protocol or syncing to another machine by using rsync. Given the quick CD installation, it is not necessary to backup anything but the data.

8.3 Document format – *e*-ULE|doc

The *e*-ULE project uses a proprietary XML based data format called *e*-ULE|doc especially engineered to fulfil the needs of a university level hypertext *e*-learning documents. The following section explains the document format and the technology behind it and contrasts other XML based formats.

[1] http://www.tldp.org/HOWTO/Diskless-HOWTO-3.html

8.3.1 SGML, XML and applications

SGML and XML are both languages that are used for defining markup languages. More specifically, they are metalanguage formalisms that facilitate the definition of descriptive markup languages for the purpose of electronic information encoding and interchange. SGML and XML support the definition of markup languages that are hardware- and software-independent, as well as application-processing neutral.

The key philosophy underlying SGML is separating the representation of information structure and content from information processing specifications. Information objects modeled through an SGML markup language are named and described (using attributes and subelements) in terms of what they are (from a defined perspective) not in terms of how they are to be displayed or otherwise processed. XML is a dialect of SGML that is designed to enable "generic SGML" to be served, received, and processed on the World Wide Web. XML originated in 1996, as a result of frustration with the deployment of SGML on the Internet.

Although SGML dates back to late 1960s/early 1970s its core technologies SGML (the modeling framework), DSSSL (the transformation framework for presentation) and HyTime (the linking and timing framework) are very complex and difficult to implement and aroused little interest outside of specialist fields of expertise. The only SGML dialect to receive world wide attention is HTML. XML simplified the requirements for implementation, with the specific intention of enabling deployment of markup applictions on the Internet. The introduction of XML saw a real explosion of new formats including graphics Formats (e.g. SVG), formula languages (e.g. CML, MathML) and specialised document formats of every flavour.

8.3.2 XML educational applications

e-ULE|doc is by far not the first XML based *e*-learning document format. The Open University of the Netherlands has recently released a new version of their EML (Educational Markup Language) (Educational Markup Language) (Open University Netherlands, 2002). Although this specification looks promising, the system is naturally geared towards the needs of a virtual campus, thus containing many concepts that are of no use for *e*-ULE . Major industrial players have consorted and produced the SCORM (Sharable Content Object Reference Model) (Advanced Distributed Learning Initiative, 2002) and the IMS (IMS, 2002) standards for educational environments.

Although these standards are very complete and cover everything from content description to learner data exchange and meta data, they are somewhat of an overkill for a university-centred learning environment. Implementing a system that conforms to these standards is likely to result in yet another "jack of all trades" authoring environment that requires months of training to master. Therefore we decided to create our own standard that integrates more easily with our framework.

More information about XML standards concerning *e*-learning systems can be obtained from Oasis' XML.org (OASIS, 2002). Given, that ease of conversion is one of the main ideas of XML it will be be possible to convert *e*-ULE|doc to and from EML, IMS or SCORM.

8.3.3 *e*-ULE|doc Structure

e-ULE centres around *topics.* Topics are information "atoms" which can not be subdivided[2] meaningfully. A strict process ensures that all meta information necessary to automatically build structures is available. Every topic comes with:

- a title, an optional subtitle, keywords, an abstract
- other meta information (e.g. time, place information, person)
- prequel and sequel information
- level information
- optional extends / requires informations
- optional objects, examples, exercises
- optional server modules (see below)
- optional bibliography and web-links

Figure 8.9 shows the internal topic structure. Detailed information on *e*-ULE|doc is found in appendix A.

Topics are organised via TOT and linked by prequel and sequel identifiers, thus resembling book-like structures. In addition topics come in three flavours: *essential, important* and *additional.* This *level information* allows the student a faster access to the core elements of a lecture. Relations between the levels of the topics are marked by *extends* and *requires* identifiers. Figure 8.10 depicts the interplay between related topics.

Meta information

Abstracts are used to generate a padded table of contents (the often wished for "90-minutes notes" or "executive summary"), which are suitable for a first overview of the lecture (see 5.2.1 - phase 1).

Keywords denote the actual topic and semantic variations thereof. This information is used by the link-builder to automatically create a glossary, an index and appropriate hyperlinks. The keywords play a crucial role in the generation of "views" (see 7.3.2).

[2]You can think about them as subsections, or subchapters

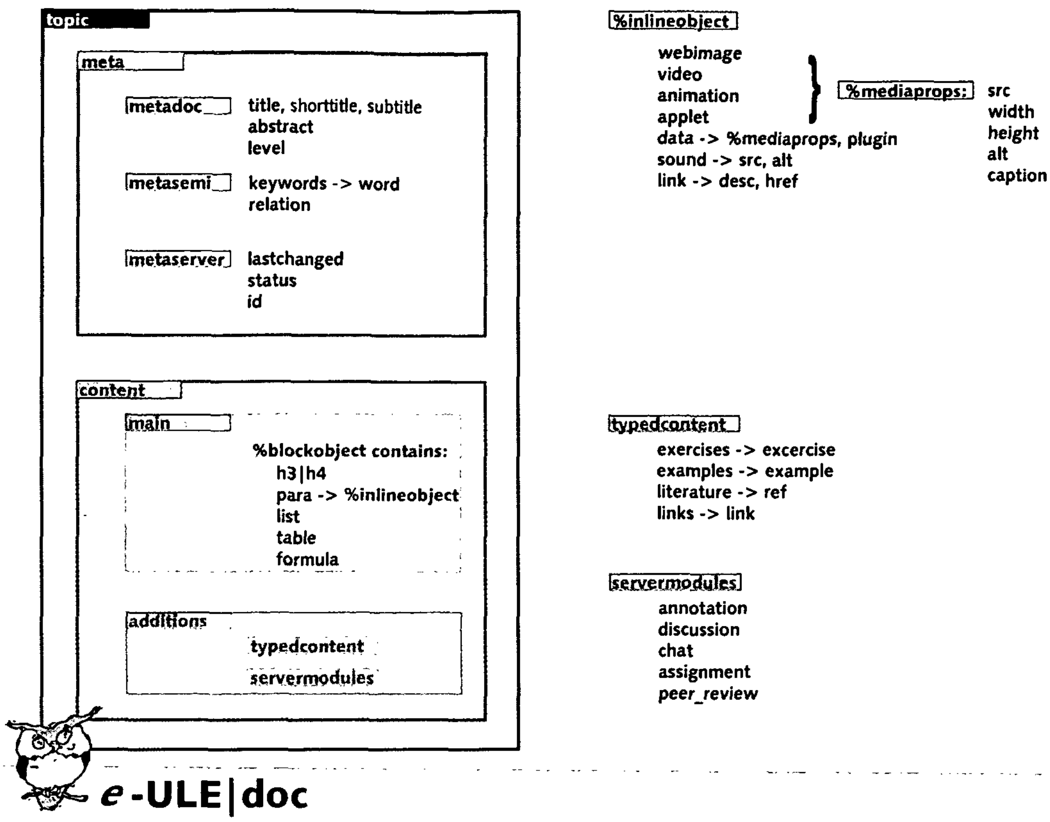


Figure 8.9: The diagram shows the internal structure of a single topic.

Category information is a special type of meta information that allows the creation of special views of the topics. Predefined categories are *person*, *place* and *date* that can automatically construct timeline, regional of person linked views of the content.

Content

Text Text is grouped into paragraphs and can contain internal (inside *e*-ULE) and external hyperlinks as well as objects (graphics, ...) and formulas.

Formulas A lot of university level material makes extensive use of formulas (mathematical and chemical). As of now, the de-facto standard for mathematical formulas is TEX/LATEX whereas chemical formulas are not standardised. The XML family offers two new specialised standards for formulas: MATHML and CML. While MathML is supported natively by the Mozilla web-browser family, CML requires a plug-in. There exists a project dealing with CML to SVG conversion (Murray-Rust and Rzepa, 2003). The Mozilla browser already offers a version that can display SVG natively and plug-ins are available for all major browsers. Till MathML and CML support becomes

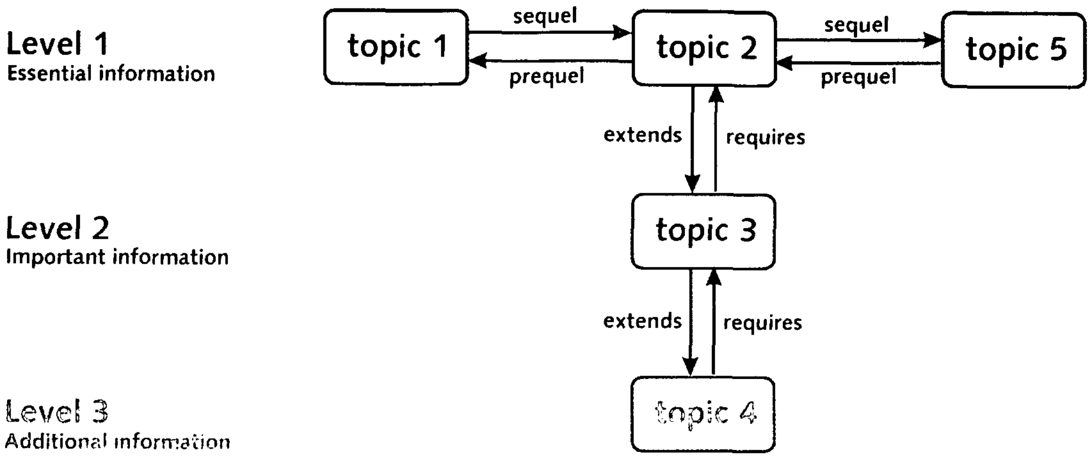


Figure 8.10: The diagram shows the interplay between related topics.

standard, both formats as well as SVG can be converted to GIF/PNG formats by the Cocoon server application. $\TeX$/$\LaTeX$ formulas can also be converted to graphics by the server.

Objects are digital non-textual information that can be transfered via the web, like images, sound or video. Any digital content can be included in a topic via a hyperlink. Common web-bounded formats as GIF, JPEG or MP3 are included automatically, other formats are converted on the server side. Large images are automatically reduced and re-linked by the server. The GNU platform offers a host of converting tools, e.g. imagemagic, pnmtools. In contrast to common authoring systems object handling in *e*-ULE requires no detailed format knowledge on the authors part. Objects that can not be automatically converted by the server, will be included as references to external files.

Additional information

This part of the document lists bibliographic information and references to additional online or printed resources. This part can also integrate automatically generated, content-based links to documents residing in the document repository.

Practice

This part of the document contains (optional) examples, exercises and solutions. Examples demonstrate the practical application of the theoretical content of the chapter, whereas exercises shall stimulate the student into further action. Solutions to the exercises or a facility to submit assignments may be provided.

Communication

Discussions & chat

Communication is a vital aspect of learning. Since the aim of the *e*-ULE system is to further student - lecturer communication by reducing the pure information transfer aspect of the lectures there is room for communication in the lecture hall. Still there will be the need to communicate electronically: In the forums students can talk to each other, to the lecturer or to tutors. Different communication needs are served by different types of communication channels: asynchronous, newsgroup like systems (forums) and synchronous chat systems.

Annotations

Annotations are another means of communications. In a notice board manner they can serve as platform for collecting additional information about each topic, like errata, implementation examples, notes about practical applications or additional web-links.

8.4 Authoring tool

e-ULE relies on a XML-based as document format called *e*-ULE|doc (see section 8.3). *e*-ULE|doc incorporates other XML applications (languages) like MATHML or CML which will be most useful in the university field. Being text-based *e*-ULE|doc documents can be generated by any XML editor, but the *e*-ULE authoring tool offers many "convenience features".

As university computer equipment tends to be heterogenous rather than homogenous the *e*-ULE authoring tools is designed to multi-platform. Binaries will exist for Linux, MS Windows and Mac OS systems, but the open source will allow for porting to other platforms as well.

Looking at various existing authoring systems, one can find many different approaches to the subject: WEAR (Moundridou and Virvou, 1992) and AHA (Bra and Calvi, 1998) use HTML files which are augmented with meta information on the server side, while Interbook (Brusilovsky, 1998) relies on specially structured MS-Word files plus concept-based annotations. WebCT (WebCT, Inc, 2003) provides a server based HTML forms/javascript interface.

To avoid the limitations of browser-based editors (few features, slow responses, only suitable for one type of browsers) as well as the administrative and maintenance overhead of full blown authoring workbenches, we designed the authoring tools as slim client to the *e*-ULE server. Thus everything aside the entering of content will be handled by the server.

8.4.1 Development frameworks

Cross-platform development tools

There are only a few (free) multi-platform development frameworks available. We decided to analysed the options and decided to evaluate the possibilities of a JAVA client and a Mozilla based client in detail. The popular QT framework (Trolltech Inc., 2003) did not enter the competition, as a GPL (Gnu Public License)ed version of the Mac libraries was not available at that time. Various other frameworks where rejected because they lacked sufficient XML support or had no HTML rendering widgets.

Java client

JAVA was designed to facilitate cross-platform development. However, nowadays it is mostly used for single-platform server applications (*e*-ULE uses a JAVA based back-end). JAVA offers excellent XML support, web-install features for zero administration installs and many more goodies, but lacks in HTML/XML rendering and web-browsing. No free components for this tasks could be found. JAVA also lacks in the department of spell checking, which is an important basis for the Common-Sense-Checker (see section 8.4.2).

Mozilla framework

The Mozilla framework is a very young player in the field of cross-platform development. Originally designed to facilitate the creation of the user interface for the Mozilla (Mozilla group, 2003b) web-browser, it has developed a life of its own. The Mozilla framework is especially useful for HTML/XML centric projects (e.g. Komodo IDE (ActiveState, 2003)). The Mozilla web-browser itself has a modular structure. The main components is the rendering engine *Gecko* which also generates the user interface, written in XUL (eXtensible User interface Language). There are modules for editing (*Composer*), *e*-mail and news (*Thunderbird*) as well as for spell-checking (Mozdev Oragnisation, 2003b) and instant messaging via IRC *Chatzilla* (Mozilla group, 2003a) and *Jabberzilla* (Jabberzilla Project, 2003). Additionally Mozilla boast SVG and MathML support (Fritze, 2003; Sidje, 2003) and a LDAP interface (Mozilla Organisation, 2003a).

The Mozilla framework is component based. These components (*XPCOM*) are developed in C and connected by XUL/javascript based GUI. Mozilla applications can be deployed via the integrated installation and distribution system *XPinstall* (Mozilla Organisation, 2003c). Currently many different Mozilla based projects are being developed at Mozdev.org (Mozdev Oragnisation, 2003a). Among those is a graphical MathML editor (Swanson, 2003) that will also be used *e*-ULE to generate MathML formulas.

Despite the fact the Mozilla Framework is far from mainstream technology, it is ideally suited for the *e*-ULE development. The sparse documentation is leveraged by the host of components that can readily be included into *e*-ULE . Detailed information on the development with the Mozilla framework and the development of extension with XPCOM can be found in the following books (Turner and Oeschger, 2003; Bullard et al., 2001; Boswell et al., 2002) and on the web (Mozilla Organisation, 2003b; O'Reilly, 2003).

8.4.2 Common sense checker & auto linker

e-ULE features semi-automatic hyperlinking. This hyperlinking is achieved via keywords. Every keyword corresponds to one or more topics which offer detailed information on the keyword. Keywords are incorporated into the meta-information of the topic. Every occurance of the keyword or a grammatical derivation thereof inside the text automatically creates a set of hyperlinks connected to the keyword.

In addition to the keywords in the meta information, every word of the text can turned into a keyword by the user. The common sense checker helps the author in identifying useful keywords. In a process similar to a spell-checker, the authoring tool will mark all unknown words. As we are dealing with academic texts, any word not found inside the spell checker dictionary has an increased likelihood of being a keyword. The actions on the unknown word include: correct spelling mistake (choose from list), mark as new keyword, mark as a deviation from a known keyword (choose from list). See figure 7.4 for an example.

New keywords come in two flavours: prerequisites, which are used to generate a special list of "should know" items and actual keywords. The actual keywords are immediately associated with a new topic and a glossary entry. Every topic can be manually associated with several keywords. These entries can be selected from each occurrence inside the text.

There are currently two large free spell-checking system: aspell (Kevin Atkinson, 2003) (pspell, ispell) and the open office spell checker MySpell (Open Office Group, 2003) that is also used by the Mozilla spell-checker project (Mozdev Oragnisation, 2003b). MySpell in fact uses modified ispell dictionaries. Although aspells quality is superior, *e*-ULE will rely on MySpell because of the easier integration and better platform support (aspell runs only on Unix platforms, compilation runs under MS Windows have so far been unsuccessful).

The server maintains a list of all keywords. The list will be sent to the authoring tool upon checkout of a topic, and the topic and the list will be retransmitted to the server in regular intervals or on saving events in the authoring tool. Words contained in the keyword list (and grammatical derivations thereof) will be automatically marked as keywords. In addition, an autonomous server process (e.g. run nightly) checks all files for new keyword occurance.

8.4.3 Just-in-time information

Every lecturer or scientist is bound to amass virtually tons of information. Frequently information turns up by serendipity. Of course it is squirrelled away for "later perusal", but most of the time this information will never be accessed again, because its owner may not remember it was ever stored in the first place.

"A just-in-time information retrieval agent (JITIR agent) is software that proactively retrieves and presents information based on a person's local context in an easily accessible yet nonintrusive manner." Rhodes describes the concept in (Rhodes and Maes, 2000) and (Rhodes, 2000). Rhodes also developed the *Remembrance Agent* (Rhodes, 2002) for the Emacs editor. The system basically consists of two parts: the indexer will create indices of text collections and the agent will analyse user input and lookup terms in the index. Matches are then announced to the user. *e*-ULE will incorporate Rhodes technology to assist the lecturer in regaining information stored in the document and idea repository and existing *e*-ULE topics. See figure 7.15 for an example.

8.5 Client

The client, in fact, is a web-browser. For the sake of usability we decided against a specialised client to minimise the need for technical support.

The major drawback in relying on the server to handle all student side interaction lies in the fact, that a permanent connection to is needed. This disadvantage is mitigated by the possibility to download chunks of the material for offline viewing in the browser, allowing the use of dial-up connections to the Internet. As high bandwidth flatrate Internet access is increasing, this will not pose much of a problem in the near future.

The creation of specialised versions or views (see 7.3.2) of the content is left to the server, the result can be browsed in the usual manner. The server also takes care of necessary LaTeX or MathML formular to GIF conversions.

The Cocoon framework uses XSLT stylesheets to produce the various output formats. The user can choose between a feature-rich version designed for modern browsers and a serialised version for other clients. Additional stylesheets handle offline and print versions.

8.5.1 Modern browsers

Today's standard browsers like IE5+, Opera 6+, Netscape 6+, or Mozilla 1+ are sufficiently equipped to allow for application-like handling of *e*-ULE pages. The standard interface uses javascript and CSS (Cascading Stylesheets) and makes use of the now commonly found 17" displays with a resultion of 1024x768 points.

Modern browsers can integrate various specialised file formats via plug-ins, and some (Mozilla has just completed MathML support) are even able to natively display XML applications such as MathML or SVG.

8.5.2 Older/special browsers

To avoid frustrating users of older browsers or handicapped people requiring more flexibility of the layout, there exists a serialised version of the *e*-ULE documents that is suitable for older browsers, braille output or screen readers and allows for very large text sizes. This output is also usable on very small screens like those of sub-notebook computers or PDAs. To enhance accessibility a *e*-ULE page comes complete with keyboard accelerators and link short-cuts.

8.5.3 Format conversion

Printing versions are generated via LaTeX. The W3C formating language XSL-FO (eXtensible Styleheet Language Formating Objects) is currently not good enough to produce acceptable outcome. The XML – LaTeX – PDF conversion is integrated into the Cocoon pipeline as an additional set of transformers and serialisers.

8.6 Accessibility

Accessibility is an important feature in *e*-ULE . Major player in the field (e.g. (WebCT, Inc, 2003)) are currently suffering from the effects of tighter laws on accessibility of software products. Basically accessibility is not very difficult to achieve, it is more question of making the effort, than a technological problem. *e*-ULE already features a set of stylesheets, that renders content in a serialised "no frills"' version. This version is ideally suited for accessibility enhancement. Depending on the type of impairment, several enhancements are necessary:

visual impairment

- Layout tolerant to very large font sizes
- Layout suitable for braille display
- Layout suitable for screen readers (text to speech)
- Possibility to skip navigational menus
- Additional text information for all graphical elements
- Keyboard based navigation
- Keyboard accelerators

Motoric impairment, e.g. spasticity

- Layout tolerant to very large font sizes
- No small navigational elements
- Keyboard based navigation
- Keyboard accelerators

e-ULE supports all these enhancements and the presentation style can be changed in the (fully accessible) login page. More information on web accessibility is a available from the W3Cs WAI (Web accessibility Initiative) (World Wide Web Consortium, 2003).

Chapter 9

Summary and prospect

This work has shown an ab initio approach to designing a university level teaching and learning environment with a strong focus on usability on lecturer/author and student/reader side.

In contrast to many ITS/ILE, this system does not aim at replacing the lecturer (Shackelford, 1990), or the university as a whole, but shall lift some of the burden of information transfer from the lecturer and thus allow the students to spend more "quality time" with their teacher.

An *e*-(nhanced) learning environment can only be successful if it fulfils student and lecturer need alike. The student needs to be supported in various stages of learning, whereas the lecturer can't afford to spent more time on generating lecture support materials.

Investigation of lecturer and student requirements resulted in the concept and design of *e*-ULE (*e*-Usable Learning Environment), a university level teaching and learning environment with a strong focus on usability.

In order to ensure learning materials to be helpful for students in any learning situation, from gaining an overview to exam preparation and further reference, an equally usable authoring tool is required: *e*-ULE 's authoring system is geared towards a typical lecturer, requiring no undue amount of IT or pedagogical skills, but offering support for the academic workflow by supporting tasks like literature research and integration, and collaborative editing in large groups (e.g. together with students).

Another major aspect of the *e*-ULE authoring tool is the semi-automatic keyword-based hyperlink generation, which ensures correctly and meaningfully hyperlinked pages, without burdening the author with the actual process of link generation. The keyword system also allows the author to identify prerequisites for the course.

e-ULE keeps track of work in progress and imposes a minimum structure upon the edited document, thus building up to-do lists of missing parts.

In compliance with the meagre university budgets the system is opensourced and relies on several prominent opensource projects.

e-ULE is an ongoing project. At the time of writing, the server backend is finished, the Server API is being implemented, the server is being documented and, transformation stylesheets are being built. Work on the system distribution and converters (LaTeX $\leftrightarrow$ <XML>) has started. There also is some ongoing effort to create a sample project to provide a show case of student benefits.

Earlier rounds of proof-of-concept developments found that the common-sense-checker is feasible, as is the use of the Mozilla framework (Mozilla group, 2003b) as development framework. Development of the authoring tool started in summer 2003 with a detailed usability study of paper mockups and is scheduled to finish in 2004.

e-ULE's workflow centric approach roused quite an interest in the scientific community. Various lecturers have already voiced their interest in an early trial version and several are eager to participate in usability tests of the mockup.

It is our hope that the strict user centred approach of *e*-ULE will not only delight its users, but also incite the competition to produce more usable *e*-learning software.

The *e*-ULE home can be found at http://eule.swt.tuwien.ac.at/.

Chapter A

e-ULE|doc

This appendix proposes a preliminary DTD (Document Type Definition) of the *e*-ULE|doc format described in section 8.3. Some deviations are still to be expected and the final version will be a XSD (eXtensible Schema Description) schema rather than a DTD. A XSD Schema offers better integration of other XML-based formats such as MATHML or CML.

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<!-- ============================================================ -->
<!--                   e-ULE doc: topic DTD                       -->
<!-- ============================================================ -->



<!-- ============================================================ -->
<!--                   Notations                                  -->
<!-- ============================================================ -->

<!NOTATION gif         SYSTEM "image/gif"                          >
<!NOTATION jpeg        SYSTEM "image/jpeg"                         >
<!NOTATION png         SYSTEM "image/png"                          >

<!--                   Entities                                   -->



<!-- ============================================================ -->
<!--                   Parameter Enities                          -->
```

```
<!-- ============================================================= -->

<!--                     headings inside the topic: 2 levels should
                         suffice: h3, h4 in compliance with HTML    -->
<!ENTITY % heading      "h3|h4"                                      >


<!--                     text in different representations:
                         paragraphs, headings, list, table, etc     -->
<!ENTITY % blockobject  "h3|h4|para|list|table|formula"              >

<!--                     inline objects (inside para)               -->
<!--                     inline objects swim with the textual
                         content and inlcude images, media, etc     -->
<!ENTITY % inlineobject "webimage|sound|video|animation|
                          link|applet|data"                          >



<!--                     properties common to digital media         -->
<!ENTITY % mediaprops   "src, width, height, alt, caption"           >

<!--                     applet properties: java applets            -->
<!ENTITY % appletprops  "javaparam"                                  >



<!-- ============================================================= -->
<!--                     complex elements                           -->
<!-- ============================================================= -->


<!-- ============================================================= -->
<!--                     root element                               -->
<!-- ============================================================= -->
<!--                     topic                                      -->
<!--                     root element                               -->
<!ELEMENT  topic         (meta,content)                              >


<!-- ============================================================= -->
<!--                     meta information                           -->
<!-- ============================================================= -->
```

```
<!--                    meta                                    -->
<!--                    all meta information                    -->
<!ELEMENT  meta         (metadoc, metasemi, metaserver)           >

<!--                    metadoc                                 -->
<!--                    meta information about the content      -->
<!ELEMENT  metadoc      (title, shorttitle?, subtitle?,
                         abstract, level)                         >

<!--                    metasemi                                -->
<!--                    meta information seminautomatically
                        generated                               -->
<!ELEMENT  metasemi     (keywords, relation)                      >

<!--                    keywords                                -->
<!ELEMENT  keywords     (word+)                                   >


<!--                    relation                                -->
<!--                    prequel, sequel information
                        semiautomatically generated from the TOT -->
<!ELEMENT  relation     (prequel,sequel)                          >


<!--                    metaserver                              -->
<!--                    meta information pertaining to the server -->
<!ELEMENT  metaserver   (lastchanged, status, id)                 >


<!-- ============================================================ -->
<!--                    content                                 -->
<!-- ============================================================ -->

<!--                    actual content of the topic             -->
<!ELEMENT  content      (main,additions?)                         >

<!--                    main                                    -->
<!--                    text and inline objects                 -->
<!ELEMENT  main         (#PCDATA|%blockobject;)*                  >
```

```
<!-- ============================================================ -->
<!--                      block objects                          -->
<!-- ============================================================ -->

<!--                      h3                                      -->
<!ELEMENT  h3             (#PCDATA|em|keyword)*                    >
<!--                      h4                                      -->
<!ELEMENT  h4             (#PCDATA|em|keyword)*                    >

<!--                      para                                    -->
<!--                      parageaph: container for text and other
                          inlineobjects                           -->
<!ELEMENT  para           (#PCDATA|%inlineobject;)*                >


<!--                      list                                    -->
<!--                      all types of lists                      -->
<!ELEMENT  list           ((desc*,item)+)                          >
<!--         type         which list do you want                  -->
<!ATTLIST  list
             type         (number|bullet|description)    #REQUIRED >


<!--                      formula                                 -->
<!--                      placeholder for formulars               -->
<!ELEMENT  formula        ANY                                      >

<!--                      table                                   -->
<!--                      container for tabular material, comp. html -->
<!ELEMENT  table          (tr+,caption?)                           >
<!--                      tr                                      -->
<!ELEMENT  tr             (td+)                                    >
<!--                      td                                      -->
<!ELEMENT  td             (#PCDATA)                                >
<!ATTLIST  td
             colspan      CDATA                          #IMPLIED
             rowspan      CDATA                          #IMPLIED
             width        CDATA                          #IMPLIED
             height       CDATA                          #IMPLIED
             align        (left|center|right)            #IMPLIED
             bgcolor      CDATA                          #IMPLIED
             valign       (top|bottom|baseline|middle)   #IMPLIED  >
```

```
<!-- ============================================================ -->
<!--                    inline objects                            -->
<!-- ============================================================ -->

<!--                    webimage                                  -->
<!--                    image types common to the web: gif, jpg,
                        png, require no plugin                    -->
<!ELEMENT  webimage     (%mediaprops;)                             >
<!ATTLIST  webimage
             type       NOTATION   (gif | jpeg | png)      #REQUIRED
             align      (left|center|right|inline)         #REQUIRED >

<!--                    sound                                     -->
<!ELEMENT  sound        (src,alt,caption?)                         >

<!--                    video                                     -->
<!ELEMENT  video        (%mediaprops;)                             >
<!--                    animation                                 -->
<!--                    flash or other animations                 -->
<!ELEMENT  animation    (%mediaprops;)                             >

<!--                    link                                      -->
<!--                    hyperlink                                 -->
<!ELEMENT  link         (name, href, linktitle?)                   >

<!--                    applet                                    -->
<!--                    java applets                              -->
<!ELEMENT  applet       (%mediaprops;, %appletprops;)              >
<!--                    javaparam                                 -->
<!--                    paramezter for java applet                -->
<!ELEMENT  javaparam    (javaname,javavalue)                       >

<!--                    javaname                                  -->
<!--                    name of a java parameter                  -->
<!ELEMENT  javaname     (#PCDATA)                                  >
<!--                    javavalue                                 -->
<!--                    value of java parameter                   -->
```

```
<!ELEMENT  javavalue     (#PCDATA)                                 >


<!--                     expert data types                       -->
<!--                     data types not normally supprted by web
                         browser - require plug-ins, or seperate
                         download                                -->
<!ELEMENT  data          (%mediaprops;, plugin)                    >

<!--                     plugin                                  -->
<!--                     which plug in to load                   -->
<!ELEMENT  plugin        (name, desc, href)                        >



<!-- ======================================================== -->
<!--                     additional content                      -->
<!-- ======================================================== -->


<!--                     additions                               -->
<!--                     addtional content: teaching operations,
                         references, links, server modules       -->
<!ELEMENT  additions     (typedcontent+,servermodules+)            >


<!-- ======================================================== -->
<!--                     typed content                           -->
<!-- ======================================================== -->

<!--                     typedcontent                            -->
<!--                     teaching operations and references      -->
<!ELEMENT  typedcontent  (examples?, exercises?,
                         literature?, links?)                      >


<!--                     exercises                               -->
<!--                     exercises left for the students         -->
<!ELEMENT  exercises     (exercise*)                               >
<!--                     exercise                                -->
<!ELEMENT  exercise      (name, desc, exercise_content, solution)  >
<!--                     exercise_content                        -->
<!ELEMENT  exercise_content
```

```
                                  (%blockobject;)*                              >
<!--                              solution                                    -->
<!ELEMENT  solution               (%blockobject;)*                              >



<!--                              examples                                    -->
<!--                              commented examples                          -->
<!ELEMENT  examples               (example*)                                    >

<!--                              example                                     -->
<!ELEMENT  example                (name, desc, example_content)                 >

<!--                              example_content                             -->
<!ELEMENT  example_content            (%blockobject;)*                          >



<!--                              references                                  -->
<!--                              reference used work                         -->
<!ELEMENT  literature             (ref*)                                        >

<!--                              ref                                         -->
<!--                              literature references                       -->
<!ELEMENT  ref                    (author, title, year,
                                   (journal|proceeding|link), volume?, issue?) >
<!--         localcopy            identifier of a locally stored copy of the
                                  material                                    -->
<!ATTLIST  ref
             localcopy            CDATA                              #IMPLIED >


<!--                              links                                       -->
<!ELEMENT  links                  (link*)                                       >


<!-- ============================================================ -->
<!--                              server modules                              -->
<!-- ============================================================ -->

<!--                              servermodules                               -->
<!ELEMENT                         servermodules (annotations?, discussions?,
                                  assignements?, peerreviews?)                  >
```

```
<!--                          annotations                           -->
<!ELEMENT  annotations   (annotation*)                              >
<!--                          annotation                            -->
<!ELEMENT  annotation    (by,atitle,(%blockobject;)*)               >

<!--                          assignements                          -->
<!ELEMENT  assignements  (assignment*)                              >
<!--                          assignment                            -->
<!--                          homework, to handed in                -->
<!ELEMENT  assignment    (name, desc, ass_content)                  >
<!--                          exercise_content                      -->
<!--                          ass_content                           -->
<!ELEMENT  ass_content   (%blockobject;)*                           >

<!--                          peerreview                            -->
<!--                          homework reviewed by other students   -->
<!ELEMENT  peerreviews   (peerreview)                               >
<!--                          peereview                             -->
<!ELEMENT  peerreview    (%blockobject;)*                           >


<!--                          discussions                           -->
<!--                          interactive elements                  -->
<!ELEMENT  discussions   (forum?,chat?)                             >
<!--                          forum                                 -->
<!--                          discussion forum                      -->
<!ELEMENT  forum         (#PCDATA)                                  >
<!ATTLIST  forum
             id          CDATA                            #REQUIRED >
<!--                          chat                                  -->
<!ELEMENT  chat          (#PCDATA)                                  >
<!ATTLIST  chat
             id          CDATA                            #REQUIRED >




<!-- ============================================================== -->
```

```
<!--                    simple elements                           -->
<!-- =========================================================== -->


<!--                    titles                                    -->
<!ELEMENT   title       (#PCDATA)                                  >
<!ELEMENT   subtitle    (#PCDATA)                                  >
<!ELEMENT   shorttitle  (#PCDATA)                                  >
<!--                    abstract                                  -->
<!ELEMENT   abstract    (#PCDATA)                                  >
<!--                    level                                     -->
<!ELEMENT   level       (#PCDATA)                                  >


<!--                    word                                      -->
<!ELEMENT   word        (#PCDATA)                                  >
<!--                    prequel                                   -->
<!ELEMENT   prequel     (#PCDATA)                                  >
<!--                    sequel                                    -->
<!ELEMENT   sequel      (#PCDATA)                                  >


<!--                    lastchanged                               -->
<!ELEMENT   lastchanged (#PCDATA)                                  >
<!--                    status                                    -->
<!--                    status of the topic: in work, ready for
                        review, reviewed (ready to publish)       -->
<!ELEMENT   status      (#PCDATA)                                  >
<!--                    id                                        -->
<!--                    unique id number for every topic          -->
<!ELEMENT   id          (#PCDATA)                                  >


<!--                    em                                        -->
<!ELEMENT   em          (#PCDATA)                                  >


<!--                    keyword                                   -->
<!ELEMENT   keyword     (#PCDATA)                                  >
```

```
<!--                    name                                    -->
<!ELEMENT  name         (#PCDATA)                                >
<!--                    desc                                    -->
<!ELEMENT  desc         (#PCDATA)                                >
<!--                    href                                    -->
<!ELEMENT  href         (#PCDATA)                                >
<!--                    linktitle                               -->
<!--                    additional information about this link  -->
<!ELEMENT  linktitle    (#PCDATA)                                >

<!--                    src                                     -->
<!--                    image source                            -->
<!ELEMENT  src          (#PCDATA)                                >
<!--                    width                                   -->
<!--                    object width                            -->
<!ELEMENT  width        (#PCDATA)                                >
<!--                    height                                  -->
<!--                    object height                           -->
<!ELEMENT  height       (#PCDATA)                                >
<!--                    alt                                     -->
<!--                    alternate for image                     -->
<!ELEMENT  alt          (#PCDATA)                                >

<!--                    caption/description                     -->
<!--                    object description/caption, can be added
                        under the object                        -->
<!ELEMENT  caption      (#PCDATA)                                >


<!--                    author                                  -->
<!ELEMENT  author       (#PCDATA)                                >
<!--                    year                                    -->
<!ELEMENT  year         (#PCDATA)                                >
<!--                    journal                                 -->
```

```
<!ELEMENT   journal      (#PCDATA)                    >
<!--                     proceeding                 -->
<!ELEMENT   proceeding   (#PCDATA)                    >
<!--                     volume                     -->
<!ELEMENT   volume       (#PCDATA)                    >
<!--                     issue                      -->
<!ELEMENT   issue        (#PCDATA)                    >


<!--                     item                       -->
<!--                     list item                  -->
<!ELEMENT   item         (#PCDATA)                    >


<!--                     by                         -->
<!--                     annotations by xx          -->
<!ELEMENT   by           (#PCDATA)                    >
<!--                     atitle                     -->
<!--                     title of the annotation    -->
<!ELEMENT   atitle       (#PCDATA)                    >
```

Acronyms & Glossary

List of Acronyms

ADL .... Advanced Distributed Learning   Organisation involved in the creation of the e-learning family of standards SCORM.

AIM .... AOL Instant Messenger   AIM is a representative of the growing number of IMS.

ASF .... Apache Software Foundation   A goup developing and maintaning various important web-realted free software projects. To be found at http://www.apache.org/

BLOB .... Binary Large OBject   Datatype for storing binary data such as images inside a relational database system.

CASE......... Computer Aided Software Engineering   Software that generates source code.

CBT .... Computer Based Training   All types of computer mediated training, especiallly educational CD-ROMs and Edutainment software.

CERN .... Centre Europeene a la Reserche Nucleaire   European particle/high energy physics research lab near Geneva, Switzerland. Tim Berners Lee developed an in-house hypertext information system that later became know as the WWW.

CGI........... Common Gateway Interface   Standardised means for communication between web-server software and other software programs.

CML.......... Chemical Markup Language   A XML based markup language for writing chemical formulas.

CMS .... Content Management System Software to facilitate the collaborative creation of web content. CMS aim at a separation from content and presentation and often supply worklflow support.

CSS .... Cascading Stylesheets Style language for HTML/XML.

CVS .... Concurrent Version System Version control system commonly used in open source development. Boasts concurrent checkouts. See (CVS Organisation, 2003; Subersion Group, 2003) for details.

DL .... Distance Learning Any form of education where teacher and pupil are geographically separated. DL has been around since the 19th century. Currently many DL programms are switching from mail to Internet as their foremost transport channel.

DOM .... Document Object Model Tree based document model associated with SGML/XML languages. Also one of two major XML processing concepts (the other being SAX).

DTD .... Document Type Definition A Kind of Grammar for XML applications. Lists possible elements and their occurance. Compare XSD.

EML .... Educational Markup Language XML-based language to markup educational content. Developped by Open University Netherlands.

FRESS .... File Retrieval and Editing System First Hypertext System to use vector graphics and windowing technology. also the first hypertext system used for teaching.

GPL .... Gnu Public License Copyleft license by the GNU project. Free speechh, not free beer software.

GUI .... Graphical User Interface Computer interface that relies on graphical representation

HDM .... Hypertext Design Model

HES .... Hypertext Editing System First Hypertext system to see reals world use in the Apollo spacecraft program.

HTML .... HyperText Markup Language Language in which pages for the WWW are written. Derived from SGML. Now being replaced by XHTML derived from XML.

HTTP .... HyperText Transfer Protocol Protocol to transfer (X)HTML pages and other digital media through the Internet.

ICQ........... I Seek You ICQ is a representative of the growing number of IMS.

IMS........... Instant Messaging Systems Synchronous Person-to-person message, chat, file transfer tools. Many of these systems allow many-to-many communication and one-to-one video-conferencing. Representatives include: MS Netmeeting, ICQ, AIM, Yahoo Messenger,...

IRC........... Internet Relay Chat Synchronous many-to-many communicating tool. It consists of various separate networks of IRC servers. These Servers provide various channels (each designated to special topic) where people can converse with each other.

ITS .... Intelligent Tutoring System

JDBC......... Java DataBase Connectivity Unfied JAVA API for database access.

JRE........... JAVA Runtime Environment The Java Runtime Environment provides the minimum requirements for executing a Java application; it consists of the Java Virtual Machine (JVM), core classes, and supporting files.

JSP .... Java Server Pages A way of mixing JAVA source code and HTML markup code to produce dynamic web-pages.

K12........... Kindergarten to Grade 12 or graduation Anglo Sachsan term for primary and secondary (high school) education

LAMP .... Linux – Apache – MySQL – PHP A commonly used web-application development framework actually comprised of four distinct parts like the operating system Linux, the Apache Webserver, the MySQL relational database and the PHP programming language.

LDAP .... Lightweight Directory Access Protocol System for user data storage, address book services and user authentification.

LMP .... Learning Management Platform see LMS

LMS .... Learning Management System Backbone of virtual and *e*-nhanced universities provides everything from university management, student enrolment, content creation, content distribution to exams.

MATHML ..... Mathematical Markup Language A XML based markup language for writing mathematical formulas.

MBTI .... Myers-Briggs Type Indicator An instrument commonly used in the USA to measure a persons preferences/styles. The four scales are: extrovert-introvert, sensing-feeling, judging-perceiving, sensing-intuiting.

MIME .... Multi-purpose Internet Mail Extensions Standard way of classifying file types on the Internet (see glossary on mime-types).

MUSICML ..... Music Markup Language A XML based markup language for writing musical scores.

MVC .... Model View Controller Software deign paradigm that demands a separation of presentation layer, business logic and data access layer.

NFS .... Network File System ile sharing protocol mainly used by UNIX operating systems.

NLP .... Neuro Linguistic Programming NLP is a constantly evolving set of models, presuppositions, patterns, techniques, and observation-based theories resulting from the study of the structure of subjective experience, behavior and communication. Beyond understanding, NLP seeks to enable remedial and generative change quickly and ecologically.

NLS .... oNLine System Document centric System featuring word-processing and real-time computer interaction, invented by D. Engelbart in the early 1960s.

OCR .... Optical Character recognition Process in which textual information embdeed in bitmapped file is reclaimed and reconverted to text.

OOHDM ..... Object Oriented Hypertext Design Model A hypertext design model with an obeject orientated approach.

PDA .... Personal Ditgital Assistant Small handheld computers commonly used to store addresses, dates and memos, but recently turning into mobile Internet access tools.

PDF .... Portable Docuent Format WYSIWYG Document format for printing purposes, introduced by Adobe Inc..

PHP .... Personal Hypertext Processor A programming language frequently used for small to medium sized web-projects. Nown for its wide database support.

RMM .... Relationship Management Methodology A hypertext design model derived from Entity Relation Ship Modelling.

RTF .... Rich Text Format Data exchange format used by many word processors.

SAX .... Simple API for XML One of two major XML processing concepts (the other being DOM). Processes a XML file linearly throwing events along the way.

SCORM ...... Sharable Content Object Reference Model XML-based markup language proposed by ADL. Overly complete language designed to be useful for any type of learning application.

SGML .... Standard Generalised Markup Language A system to design new markup languages such as HTML or Docbook. Now mostly replace by its subset XML.

SMB .... Server Message Block File sharing protocol mainly used by the Windows operating system.

SMS .... Short Message Service Short messages interchanged in GSM mobile telephone networks.

SVG .... Scalble Verctor Graphics XML-based vector graphics format designed for use on the WWW.

TCO .... Total Cost of Ownership Cost introducing a new technology, including development or acquisition costs, training and maintenance cost and cost resulting from time loss by system usage.

TOT .... Table Of Topics Central tool for structuring *e*-ULE projects.

UML .... Unified Modelling Language Standard graphical tool for object oriented analysis and design.

WAI .... Web accessibility Initiative W3C Initiative to promote better accessibility of web pages.

WBT .... Web Based Training All types of WWW (and Internet) mediated Training. Essentially a subset of CBT.

WML .... Wireless Markup Language XML-based markup language (similar to HTML commonly used on **GSM!** celular phone platform.

WWW .... World Wide Web Collection of worldwide distributed documents written in HTML, transfered by HTTP an viewed in web-browsers. The WWW is a subsystem of the Internet.

WYSIWYG .. What You See Is What You Get Programs (e.g. word processors) that allow the user to preview and manipulate the material in the same view it would appear printed on paper.

XHTML ...... eXtensible HyperText Markup Language Language in which pages for the WWW are written. Derived from XML. Rreplacing HTML derived from SGML.

XML .... eXtensible Markup Language A subset of SGML. Used in the creation of new languages such as XHTML.

XSD .... eXtensible Schema Description Alternate way of describing XML grammars (schemas). Compare DTD.

XSL-FO ...... eXtensible Styleheet Language Formating Objects Stylesheet language for XML files. Printout orientated.

XSLT .... eXtensible Stylesheet Language Transformation XML based language used to convert between different XML applications.

XSP .... eXtensible Server Pages Technology similar to JSP, but using XML files in place of HTML files.

XUL .... eXtensible User interface Language XML application for describing user interfaces. Used by the Mozilla webbrowser.

Glossary

brain storm mode Special authoring mode in the *e*-ULE authoring client that memorises concepts at the press of the enter key. Concepts are stored in the idea repository for later reference (e.g. by the information agent).

checkout All *e*-ULE project data resides on the server. Documents are checked out individually for editing. The server takes care of versioning, collaborative editing and backup.

Common-sense-checker An enhanced spell-checker that can not only correct the word spelling but can also turn word into *e*-ULE keywords.

document repository A place where documents are stored. Documents include scientific papers, web-pages (offline copies) and legacy documents in various formats. Compare repository, idea repository.

docuverse Term coined by Ted Nelson to describe his universal document database/hypertext system Xanadu.

examples Examples are demonstrating practical application of the material covered in the topic

exercises Exercises are to be solved by the student, an optional solution can be provided

idea repository A place where random user thoughts and brainstorming sessions are stored. Compare repository, document repository.

Link-builder Semi-automatic construction of hypertext based on author defined keywords. The identification of keywords is helped bye the Common-sense-checker.

The link builder will automatically replace any keyword (or grammatically variation thereof) with a hyper-link to the topic explaining the keyword, the glossary entry and other occurances of the keyword in question.

logicsheets Special XSLT Stylesheet used to recombine the presentation layer and business logic that were separated according to the MVC model. Used in the Cocoon publishing framework.

mime-type Standard way of classifying file types on the Internet (see MIME (Multipurpose Internet Mail Extensions)).

MVC *Model View Controller* Software deign paradigm that demands a separation of presentation layer, business logic and data access layer.

project Every course is mapped to *e*-ULE project. A project collects all information including notes, literature and the materials developed therein as well as course meta information like times and places. A project is divided into topics

semantic web The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.

sub lecture note Excerpt of a *e*-ULE project dealing only with an aspect of the contained material.

taglib Aggregation of business logic in form of XML tags to be included in the presentation layer. Necessary for implementing a MVC model.

topic *e*-ULE term for a undividable content block. A topic is the smallest information unit in the *e*-ULE system. Think of it as subsection or subusbsection.

TOT *Table Of Topics* Central tool for structuring *e*-ULE projects.

trail A succession of hyper-links forming a path to knowledge through the vast body of information. First proposed by Vannevar Bush in his article *As we may think* (Bush, 1945a), (Bush, 1945b).

transclusion A term coined by Ted Neslon for his Xanadu hypertext system. Transclusion provides methods for deep structuring of text and links as objects, allowing version management, re-use and republication, and ownership concepts of these objects.

view *e*-ULE support various ways to the information. The lecturer can provide various TOT and the student can rearrange and excerpt the content into views.

Wiki The simplest online database that could possibly work. Wiki is a piece of server software that allows users to freely create and edit Web page content using any Web browser. Wiki supports hyperlinks and has a simple text syntax for creating new pages and crosslinks between internal pages on the fly.

XSLT *eXtensible Stylesheet Language Transformation* XML based language used to convert between different XML applications.

Bibliography

Ackoff, R. L. (1981). *Creating the Corporate Future : Plan or Be Planned.* John Wiley & Sons.

ActiveState (2003). Komodo IDE. http://www.activestate.com/Products/Komodo/.

Advanced Distributed Learning Initiative (2002). Scorm. http://www.adlnet.org/.

Alliance, L. (2003). Project Liberty. http://www.projectliberty.org/.

Anderson, J. R. (1993). *Rules of the Mind.* Lawrence Erlbaum Associates.

Apache Software Foundation. Xindice – native XML database. http://xml.apache.org/xindice/.

Apache Software Foundation (2003a). Apache Jakarta Project. http://jakarta.apache.org/.

Apache Software Foundation (2003b). Apache Tomcat Server. http://jakarta.apache.org/tomcat/index.html.

Apache Software Foundation (2003c). ASF. http://www.apache.org/.

Apache Software Foundation (2003d). Coocon XML Publishing Framework. http://xml.apche.org/cocoon/.

Assink, E. and van der Linden, J. (1991). Computer controlled spelling instruction: a case study in courseware design. In *American Educational Research Association, Annual Meeting 1991.*

Baumgartner, P., Häfele, H., and Maier-Häfele, K. (2002). *E-Learning Praxishandbuch, Auswahl von Lehrplattformen.* Studienverlag.

Blackboard Inc (2003). Blackboard LMS. http://www.blackboard.com/.

Bodendorf, F. (1990). *Computer in der Fachlichen und universitären Ausbildung.* Oldenburg.

Boswell, D., King, B., Oeschger, I., Collins, P., and Murphy, E. (2002). *Creating Applications with Mozilla.* O'Reilly.

Bra, P. D. and Calvi, L. (1998). AHA: a generic adaptive hypermedia system. In *Proc. 2nd workshop on Adaptive Hypertext and Hypermedia.*

Brightman, H. J. and Robinson, J. M. (1998). Master teacher brochure. http://www.gsu.edu/ dschjb/masterteacher.html, http://www.gsu.edu/ dschjb/wwwmbti.html.

Brooks, J. G. and Brooks, M. G. (1999). *In Search of Understanding: The Case for Constructivist Classrooms.* Association for Supervision & Curriculum Development;.

Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3):87–129.

Brusilovsky, P. (1998). Adaptive educational systems on the world-wide-web: A review of available technologies. In *Proceedings of Workshop WWW-Based Tutoring.*

Brusilovsky, P. (2000). Course sequencing for static courses? Applying its techniques in large-scale web-based education. In *Proceedings of the* 5^{th} *International Conference on Intelligent Tutoring Systems*, number LNCS 1839. Springer.

Brusilovsky, P., Schwarz, E., and Weber, G. (1996). Elm-art: An intelligent tutoring system on world wide web. In .Frasson, C., editor, *Intelligent Tutoring Systems. Lecture Notes in Computer Science, Vol. 1086.*

Bullard, V., Smith, K. T., and Daconta, M. C. (2001). *Essential XUL Programming.* Wiley.

Bush, V. (1945a). As we may think. *The Atlantic Monthly*, 176(1).

Bush, V. (1945b). As we may think. *Life*, 19(11). Abridged version of the articel with additional illustrations.

Conklin, J. (1987). Hypertext: An introduction and survey. *IEEE Computer.*

Constantine, L. L. and Lockwood, L. A. D. (1999). *Software for use. a practical guide to the models and methods of usage-centered design.* New York, NY: ACM Press/Addison-Wesley Publ. Co., p. 579, $44.95 .

CVS Organisation (2003). CVS concurrent version system. http://www.cvshome.org/.

Debian Project (2003). Debian Linux. http://www.debian.org/.

Edelmann, W. (2000). *Lernspychologie.* Urban & Schwarzenberg.

Eden, A. (2003). Wafer. http://www.waferproject.org/index.html.

Engelbart, D. C. (1962). Augmenting human intellect: A conceptual framework. Technical report, STANFORD RESEARCH INSTITUTE.

Felder, R. M. (1993). Reaching the second tier: Learning and teaching styles in college science education. *Journal of College Science Teaching*, 23(5):286–290.

Felder, R. M. (2003). Resources in science and engineering education. http://www.ncsu.edu/felder-public/.

Felder, R. M., Felder, G. N., and Dietz, E. J. (2002). The effects of personality type on engineering student performance and attitude. *Journal of Engineering Education*, 91(1):3–17.

Felder, R. M. and Silverman, L. K. (1988). Learning and teaching styles. *Engineering Education*, 78(7):674–681.

Ferguson, D. (1992). Computers in teaching and learning. In *New Direction in Educational Technology.* Springer Heidelberg.

Foundation, F. S. (1996a). The free software definition. http://www.gnu.org/philosophy/free-sw.html.

Foundation, F. S. (1996b). Philosophy of the gnu project. http://www.gnu.org/philosophy/philosophy.html.

Foundation, F. S. (1999). Various licenses and comments about them. http://www.gnu.org/licenses/license-list.html.

Fritze, A. (2003). Mozilla SVG Project. http://www.mozilla.org/projects/svg/.

Garrison, R. and Anderson, T. (2000). Transforming and enhancing university teaching: stroger and weaker technological influence. In Evans, T. and Nation, D., editors, *Changin University teaching*, chapter 2.

Garzoztto, F., Paolini, P., and Schwabe, D. (1993). HDM: A model-based approach to hypertext application design. *ACM Transactions of Information Systems.*

Gloor, P. and Streitz, N. (1990). *Hypertext und HYpermedia.* Springer Heidelberg.

Goldberg, M. W., Salari, S., and Swoboda, P. (1996). World wide web - course tool: An environment for building www-based courses. *Computer Networks and ISDN Systems*, 28.

Google Inc. (2003). Google. http://www.google.com/.

Grasha, A. F. and Richlin, L. (1996). *Teaching with Style: A Practical Guide to Enhancing Learning by Understanding Teaching and Learning Styles.* Alliance Publishers.

Herkner, W. (1992). *Psychologie.* Springer-Verlag.

Holland, J. L. (1973). *Making vocational choices.* Prentice Hall.

Holmgren, C. and Tomitsch, M. (2003). Discussion forums and chats for e-ULE. Technical report, RISE, TU Wien.

Howell, J. (2003). FAQMatic. http://faqomatic.sourceforge.net/.

Hull, C. (1943). *Mathematico-Deductive Theory of Rote Learning.* Yale University Press.

IBT (2003). IBT eLearning suite. http://www.time4you.de.

IETF (2003). RFC2068 – hypertext transfer protocol – HTTP/1.1. ftp://ftp.isi.edu/in-notes/rfc2616.txt.

Ilias (2003). ILIAS LMS. http://ilias.uni-koeln.de/.

IMS (2002). IMS Specifications. http://www.imsproject.org/.

Ingenta INC (2003). ingenta. http://www.ingenta.com/.

Inspiration Software, Inc. (2003). Inspiration. http://www.inspiration.com/.

Isakowitz, T., Stohr, E. A., and Balasubramanian, P. (1995). RMM: A methodology for structured hypermedia design. *Communications of the ACM*, 38(8):34–44.

ITU (1995). X.500. Technical report, ITU.

Jabberzilla Project (2003). Jabberzilla Instant Messaging and Whiteboard. http://jabberazilla.mozdev.org/.

Jones, V. and Jo, H. (1998). Interactive multimedia based on learning theories to enhance tertiary education. In *Conference proceedings ICCMA '98.*

Kevin Atkinson (2003). aspell Spell-checking engine. http://aspell.sourceforge.net/.

Knopper, K. (2003). Knoppix Linux Live CD. http://www.knopper.net/knoppix/.

Knorr, D. (1997). Verwaltung von fachtextinformationen. In Knott, D. and Jakob, E.-M., editors, *Textproduktion in elektronischen Umgebungen.* Peter Lang.

Koch, N. (1999). A comparative study of methods for hypermedia development.

Kolb, D. (1997). Scholarly hypertext: Self-represented complexity. In *Hypertext 97*.

Kolb, D. A. (1978). *Learning Style Inventory technical manual*. McBer & CO.

Krapp, A. and Weidemann, B. (2001). *Pädagogische Psychologie*. Urban & Schwarzenberg.

Lefrancois, G. R. (1994). *Psychologie des Lernens*. Springer-Verlag.

Leuf, B. and Cunningham, W. (2003). Wiki system. http://wiki.org/.

Mayhew, Deborah J. [Deborah J. Mayhew & Associates, M. (1999). *The usability engineering lifecycle. a practitioner's handbook for user interface design.* San Francisco, CA: Morgan Kaufmann Publishers Inc., p. 542 .

Mayo, E. (1933). *The Human Problem of an Industrialised Civilisation*. Macmillan Publishing.

McKendree, J., Reader, W., and Hammon, N. (1995). The "homeopathic fallacy" in learning from hypertext. In *Interactions*.

McKnight, C., Dillon, A., and Richardson, J., editors (1993). *Hypertext: A Psychological Perspective*. Prentice Hall.

Meier, W. M. (2003). eXist – native XML database. http://exist.sourceforge.net/.

Mendes, M. and Hall, W. (1999). Hyper-authoring for education: a qualitative evaluation. *Computer & Education*.

Mindjet GmbH (2003). Mindmanger. http://www.mindjet.de/.

MIT (2003). Mit opencouseware. http://web.mit.edu/ocw/.

Mitchell, C., Dipetta, T., and J., K. (2001). he frontier of web-based instruction. *Education and Information Technologies*, 6(2).

Molitor-Lübbert, S. (1997). Wissenschaftliche textproduktion unter elektronischen bedingungen. In Knott, D. and Jakob, E.-M., editors, *Textproduktion in elektronischen Umgebungen*. Peter Lang.

Montelpare, W. J. and Williams, A. (2000). Web-based learning: Challenges in using the internet in the undergraduate curriculum. *Education and Information Technologies*, 5(2):85–101.

Moundridou, M. and Virvou, M. (1992). Authoring and delivering adaptive web-based textbooks using WEAR. In *IEEE International Conference on Advanced Learning Technologies; Issues, Achievements, and Challenges - ICALT*.

Mozdev Oragnisation (2003a). Mozdev.org. http://www.mozdev.org/.

Mozdev Oragnisation (2003b). Mozilla spellchecker project. http://spellchecker.mozdev.org/.

Mozilla group (2003a). Chatzilla IRC client. http://www.mozilla.org/projects/rt-messaging/chatzilla/.

Mozilla group (2003b). Mozilla project. http://www.mozilla.org/.

Mozilla Organisation (2003a). Mozilla Directory (LDAP) SDK. http://www.mozilla.org/directory/.

Mozilla Organisation (2003b). XPCOM. http://www.mozilla.org/projects/xpcom/.

Mozilla Organisation (2003c). XPInstall. http://www.mozilla.org/projects/xpinstall/.

Murray-Rust, P. and Rzepa, H. (2003). CML Chemical Markup Language. http://www.xml-cml.org/.

Myers, I. B. (1962). *Manual: The Myers-Briggs Type Indicator.* Consulting Psychologists Press, Palo Alto.

Naber, L. (2001). Hypertext support for learning phases, tech. report ifs01-36. Technical report, Viennna University of Technology.

Naber, L. (2002a). Authoring tool requirements analysis. tech. report ifs02-33. Technical report, Viennna University of Technology.

Naber, L. (2002b). Student motivation survey. tech. report ifs02-31. Technical report, Viennna University of Technology.

Naber, L. and Köhle, M. (2002a). If e-learning is the answer, what was the question? *Proceedings of the AUSWEB Conference.*

Naber, L. and Köhle, M. (2002b). University teaching in the "post xerox" age – system e-ule. *Proceedings of the International Conference on Web-based Learning.*

Naber, L. and Köhle, M. (2003). e-nhance Lectures. *JoDI - Journal of Digital Information*, 3.

NCBI (2003). Journal publishing dtd. http://dtd.nlm.nih.gov/publishing/.

Nelson, T. H. (1965). A file structure for the complex, the changing, and the indeterminate. In *Proceedings of the 20th National Conference.* ACM, Lewis Winner.

Nelson, T. H. (1982). *Literary Machines.* Mindful Press.

Nielsen, J. (1993). *Usability engineering.* Orlando, FL: Academic Press. xiv, 362 p.

Nielsen, J. (1998). Fighting linkrot. http://www.useit.com/alertbox/980614.html.

Nielsen, J. (2001). First rule of usability? Don't listen to users. http://www.useit.com/alertbox/20010805.html.

Nielsen, J. (2003a). Papers, essays and recomendations. http://www.useit.com/papers/.

Nielsen, J. (2003b). Ten usability heuristics. http://www.useit.com/papers/heuristic/heuristic*list.html.*

Nkambou, R., Frasson, C., and Gauthier, G. (1998). A new approach to ITS-curriculum and course authoring: the authoring environment. *Computer & Education.*

OASIS (2002). xml.org. http://www.xml.org/xml/registry.jsp.

OASIS (2003). SAML – Security Assertion Markup Language. http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security.

Open Office Group (2003). MySpell spell-checking engine. http://lingucomponent.openoffice.org/dictionary.html.

Open University Netherlands, O. (2002). EML specifications. http://eml.ou.nl/.

O'Reilly (2003). Mozilla devcenter. http://www.oreillynet.com/mozilla/.

Ovid Technologies Inc (2003). Ovid platform. http://www.ovid.com/.

Phillips, K. A., Soltis, J. F., and Phillips, D. C. (1991). *Perspectives on Learning.* Teachers College Pr;.

PostgreSQL Group (2003). PostgreSQL. http://www.postgresql.org/.

Rhodes, B. (2002). The remembrance agent. http://www.remem.org/.

Rhodes, B. J. (2000). *Just-In-Time Information Retrieval.* PhD thesis, MIT Media Lab.

Rhodes, B. J. and Maes, P. (2000). Just-in-time information retrieval agents. *IBM SYSTEMS JOURNAL,*, 39(3&4).

SAP (2003). SAP DB. http://www.sapdb.org/.

Schell, G. (2001). Student perceptions of web-based course quality and benefit. *Education and Information Technologies.*

Schulmeister, R. (2002). *Grundlagen hypermedialer Lernsysteme.* Oldenburg.

Schulmeister, R. (2003). *Lernplattformen für das virtuelle Lernen.* Oldenburg.

Schwabe, D. and Rossi, G. (1995). The object-oriented hypermedia design model. *Communications of the ACM*, 38(8):45–46.

Shackelford, R. (1990). Educational computing: Myths versus methods. why computers haven't helped and what we can do about it. In *ACM SIGCAS Computers and Society, Proceedings of the Conference on Computers and the Quality of Life.*

Sharp, J. (1998). Learning styles and technical communication: Improving communication and teamwork skills. In *Proceedings of the Frontiers in Education 1998 29th Annual Conference.*

sharp, J. (2001). Teaching teamwork communication with kolb learning style theory. In *Proceedings of the Frontiers in Education 2001 32nd Annual Conference.*

Sidje, R. B. (2003). MathML in Mozilla. http://www.mozilla.org/projects/mathml/.

Sternberg, R. J. (1999). *Thinking Styles.* Cambridge University Press.

Subersion Group (2003). Subversion, version system. http://subversion.tigris.org/.

Sumner, T. and Taylor, J. (1998). New media, new practices: experiences in open learning course design. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 432–439. ACM Press/Addison-Wesley Publishing Co.

Sun Mcrosystems (2003). JAVA Development Framework. http://www.java.com/.

Swanson, S. (2003). mathmled - MathML Editor for Mozilla. http://www.newmexico.mackichan.com/MathML/mathmled.htm.

The ht://Dig Group (2003). ht://dig. http://www.htdig.org/.

TheBrain Technologies Corporation (2003). personal brain. http://www.thebrain.com/.

Thorndike, E. (1932). *The Fundamentals of Learning.* Teachers College Press.

Thüring, M., Hannemann, J., and Haake, J. M. (1995). Hypermedia and cognition: Designing for comprehension. *Communication of the ACM*, 38(8).

Trolltech Inc. (2003). QT Application Framework. http://www.trolltech.com/products/qt/index.html.

Tsichritzis, D. (1999). Reengineering the university. *CACM*, 42(6).

Turner, D. and Oeschger, I. (2003). *Creating XPCOM Components.* Mozilla Organisation: http://www.mozilla.org/projects/xpcom/book/cxc/.

unikat (2002). Sides4mi - das personalisierte informations- und kommunikationssystem für die lehre. http://www.lzk.ac.at/sides-4mi.html.

University of Michigan (2003). LDAP – Light Directory Access Protcoll. http://ldap-project.berkeley.edu/.

W3C (2003). Digital signature initiative. http://www.w3.org/DSig/.

Watson, D. (2001). Pedagogy before technology: Re-thinking the relationship between ICT and teaching. *Education and Information Technologies*, 6(4).

WebCT, Inc (2003). Webct campus, LMS. http://www.webct.com/.

Werneck, H. (2003). The discussion groups at the vienna university department of psychology. Direct communication.

Wolf, G. (2003). The curse of xanadu. http://www.wired.com/wired/archive/3.06/xanadu.html.

World Wide Web Consortium, W. (2003). Web accessibility initiative (wai). http://www.w3.org/WAI/.

Wu, H., Houben, G., and Bra, P. D. (1999). Authoring support for adaptive hypermedia applications. In *Proceedings of ED-MEDIA 99.*

Xanadu, P. (2003). Xanadu – the information future. http://xanadu.com.au/general/future.html.

XML:DB Initiative (2003). XML:DB initiative for XML databases. http://www.xmldb.org.

Curriculum Vitae

Curriculum Vitae

Personal information

Name	Dipl. Ing. Larissa Naber
Address	Kaiserstraße 101/1/8 1070 Wien
Phone	+43 1 524 94 84
E-mail	larissa@naber.at
Date of birth	16. 01. 1973
Maritial status	Married, no kids
Nationality	Austrian

Work Experience

04/1998-09/2000	Research Assistant, Institute for Experimental Physics, Vienna University of Technology
since 01/1998	Self-employed – Internet Consulting: Extranet conception, Interface Design, Web-design Cybertron, MBUJF, TIPS, ITC, Computer Associates
since 01/1998	Self-employed – Training and Education: Venetia, New Horizons, IP Center - Office, Internet, Web-design Wifi Wien - Internet-based Research, Web Usability, Web Project Management, Web Technology: HTML, CSS, XML, PHP, Javascript, Web-server Architecture, Unix
since 2001	University Lecturer, Institute Software Engineering and Interactive Systems, Vienna University of Technology – Usability, Interface Design, *e*-learning , Software Engineering
since 2001	Part-time Researcher, Institute Software Engineering and Interactive Systems, Vienna University of Technology – Usability, *e*-learning
since 2002	College Lecturer, FH Technikum Wien – Internet Technology, Databases
since 2003	College Lecturer, FH Hagenberg – XML

Internships

08/1991	Siemens Wien, Systems Support
07/1992	Siemens Wien (PSE – Program/System Development) – Documentation

07/1993	PSE Siemens Wien, Documentation, Tests
07/1994-08/1994	PSE Siemens Wien, Databases, Documentation

Education – University

10/1991–06/1997	Technical physics, Vienna University of Technology Principal subjects: Experimental Solid State physics, Superconductivity
10/1995–12/1995	Research Internship, Siemens Research Labs Erlangen
Since 06/1997	Vienna University of Technology Graduate School (Phd) – High Pressure Physics
Since 12/2000	Vienna University of Technology Graduate School (Phd) – Usability of University *e*-learning Systems, to be finished this summer

Education – Courses and Seminars

Rhetoric, Presentation techniques (German and English), Visualisation Techniques, Communication NLP (Neuro Linguistic Programming) – *perfact training* and Wifi Wien.

Education – School

09/1979–06/1983	Primary school, 1210 Wien
09/1983–06/1991	Grammar School, Neusprachliches Gymnasium der Schulbrüder in Wien Strebersdorf

Language Skills

German	native speaker
English	Fluent
French	Reading skills fair, writing/speaking skills basic

Publications

L. Naber and M. Köhle. *e*-nhance Lectures. *JoDI - Journal of Digital Information*, 3, 2003.

L. Naber and M. Köhle. University teaching in the "post xerox" age – system *e*-ule. *Proceedings of the International Conference on Web-based Learning*, 2002.

L. Naber and M. Köhle. If *e*-learning is the answer, what was the question? *Proceedings of the AUSWEB Conference*, 2002.

M. Ellerby, K.A. McEwen, M. Watmough, M.A. López de la Torre, L. Naber, and E. Bauer. Influence of yttrium substitution on the magnetic and transport properties of uCu_5. *Journal of Physics: Condensed Materials*, 13, 2001.

R. Pelzer, L. Naber, A. Galatanu, H. Sassik, and E. Bauer. Pressure response of $Fe_{1-x}Si_{1+x}$, - $0.003 \leq x \leq 0.025$. *Journal of Magnetic Materials*, 226 - 230, 2001.

E. Bauer, A. Galatanu, L. Naber, M. Galli, F. Marabelli, C. Seuring, K. Heuser, E.W. Scheidt, T. Schreiner, and G.R. Stewart. Onset of magnetism and Fermi-liquid instabilities in Yb compounds. *Physica B*, 281 & 282, 2000.

A. Galatanu, R. Hauser, G. Hilscher, H. Michor, L. Naber, and E. Bauer. Low-temperature behaviour of $CePd_{2-x}Ni_xAl_3$. *Physica B*, 281 & 282, 2000.

H. Michor, M. El-Hagary, L. Naber, E. Bauer, and G. Hilscher. Effect of hydrostatic and chemical pressure on the exchange interaction in magnetic borocarbide superconductors. *Physical Review*, 61, 2000.

E. Bauer, M. Liendl, L. Naber, D. Werner, H. Michor, G. Hilscher, A. Dönni, P. Fischer, F. Fauth, and M. Zolliker. Transport, magnetic and thermodynamic properties of $REPd_2Ga_3$,RE = Pr,Nd,Sm. *Z. Phys*, 102, 1997.

R. Hauser, L. Naber, G. Schaudy, E. Bauer, G. Hilscher, and W. Assmus B. Kindler. Pressure and field dependent studies of single crystalline $ybcu_4in$. *Czech. J. Phys.*, 46, 1996.

H. Michor, G. Hilscher, E. Bauer, R. Hauser, R. Krendelsberger L. Naber, and P. Rogl. Superconducting properties of $La_3Ni_2B_2N_3$. *Czech. J. Phys.*, 46, 1996.

H. Michor, R. Krendelsberger, G. Hilscher, E. Bauer, R. Hauser C. Dusek, L. Naber, D. Werner, P. Rogl, and H.W. Zandbergen. Superconducting properties of $La_3Ni_2B_2N_{3-d}$. *Physical Review B*, 54, 1996.